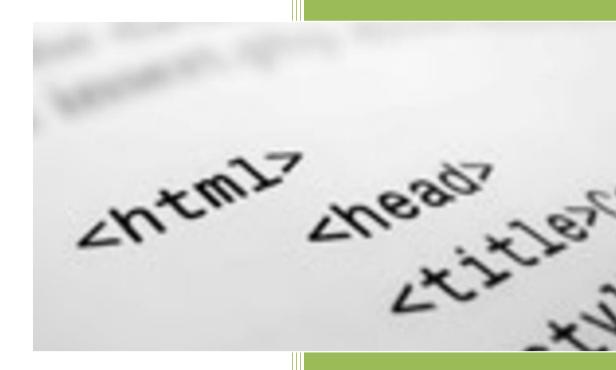


Introduction to Web Development



Martin Schedlbauer, Ph.D.

Ars Doceo

Version 1.01

Introduction to Web Development Published by Ars Doceo Press Boston, Massachusetts

Copyright © 2015 by Martin Schedlbauer ALL RIGHTS RESERVED.

Printed in the United States of America

Library of Congress Cataloging-in-Publication Data

Schedlbauer, Martin J.
Introduction to Web Development
M. Schedlbauer
p.cm
Includes index.

1. Web Development. 2. Computer Software – Development

The information contained in this document is subject to change without notice.

The programs, applications, diagrams, models, and examples presented in this book have been included for their instructional value only. They have been prepared with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs or applications. The author shall not be liable for errors contained herein, or for any direct or indirect, incidental, special or consequential damages in connection with the furnishing, performance, or use of this material.

The author reserves all rights, including copyright, in all aspects of this book, including any software and models. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher, except for brief quotations in review articles or online posts.

No challenge to the status or ownership of these or any other trademarked terms contained herein is intended by Ars Valorem Press or Martin Schedlbauer.

Special thanks for Sahil Chhabra for his diligent efforts in editing several versions of the manuscript.

Table of Contents

ARCHITECTURE OF THE WEB	11
WHAT IS THE WEB?	11
INTERNET	
4G/3G	
IEEE 802.11	
T1	
COMMUNICATION PROTOCOLS	_
HTTP	
SMTP	
FTP.	_
PROGRAMMING LANGUAGES	_
HTML	
CSS.	
JavaScript	
WHAT IS WEB DEVELOPMENT?	
WHAT DO I NEED TO LEARN?	
BROWSERS AND WEB SERVERS	
DOCUMENTS	
DYNAMIC DOCUMENTS	_
GENERATING DYNAMIC DOCUMENTS	_
THE URL	
FORMAT OF THE URL	
PROTOCOLS	
SUPPORTED PROTOCOLS	
DOMAIN OR IP ADDRESS	
DOMAIN NAME SYSTEM (DNS)	
WHAT IS WITH THE 'WWW' PREFIX?	
TOP-LEVEL DOMAIN (TLD)	
HOW DO I GET A TLD?	
HOW TO FIND IF A DOMAIN NAME IS ALREADY IN USE?	
LONG URLS AND URL SHORTENING	
URL SHORTENING	
How the HTTP works	
CONNECTING TO THE INTERNET	
How does Wi-Fi work?	_
SEARCH ENGINES	
SUMMARY	
SUMMARY	24
PLANNING A WEBSITE	25
CONTENT PLANNING	25
Know The User!	25

USE CASE	25
USE CASES AND MINI STORIES	26
UNIFIED MODELLING LANGUAGE (UML)	26
CONTENT PLANNING WITH CARD SORTING	27
TRACKING CONTENT WITH INVENTORIES	28
CONTENT INVENTORY	28
CONTENT INVENTORY ELEMENTS	29
PAPER AND SKETCHBOARDS	30
SITE MAP DIAGRAMS	33
ITERATIVE DESIGN	36
USABILITY TESTING	36
SUMMARY	38
USING MARKUP LANGUAGES: HTML & MARKDOWN	39
What is HTML?	39
ROLE OF HTML	39
HTML MARKUP TAGS	41
A SIMPLE HTML PAGE	42
Head tag	42
HEADINGS	43
PARAGRAPHS	43
EXAMPLE	44
PRE TAG	44
EXAMPLE	45
FORMATTING	45
NESTING HMTL TAGS	47
HORIZONTAL LINES	48
LINE BREAKS	48
LINKS	49
OPENING A NEW WINDOW/TAB	49
LINKS WITHIN A PAGE	50
IMAGES	50
IMAGE LINKS	51
LISTS	52
UNORDERED LIST	52
ORDERED LIST	52
DESCRIPTION LISTS	53
NESTED LISTS	54
BLOCKS OF TEXT	54
<div></div>	55
	55
TABLES	56
TABLE BORDER	56
CELL SPACING	56
CELL SPACING	57

TABLE HEADERS	57
TABLE CELL SPANS	58
Table Row Spans	58
Table Column Spans	59
TABLES WITH CSS	59
ALIGNING TEXT WITHIN TABLES	61
ARRANGING ITEMS IN COLUMNS	62
STYLES WITH CSS	62
Inline Styles	63
Internal Style Sheets	63
EXTERNAL STYLE SHEETS	63
CSS STYLE FORMATS	66
BACKGROUND STYLES	66
SPECIFYING COLOR	66
TEXT FORMATTING	67
FONTS	67
FONTS, SPANS AND STYLES	
Margin	70
PADDING	
LAYOUT USING CSS	
ADDING CUSTOM FONTS	
HTML Entities	
IFRAMES	
IFRAME AS A TARGET FOR A LINK	
EMBEDDING YOUTUBE VIDEO IN IFRAME	
EMBED USING <embed/> TAG	
FRAME V/S <embed/>	
IFRAME	
Емвер	
HTML 5	
Adding New Elements to HTML	
AUDIO ELEMENT	
VIDEO ELEMENT	
HTML 5 GRAPHICS	
CANVAS	•
SVG	
HTML 5 SEMANTIC ELEMENTS	
HTML 5 SEMANTIC ELEMENTS	•
CREATING HTML PAGES WITH NOTEPAD	
FREQUENTLY ASKED QUESTIONS	
SUMMARY	
SUMINARY	03
INTEGRATING MULTIMEDIA CONTENT	<u>85</u>
IMAGES	85
IMPORTANCE OF IMAGES	
Sources for Images	

EXAMPLES OF IMAGES	86
USING IMAGES FROM THE WEB	87
CREATIVE COMMONS LICENSE	88
SCANNING IMAGES	89
Screen Resolution	89
PIXELS AND COLOR	
IMAGE SCALING	91
IMAGE FILE FORMATS	91
IMAGE COMPRESSION	92
GIF	93
ANIMATED GIFS	93
PNG	94
TIFF	94
TIFF VERSUS JPG	95
TIFF VERSUS GIF	95
FILE SIZE DIFFERENCES	95
In Practice	96
THUMBNAILS	97
THE TAG	97
STREAMING VIDEO AND AUDIO	97
ANIMATION	97
DOWNLOADING VS. STREAMING	97
Downloading:	98
Streaming:	98
AUDIO	98
Mono vs Stereo	98
Sampling rate	99
BIT DEPTH	99
8 bit audio vs 16 bit vs 128 bit	99
CODECS	99
STREAMING AUDIO	99
Stream Hosting	
VIDEO	100
Audio	100
DYNAMIC WEBSITES	101
DINAMIC WEDSITES	101
STATIC AND DYNAMIC WEBSITES	
WHY DYNAMIC WEBSITES?	
SCRIPTING ENGINES	
COMMON TECHNOLOGY STACKS	
Our Stack	
DYNAMIC PAGE GENERATION IN ZOHO	
ZOHO CREATOR	
DATA: DEFINITION & GENERATION	
ROLE OF A ZOHO CREATOR FORM	105

ADDING DATA TO THE DATABASE	106
Adding Data to ZohoDB by running the form	106
Adding Data to ZohoDB using built-in view	106
RETRIEVING THE DATA FROM ZOHODB	108
BUILT-IN LAYOUTS / REPORTS	108
CUSTOM DELUGE PAGES	109
EDITING PAGES IN ZOHO	111
WRITING DELUGE CODE	112
ACCESSING DATA	113
BASIC DELUGE DATA ACCESS	113
Access data from database	
FINDING MATCHING RECORDS	114
DELUGE MODE VS HTML	114
A COMPLETE EXAMPLE	115
EMBEDDING THE DYNAMIC PAGE	116
DYNAMIC PAGES IN SUMMARY	116
REFERENCES & BIBLIOGRAPHY	

Architecture of the Web

Upon completion of this chapter, you will be able to:

- ✓ Understand what is Web
- ✓ Basic structure of the Web and the most commonly used protocols
- ✓ What all you need to build a website
- ✓ How do computers connect to Internet &
- ✓ How do they interact with the web servers
- ✓ Static & Dynamic documents
- ✓ URL format and URL shortening
- ✓ DNS and TLDs
- ✓ How to get your own TLD and domain
- ✓ Ways to connect to the internet & more about Wi-Fi
- ✓ Search Engines

What is the Web?

WWW (World Wide Web) is a combination of all public websites connected to the Internet, including the clients as well, which have access to the web. It was invented by Tim Berners-Lee in 1989 while working at CERN in Geneva, Switzerland. The standards associated with the web are now maintained by the W3C (World-Wide Web Consortium) housed at MIT in Cambridge, MA.

The web is a combination of various technologies, programming languages, communication protocols, and standards:

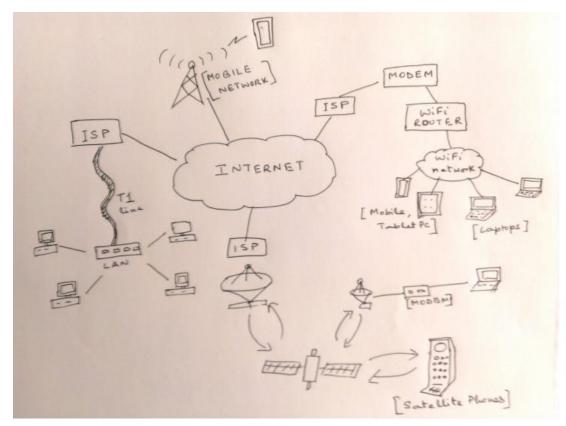


FIGURE 1.1 THE INTERNET

Internet

The Internet, which is a global system of interconnected computer networks, serves as a network transport mechanism. The computer networks exist in all kinds of sizes. Some popular and commonly used networks are:

4G/3G

4G (Fourth Generation) and 3G (Third Generation) are mobile communication standards, which allow wireless Internet service at high speeds. 3G came to the US in 2003. It marked the genesis of mobile broadband with minimum Internet speed of 144 Kbps. With time, many varieties of 3G came into existence, which could offer Internet speeds anywhere from 400 Kbps to almost ten times that. 4G on the other hand was supposed to be much faster, but that's not true always. There are various 4G technologies offering different Internet speeds. The most popular ones are HSPA+ 21/42, WiMAX and LTE.

IEEE 802.11

The 802.11 and 802.11x is a family of protocols (specifications or standards) developed by the IEEE for wireless local area network (WLAN). 802.11 specifically provide 1 or 2 Mbps speed on a 2.5 GHz band. There are other extensions of the 802.11 named the 802.11 family. Some of them are 802.11a, 802.11b, 802.11g, 802.11n, etc.

T1

T1 or the T-1 carrier is a digital transmission system for long distance communications. It is a very common type of leased lines used in telecommunications. A leased line is basically a reserved circuit between two distant points, which serves as a medium for voice and/or data communication. They maintain a single open circuit (operates over either copper or fiber optic cables) at all times between two points and do not use switching (used in traditional telephone service which uses the same line for different conversations). Their costs are very high and hence are generally rented out by organizations.

Communication Protocols

Collection of communication protocols such as:

HTTP

HTTP (Hypertext Transfer Protocol) is an application level protocol, which forms the basis of all kinds of communications over the Web. HTTP defines how the messages are formatted and transmitted, and how the browsers and web servers respond to certain commands, etc. For example, when you type an URL in a browser, a HTTP request is sent to the webserver, which contains the details about the requested resource and the directions on fetching it.

SMTP

SMTP (Simple Mail Transfer Protocol) is a set of communication guidelines/ Internet standards for electronic mail transmissions. Basically, it allows software to send and receive mails over the Internet. So, when you send an email, the email software interacts with the SMTP server to handle the email sending request. The SMTP server on your host may have to interact with other SMTP server over the Internet to deliver the email to the recipient.

FTP

FTP (File Transfer Protocol) is a network protocol used to transfer files from one computer to another over a network or the Internet. So, FTP can transfer files between any two computers, which are connected to the Internet. The entire transfer process is broken into two steps, uploading and then downloading. Transferring files from a client computer to a FTP server is called uploading, and transferring files from server to a client is called downloading.

Programming Languages

Along with the Internet & all the protocols, various programming languages such as HTML, JavaScript, CSS, PHP, etc. play an essential role in defining the Web.

HTML

HTML (Hyper Text Markup Language) is a markup language for describing the contents of a web page. It provides a medium to structure the contents of a web page. Contents are structured using something called as HTML tags, which have different properties and are meant to describe different contents of a webpage e.g. text, audio, image, headers, paragraphs, etc.

FIGURE 1.2 SAMPLE HTML CODE

Create a text document with an ".html" extension and paste the code above and run that file on a web browser.

CSS

CSS (Cascading Style Sheets) is used to define the look of an HTML document. It provides a framework to provide styling/formatting to a web page. HTML and CSS go hand-in-hand. CSS generally uses the "property: value" format which mostly applies to all HTML tags.

JavaScript

After HTML & CSS, JavaScript is one language all web developers must learn. JavaScript allows client-side scripting to interact with the users and manipulate the contents of the document. It can do so by changing attributes of HTML elements of a web page. JavaScript can even change the HTML styling i.e. CSS. The most common use of JavaScript is to validate the data entered by the users on a web page. For example, with JavaScript you can validate whether a user has entered a valid zip code on a registration form.

Even though you can build a good website with just HTML, CSS and JavaScript, knowledge of programming languages like PHP, Java, ASP.NET, etc. helps in building complex websites with business logic and for establishing client-server architecture.

What is Web Development?

Website development is the process of developing a website or web based applications for a network such as the Internet. It can be done using one or many programming languages. Web development is usually associated with the non-design aspects of the whole process such as writing markup (HTML, CSS) and coding (JavaScript, PHP, etc.).

Web development may involve client-side (web browser-side) and/or server-side programming. As discussed above, client-side programming is achieved using HTML, CSS & JavaScript. And server-side programming involves one or more programming languages like PHP, Java and ASP.NET.

What do I need to learn?

For client-side programming, the knowledge of HTML, CSS and JavaScript is required. It is advisable to start with HTML and learn the basic structure of an HTML document. Once we are familiar with the HTML tags and how they get displayed on the browser, we can understand how CSS works. JavaScript should come at last once we are familiar with the markup language.

And for server-side programming, the knowledge of at least one of the following is required:

- PHP
- Java (Java Server Pages, Java Servlet, etc.)
- Asp.Net
- Deluge
- Rubv

Browsers and Web Servers

The web is a simple client-server architecture. The client such as the web browser and the servers communicate via the HTTP. The client initiates the communication by sending requests for documents (mostly HTML documents which may include text, images, scripts, etc.) to the web servers. The web server then responds with the appropriate document or an error message if it is unable to do so. The browser interprets and displays the HTML files according to the HTML and CSS specifications. The W3C (World Wide Web Consortium) organization is responsible for maintaining these standards and the specifications of the web. Most of the browsers of today's age more or less conform to the specifications laid down by the W3C.

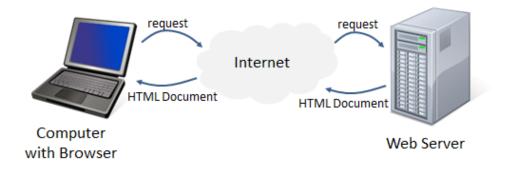


FIGURE 1.3 CLIENT-SERVER MODEL

Documents

Clients such as web browsers request documents from the web servers. These documents are mostly HTML documents but can also be PDF documents or image, audio or video files.

Such documents can be either static or dynamic:

- Static Documents are stored on the web server or some machine on the network. They are delivered as it is to the client machines. They can be images, audio files, text documents or even a web page, which does not change with time or user. Hence, we can say that all the users will get the same document.
- Dynamic Documents are generated "on-the-fly" (at the time it is being viewed or downloaded) by one or many programs running on the web server(s). The contents of a dynamic document vary with the user preferences and continuous flow of information.

The browser doesn't differentiate between static and dynamic documents. It treats them the same and just displays the contents according to the HTML document it receives.

Dynamic Documents

As discussed above, dynamic documents (also referred to as living documents) are generated "on-the-fly" i.e. only when it is requested. It may include some data retrieved from the database or some calculations based on the request. This allows the HTML document (page) to have the latest information for the client. This is useful since the clients receive live information and the user experience becomes interactive. Moreover, the execution of dynamic documents on the client side may involve a number of actions to generate the final view to the user. These may include:

- Other related documents to be fetched
- New HTML content or a completely new HTML page may be generated
- Some local client information may be accessed

Generating Dynamic Documents

Dynamic documents are generated through programs running on the web servers. These programs may have some programming logic, which generates content dynamically. It may be accessing the client's local time and location to present the local weather report or show all the local football matches being held over the week. Or, it may access some live database to fetch stock information and show the user the live stock positions of the hour. A huge majority of websites generate dynamic documents and the whole IT industry is always looking to find better programming models to achieve that. Such programs can be written in any of the following languages:

- PHP
- Iava
- Asp.Net
- Deluge
- Ruby

The URL

URL (Uniform Resource Locator) is the standard way of specifying the location of the documents on a web server. It is also called a *web address* when used along with HTTP. It is simply is sequence of characters that together forms a reference to the resource.

Tim Berners-Lee, the father of the World Wide Web, in 1995 brought the standard of URI (Uniform Resource Identifiers). The name was later changed to what we know it as today i.e. URL. He took the idea from a telephone directory and applied it to address millions of web pages and computers. As per Google, there were 26 million pages in 1998, which rose to 1 trillion in 2008.

Some examples:

- is1500.weebly.com/index.html
- www.google.com
- http://www.youtube.de

Format of the URL

The URL has a standard format. Some parts in it are optional and are automatically filled in by the browser based on conventions. All these components together can define the location of all web pages and computer machines over the Internet. URLs use forward slashes (/) and dots (.) to separate its parts.

The format is:

protocol://domainOrIPAddress:port/path/filename

protocol: Defines how the resource is obtained (https, http, ftp, etc.)

- domainOrIPAddress: It is the location of the URL. Domain names are not case sensitive.
- *port*: It is a decimal number and is optional. When absent from the URL, the default number for the specified protocol is used.
- *path & filename*: The path is used to specify and locate the requested resource on the server. Some web servers treat paths as case-sensitive and some don't.

Protocols

The protocol specified in a URL indicates which protocol will be used to obtain the resources. Although they are case-insensitive, but generally lowercase is used. They are optional for the user to type in. Modern browsers usually append the default protocol to the URL request before sending it to the servers.

The most common protocols used are HTTP, HTTPS (HTTP on top of SSL) and FTP. The SSL (Secure Socket Layer) is nothing but a security standard and works by establishing an encrypted connection between the clients and servers. It is important to protect sensitive information like social security, credit card number and login credentials. The HTTP is typically used to serve hypertext documents. Some URLs are not associated with any specific protocol (e.g. "file") and then there are some, which don't use the name of the protocol (e.g. "news").

Supported Protocols

While there are a lot of protocols, not all browsers support all of them. But the most commonly used protocols which are supported widely are:

- http & https
- file
- ftp
- news
- mailto

The web browsers cannot handle certain protocols directly like *news* and *mailto*. When a web browser gets a request with such protocols, it hands them off to another application. For example, all *mailto* requests are transferred to the default email application.

Domain or IP address

Domain name or the IP address is the address of the webserver which has the requested document/resource. Domain names are associated/mapped with IP addresses via DNS (Domain Name System). Although domain names and IP addresses can be used interchangeably, domain names are most commonly used. This is simple because users would remember domain names like "www.google.com" instead of "173.194.34.5". Also, typing a domain address on the address bar of a browser is much easier than typing an IP address, which has four numbers, separated by dots. But of course, all domain names are unique and map to a single IP address. Apart from all the technical reasons, domain

names have become very important for adding credibility to businesses and it also forms a medium for gaining successful presence in the cyber world.

Some examples of domain names are:

- www.company.tld
- www.yahoo.com

Domain Name System (DNS)

All the computer on the Internet are identified through a number (which is called the IP address) e.g. "173.194.34.5". It's the DNS who translates a Domain name to a specific IP address. It is often referred to as the phone book for the Internet. This is because the DNS translates some human-friendly computer names (domain names) into IP addresses. DNS are nothing but a cluster of servers over the internet which work together and translate all requests from the web clients.

For example: The domain "www.google.com" is mapped to its IP address of "173.194.121.17".

Anyone can find the IP address of a domain on their machines. You just have open your command prompt and type "nslookup <domain name>". This will list all the servers, which are currently handling all the requests to the provided domain.

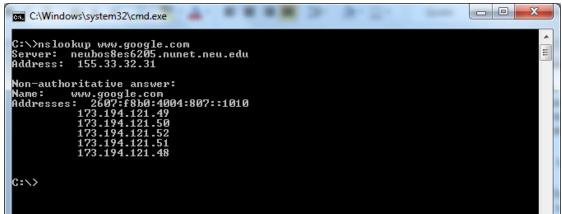


FIGURE 1.4 FETCHING IP ADDRESS USING COMMAND PROMPT

Go ahead and try out a few domain names to see what IP address they are mapped to.

Why does this command return multiple IP addresses? Didn't' we just discuss about domain names being unique and pointing to a single IP address. Well, all of that is true. The multiple addresses just means that there are multiple servers are serving the requests to that domain. This is a common load balancing practice since the loads are pretty high and we can't expect a single machine to cater to all the requests. Moreover, this also helps in achieving a high fault tolerance. There are many statistical models to manage the DNS's responses from the client machines and redirecting them to web servers. One such simplest model is called the round-robin DNS. Using this model, the DNS instead of returning a single IP address, now returns list of available IP addresses of servers hosting identical services. This list is returned on a round robin basis. And the

client usually attempts connection with the first address returned. Hence, the clients would receive address of a different server on the next connection attempt.

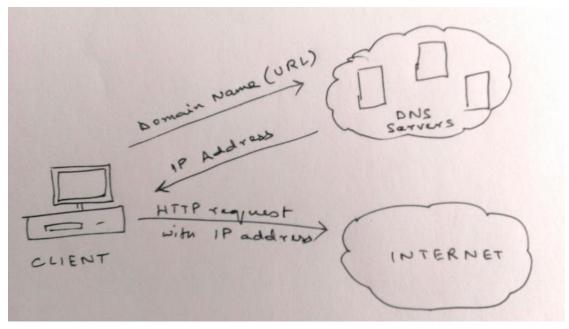


FIGURE 1.5 WORKING OF A DNS

What is with the 'www' Prefix?

The 'www' prefix that we see on most of the website's URL is actually just a convention.

Many organizations in the past started following conventions based on the services provided by the web server. For example a webserver was often given "www" prefix; for a news server, "news" or "nntp"; for an FTP server, "ftp", etc.

Apparently, most companies name the computer on which their web site is stored "www", but they don't have to. The use of "www" thus is not bound by any technical policy as such and many websites don't use it.

For example, the course website is at *is1500.weebly.com/index.html*

Top-Level Domain (TLD)

TLD (Top Level Domain) refers to the suffix added to the domain names. For example, in the domain name 'www.google.com' the top-level domain is com (pronounced as 'dot' and the word that follows). The Internet Corporation for Assigned Names and Numbers (ICANN) is a non-profit organization responsible for managing most of the TLDs that exists today. Actually, ICANN manages the Internet Assigned Numbers Authority (IANA) which in turn oversees the whole IP address and name allocations. The IANA sends every IP allocation request to Regional Internet Registries (RIR), which are responsible for allocation addresses in different areas of the world. Similarly there are a lot of organizations, which work together to handle the millions of IP address allocations and maintenance work.

The list of TLDs is predefined and hence is limited. The most common ones are:

- com: For commercial businesses.
- gov: government agencies.
- edu: educational institutions.
- org: organizations which are mostly non-profit.

How do I get a TLD?

There are many domain registration services, which work with the domain registrars to provide registration service to the public.

You register your domain through any domain registration service that will then register it with ICANN and add it to the Domain Naming System (DNS) so that browsers can find it.

Well known domain registration services:

- GoDaddy
- Yahoo
- Weebly

How to find if a domain name is already in use?

A WHOIS service can be used to find out whether a domain names is already taken or not. WHOIS is one of the oldest terms in the context of Internet and it can provide all information related to a domain name.

Traditionally, WHOIS lookups were performed with a command line interface, but now a lot of web-based services have come up. For example 'http://www.whois.net/' searches for the domain name and tells the user if that domain name already exists or not. If it finds the domain name, it displays all the information related to it.

Long URLs and URL Shortening

Many times URLs get annoyingly long when the path or the file has a long name and is stored inside a long hierarchy of folders on the server. For example 'http://docs.oracle.com/cd/B19306_01/appdev.102/b14259/xdb01int.htm'. Thus, we can see that URLs can get hundreds of characters long and can also have complex patterns. URLs of such kind can get hard to remember and type.

URL Shortening

Long URLs can be shortened using a URL shortening service, which is a technique to shorten the URL in length and still direct to the same web page. URL shortening makes it easier to type, remember and share the URL. What the URL shortening does is that it redirects the HTTP request using the 'HTTP Redirect' on a shorter domain name, which then links to the web page which has the longer URL.

Some common URL shortening services are:

bitly (bit.ly domain)

- Google (goo.gl)
- TinyURL

How the HTTP works

As we discussed above, we know that the HTTP is a protocol, which defines the rules by which the client browsers and web servers exchange data. But how does it actually work?

HTTP is simple a request-response model. A browser initiates a communication by sending a request to the server. The request contains details like:

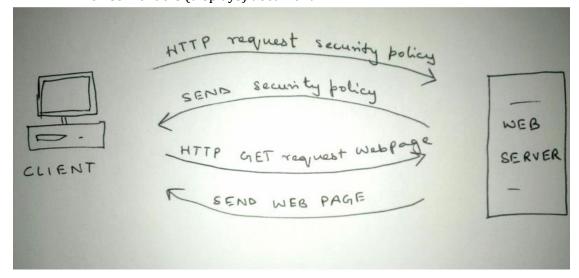
- A request line: This comprises of a method, a request-URL and the HTTP version of the request.
- A set of request headers: This has information like details about the connection, client, etc.

The server will respond to such requests with a response which will have details like:

- A response line: This consists of the HTTP version of the response, a 3-digit status code and a short description of that status code.
- As set of response headers: This has details about the response content e.eg type, length of the content and the last modified date & time.

To summarize, we can consider the whole HTTP request and response model in these simple steps:

- User selects a document by specifying its URL to the browser.
- The web browser formulates an HTTP request for that document and sends it to the web server specified in the URL.
- Server looks for the document or runs the program to generate the document.
- Server sends document back to browser.
- Browser renders (displays) document.



Connecting to the Internet

Computers connect to the internet through a variety of networks:

- Dial-up: Slowest and cheapest amongst all other Internet connections.
 Connects through phone lines and uses a dial-up modem.
- ISDN & DSL: Both are digital transmission systems, DSL is much faster than ISDN. ISDN connections can have either 64 Kbps or 128 Kbps line, whereas a DSL connection ranges from 128 Kbps to 100 Mbps. ISDN is dial-up type of connection whereas DSL is an "always-on" connection.
- Cable: Connections through cable are much faster than dial-ups and could be faster than DSL as well. It is an "always-on" type of connection and is available only where the cable is available.
- Satellite: Faster than the dial-up connection, but the data is generally delayed and hence cannot be used for real-time applications like video conferencing or online video games. As the name suggests, it connects via the satellites orbiting the earth. It can be affected by bad weather. It can be used anywhere in the world since it doesn't use any kind of wires or cables.
- Mobile Networks 3G & 4G: Available for cell phones, laptops, tablets & computers. Provides wireless connection through the cellphone provider's 3G or 4G network. Connection speeds may differ depending on the network strength in that area.

How does Wi-Fi work?

A wireless network is very similar to a two-way radio communication. A device called a wireless router is required to set up the Wi-Fi network. This wireless router is connected to the Internet through any of the ways we discussed in the section above. This is how a computer connected to a Wi-Fi works:

- A computer's wireless adapter translate the digital data into radio signals and transmit it using it inbuilt antenna.
- The wireless router receives that signal and decodes it back into digital form. Then the router sends this information to the Internet.

This process works the same way in the reverse. This is how a two-way connection is established.

The only difference between Wi-Fi and other radio networks is that a Wi-Fi device transmits at frequencies of 2.5 or 5 GHz (Gigahertz). The higher the frequency, higher the amount of data that can be transmitted.

<figure on wireless network goes here>

FIGURE 1.1 WORKING OF WIFI NETWORK

Search Engines

We know that the Internet has millions of web pages with information on variety of topics. But how does a user know about what web pages have the information he/she requires and what is the URL to that web page. The answer is "Internet Search Engines". We all use Google on a daily basis to search for information on anything and everything. Google has become almost analogous to search engine these days. But how does an Internet Search engine work? The answer simply lies in the following steps it performs:

- First, every search engine has software called a 'spider', which searches the Internet and builds an indexed list of words found on the websites. This process is called 'web crawling'. The 'spider' starts with websites, which are heavily visited, indexes all the words on that page and then follows the links found on those pages. This way the spider travels through most of the websites and building a list of words. The search engines have huge data centers where they store all this indexed information.
- The next step is answering the search queries. When a user sends search request with one or many words to search from, the search engine searches its indexed database for matching words and returns the result in order of relevance. This is very important since the user needs the best resource on top. There are a lot of complex algorithms involving hundreds of factors to come up with the best search result.

Some top Internet search engines are Google, Yahoo, Bing and Dogpile.

Summary

Summary required here...

Planning a Website

Upon completion of this chapter, you will be able to:

- ✓ Plan your website using one the industry-certified methods
- ✓ Create design outputs like sketches and sitemaps
- ✓ Write usability test cases for any sample website

Content Planning

Before you start writing your web content, it's very crucial to plan your web site. And a good website planning should start with planning the content and how you allocate the content to pages. This is called the information architecture, or IA planning. Information architecture of a website simply means 'the hierarchy of its navigation and structure'. This is a very important step in the process of web development. The better the planning, the less amount of revisions required at the end. What this means is that if you miss anything due to poor planning, then corrections will have to be made in the later stages of the web development.

You can plan your website content in several ways:

- Card Sorting
- Content Inventories
- Paper and Sketch Boards
- Site Map Diagrams

We will look into each of these methods in detail.

Know The User!

Before we get into the different methodologies of content planning, it's important to know your user: Who will be visiting your website? What would they be trying to find in your website? Does the website have content relevant to these users? For what kind of tasks will the user be visiting my website? A good website planning should have answered all such questions in detail.

Use Case

To define the users and their tasks are best expressed using 'use cases'. Use case is a very common term in all fields of software development. It defines the users, their roles, the system (the website in our case) and defines all the interactions between the user roles and the system. Each use case is defined in a list of steps portraying the actions taken by the user and the response from the system.

Use cases are hence very important because they help in explaining the behavior of the system as a whole, and also help find loopholes and correct them. In a nutshell use cases can be considered as software modelling technique that defines what is to be implemented and how to resolve errors that might occur.

Use Cases and Mini Stories

How do we write a use case? Identify the three basic elements that make up a use case:

- *Actors*: they are the type of users that interact with the website.
- *System*: the functional requirements, which specify the system behavior.
- *Goals*: All use cases are mostly initiated by a user to achieve a goal describing the steps taken to accomplish those goals.

Unified Modelling Language (UML)

Use cases are structured using Unified Modelling Language (UML) and are represented using ovals. An actor's involvement in a use case is represented with a line joining the two. And the boxes around the use cases represent the boundary of the system.

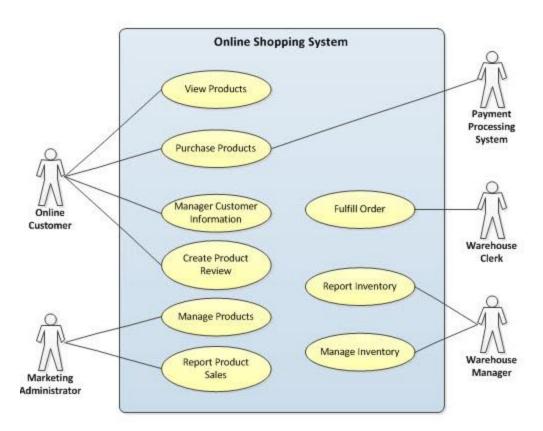


FIGURE 2.1 UML DIAGRAM FOR ONLINE SHOPPING SYSTEM

[IMAGE SOURCE:

 $\label{thm:local} HTTP://UPLOAD.WIKIMEDIA.ORG/WIKIPEDIA/COMMONS/0/07/ONLINE_SHOPPING_SYSTEM_USE_CASE_MODEL.JPG]$

After defining the users/actors and what they want to accomplish, write the use cases in the following format:

As a *<user>*, I want to *<goal>*, [so that *<purpose>*].

Examples:

- As an online customer, I want to view all the products so that I can select the best one available to buy.
- As a charter guest who has booked a charter, I want to get directions to the marina so that I know where to find my charter boat and can depart on time.

Content Planning with Card Sorting

Card sorting is a process of organizing topics/contents into categories and labelling them appropriately. It allows a website designer to organize all the content to be placed on a website in a hierarchical and an easy-to-manipulate manner. The contents, which have been identified, are written down onto simple index cards. The user groups (stakeholders of the website) are asked to sort the cards into stacks according to the hierarchy they think is logical. The card sorting technique is used mostly when there is great variety in the topics and/or the similarities among them make it difficult to classify them into groups or categories. But the users sometimes look at the content in a different way and thus have a different idea of grouping of items.

Hence, this process involves the end users to quite an extent. The whole process can be would involve the following steps:

- Create a card for each page. Each page represents a topic that has been identified.
- Write on the card what you want on the page. It can be either a word or a small phrase, whichever gives a clear idea about the functionality that page. The users are generally given pre-written cards, but they are also encouraged to write topics that they feel have been left out.
- Organize the cards into stacks. The next step is to arrange the cards in logical groupings. Also, find a name for each group. This process is generally repeated across a set of end users.
- When done, build a page flow diagram. The results from the sorting reveal a pattern, which can then be analyzed into forming a flow chart.

The sorting phase is mostly of two kinds: *open card sorting* and *closed card sorting*.

In **open card sorting**, the user groups are free to create their own logical groupings and name them accordingly. This type of sorting is used when the designers want to discover how the users classify the content.

In **closed card sorting**, the user groups are given pre-determined groups or categories. The users just have the task of assigning each card to any of those fixed categories. This kind of sorting is performed mostly to see to what degree do the categories effectively organize the data.



FIGURE 2.2 EXAMPLE OF CARD SORTING FOR A LIBRARY WESBITE

IMAGE SOURCE

HTTP://WWW.DOCTORDISRUPTION.COM/WP-CONTENT/UPLOADS/2014/02/CARD-SORT-STUDENT-04-ROUND-1-DETAIL.JPG]

The figure above was a part of the card sorting process done at the Ohio University to revamp their library website. [https://www.library.ohiou.edu/2012/07/asking-ourusers-card-sorting/]

Tracking Content with Inventories

Content Inventory

A content inventory is a spreadsheet of all your pages, the page hierarchy, and their file names along with what is on each page. It is a great way to understand the website in a whole and the motivation behind each page in the website. The spreadsheet gets more meaning with details like brief summaries of what the page does. The visual helps in getting a better understanding of the hierarchy of pages and what goes where. Content Inventories are great for tracking, but not so good for planning. This step should come after the planning stage when the pages are already sorted. Hence, use card sorting for the planning first and then track those using spreadsheets.

The best advantage of content inventories is that it makes it easy to switch contents and play around with alternate navigations. The topography of the website is clearly visible and thus makes it easy to find unnecessary and redundant pages on your website and finalize the site hierarchy.

The only disadvantage of using this for tracking is that it is a laborious task and takes quite an effort to maintain it later on when newer pages are added and the page hierarchy has to be changed.

Content Inventory Elements

Although inventories vary in what they capture, most of them include the following raw data for each piece of site content:

- Unique Content ID
- Title
- URL
- File Format (HTML, PDF, DOC, TXT...)
- Author or Provider
- Actual location of file
- Meta Description and Keywords
- Categories/ Tags
- Dates (created, revised, accessed)

GVS Content Inventory

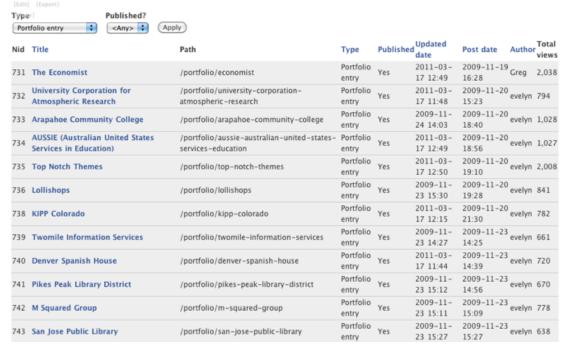


FIGURE 2.3 CONTENT INVENTORY EXAMPLE

[IMAGE SOURCE

 $\label{log:content-inventory-and-content-audit-views.} HTTP://GROWINGVENTURESOLUTIONS.COM/BLOG/CONTENT-INVENTORY-AND-CONTENT-AUDIT-VIEWS.HTML]$

As we can see from the example above, all the proposed web pages (planned in the card sorting phase) can be tracked with spreadsheets. The columns in such spreadsheets

represent raw information about the site contents. This particular example uses a content management tool called Drupal [https://www.drupal.org/home].

Paper and Sketchboards

Pictures are always louder and clearer than words. That stands true even for website planning. Sketchboards are low-tech and cheaper way for designers to evaluate possible page layouts and interactions with the user.

It starts with a clean sheet of paper that acts as a canvas to share, explore and iterate on various ideas. It's easy to write on it and carry it anywhere. On this paper we can scribble down ideas and organize them to make solutions. It's free of any format restrictions and lets the designers write and draw their thoughts, and it is fast. It is the fastest way to sketch out the contents and the user interactions collaboratively. Having said all that, using this technique designers more often can wander off from the main objective. The designer team has to make real efforts to write down ideas which are productive and lead them towards a sketch board with information appropriate for the website.

Although it is fast and cheap, the main disadvantage is that once the sketch is complete, it has to be translated into another useful format. In today's digital world, we all know the disadvantages of having something important on papers. It is easy to lose, cannot be reused, and most importantly it is not useful in the later parts of the development process. This is because everything has to be in digital form in the end to be useful in website development.



FIGURE 2.4 KRISTIN WEMMER, STARTING OUT ORGANIZED, SMASHING MAGAZINE, MARCH 17TH, 2010 [IMAGE SOURCE: HTTP://WIREFRAMES.LINOWSKI.CA/2009/05/SKETCHBOARDING/]

Many wireframing and mockup softwares have come up to help the designers to move ahead of the paper-and-pencil technique. Some common applications, which are quite user-friendly and can be used to build mockups are:

- Balsamiq
- LucidChart
- Gliffy
- Visio
- Moqups

Many people use wireframes and mockups synonymously, but they aren't. Wireframes are very rudimentary and shows only a skeleton of a webpage or an interface. They don't contain details about the contents. It main use is to understand the structure of a website. It is nowhere near to how the website will finally look. On the other hand mockups are closer to the end result. A website mockup looks somewhat like the final website, but it still doesn't have any functionality beyond what the user can see.

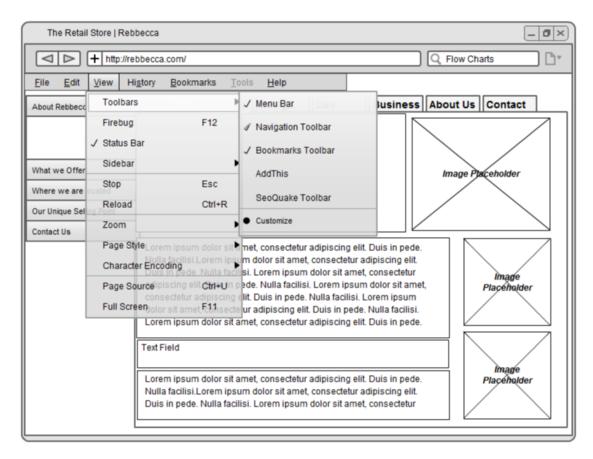


FIGURE 2.5 EXAMPLE OF A WIREFRAME

[IMAGE SOURCE: HTTP://CREATELY.COM/DIAGRAM/GFQHGT0C5/X6ENNBMJ7XC4JBM3C0LBAFQ8ZW=]

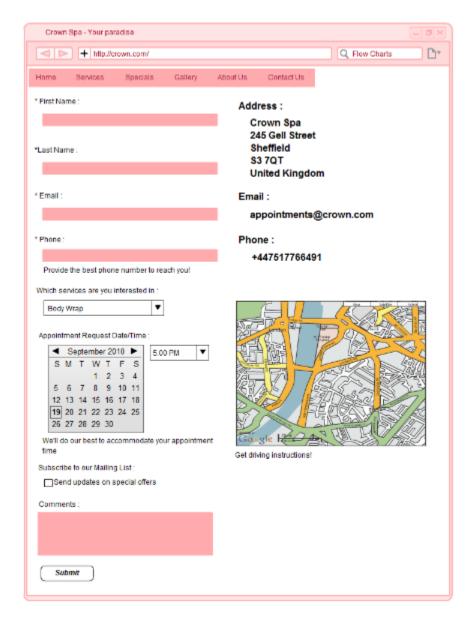


FIGURE 2.6 EXAMPLE OF A MOCKUP

[IMAGE SOURCE: HTTP://CREATELY.COM/DIAGRAM/GFQHGT0C2/93JAB3K4HAWRNMWFKUIRN5EG8=]

Site Map Diagrams

Site Map Diagrams are exactly what they sound like. They are nothing but visual representation of the website content outline and structure. In simple words, it is a diagram to shoe the hierarchical structure of the all the web pages in the website. It depicts the page flow i.e. how users get from one page to another. It is yet another visual based website planning tool which majorly concentrates on the following parts of the site architecture:

Structure

- Navigation
- Page hierarchy

Site Maps are created by designers to help the end users (stakeholders) visualize what has been planned. The end users generally don't understand the nuances of design and programming. Hence, site maps are a good way to engage the user and get feedback. The whole process of creating site maps can be summarized in the following steps:

- PLAN the logical presentation
- VISUALIZE and create the page flows (paths)
- ORGANIZE the content representation to assist the users achieve their goals.

The whole idea is to be able to create something that directs the users and encourage them to get involved with the website's structure and flow.

The major advantage of this technique is that it is a simple diagram with page names and plain arrows indicating the relationship/flow between the pages. Hence, it is easy for the users to understand it. Also, it greatly helps in detecting "dead ends" and "orphan pages" as well as the number of clicks or link navigations necessary to reach some specific piece of information.

The disadvantage of using this method is that the page flows don't actually capture the relationship between the pages. It doesn't have information about how does a user navigate from one page to another, and what the pages actually do. So, it doesn't give the user the whole picture.

Sitemaps can be sketched by hand on a sheet of paper and/or using sticky notes. They can also are created using various software tools such as:

- Microsoft Visio
- OmniGraffle
- Dreamweaver
- Fireworks, etc.

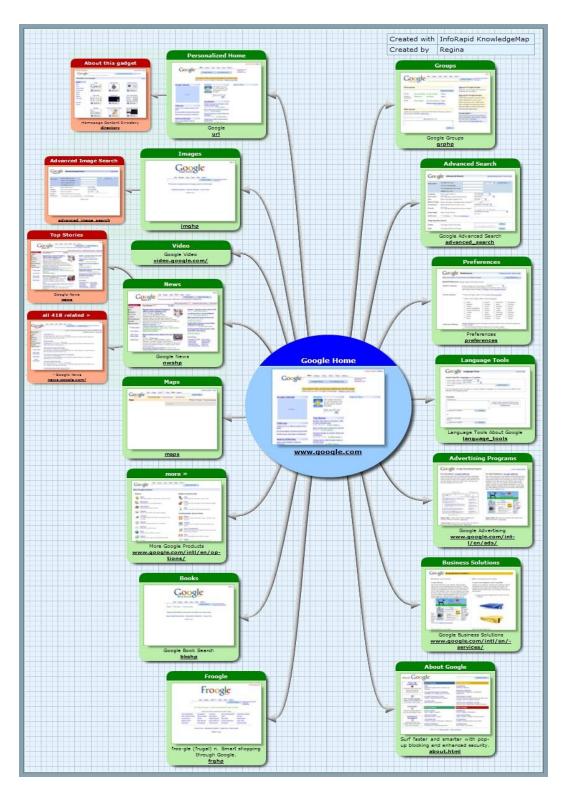


FIGURE 2.7 SITEMAP OF GOOGLE

[IMAGE SOURCE: HTTP://UPLOAD.WIKIMEDIA.ORG/WIKIPEDIA/COMMONS/2/20/SITEMAP_GOOGLE.JPG]

Iterative Design

All these methods we have discussed above will help the designers to create a design and present the user with a visual, which can be evaluated. Many times the users are not exactly sure what they have in mind. They may not always like the end product even though they said during initial meetings that they wanted something like the end product. The developers will have to do a lot of re-work in such cases.

To avoid wastage of work and time, like all software, websites are also built in an iterative fashion involving the client. Iterative design is a design methodology, which involves a cycle of prototyping, testing and refining the products. Designers move ahead only if the users are satisfied with the output till now. This is mostly done in fixed intervals of deadlines. A time box of may be 1 or 2 weeks is agreed upon, work is planned and then the designers build it. After the deadline, designers meet with the client (end users or stakeholders) and get feedback on the work done in that interval. The work for the next time box is also planned before leaving, and might include enhancements/corrections of the work done in last time box. This cyclic process is repeated until the client is satisfied with the prototype.

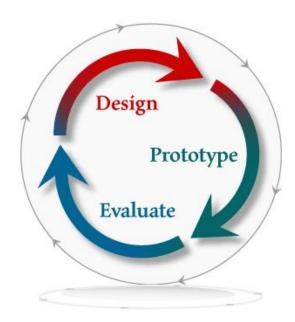


FIGURE 2.8 ITERATIVE DESIGN

[IMAGE SOURCE: HTTP://LEADSPEAK.FILES.WORDPRESS.COM/2013/01/CCD-ITERATIVE.PNG]

Usability Testing

Usability testing is a testing technique to evaluate a product by testing on users. It is the most important part of the Iterative Design that we discussed above. There is not doubt on why this is such an important part of the whole development process. It's an

irreplaceable part because it gets output directly from the client and gives the developers an insight to how the end users actually use the website. It actually lets the design and development teams to identify issues even before they are coded.

It is the next step after we have finalized the designs at each stage of the Iterative process. Usability testing may involve the following steps:

- Do all links (clickable links to redirect to other pages of the website) work?
- Can user get to the information they need, i.e., are your use cases supported?
- How long does it take to get common tasks done?
- How many mistakes do users make or how often do they get into a "dead end"?

Usability testing can also be classified into different types. Some common ones are:

- **Hallway testing:** instead of using trained testers, 5-6 random people are brought in to test the product. It is mostly effective during the early stages of development.
- **Remote Usability Testing:** In this type of testing, the testers, developers and the end users are located in different cities or may be countries and time zones. It can be done synchronously (video conferencing) or asynchronously (tester and user work separately).
- **Expert Review:** Experts in the field of testing are brought in to evaluate the product.
- Automated Expert Review: Very similar to expert review. Only difference is that instead of expert professionals, programs are used to assess the product. This approach has been a big project for the artificial intelligence community for some time now.
- **A/B Testing:** This type of testing is best used in the website development industry. Two versions A & B are compared, which are mostly identical with one or two changes that might impact the user's response.

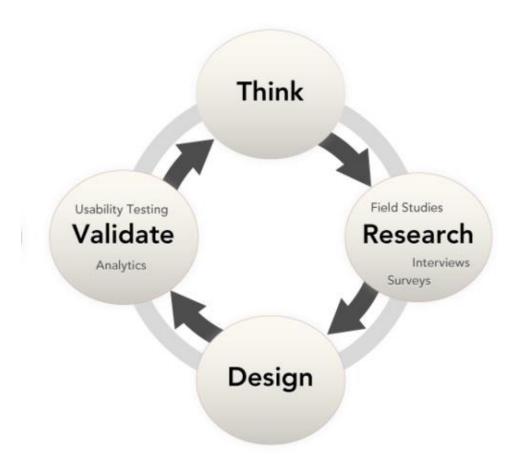


FIGURE 2.9 ROLE OF USABILITY TESTING IN ITERATIVE DESIGN

[IMAGE SOURCE:

HTTP://BASESPACE.FILES.WORDPRESS.COM/2012/08/BASESPACEUXPROCESS1.JPG?W=519&H=463]

Summary

Now that we have gone through all the methods, which one do you think is the best way to plan a website? There is no one right answer to this. Depending on the requirements, we might use one or all of the techniques together. Some attributes which effect this decision are size, type of website, how tech-savvy the client is, etc.

Using Markup Languages: HTML & Markdown

Upon completion of this chapter, you will be able to:

- ✓ Understand the basic structure of HTML document
- ✓ Understand the various elements which together form the HTML page
- ✓ Get an insight into CSS and how it works with the HTML tags
- Understand the new features of HTML5

What is HTML?

As discussed in the first chapter, HTML is a language for formatting (describing contents) web pages. HTML stands for Hyper Text Markup Language. As the name suggests, HTML is a markup language not a programming language. A markup language is a set of markup tags or called the HTML tags, which constitute an HTML document. The purposes of the tags are to describe the layout and formatting of page content. In simple words, it is used to edit and position the contents like text, image, frame, etc.

Role of HTML

The web server generates the web page as HTML and sends it to the browser for display. The web browser renders it and displays it for the user. All web browsers allow the users to view the source of any web page they visit.

Most of the web browsers show the source HTML code when you press "Ctrl + U", but not all browsers respond to that command.

Internet Explorer: Ctrl + U

Mozilla Firefox & Netscape: Ctrl + U

■ Google chrome: Ctrl + U

Opera: Ctrl + F3

Apart from keyboard shortcuts, you can always right click & find the *view source* option. Browsers like Google Chrome have a very good inspecting tool. When you click right on an element and do inspect element. You can edit the HTML on the fly, test it there and then later add it in the code. It's very helpful when it comes to complex web applications.

BoatVentures (Schedlbauer)

Call +1 888 555 5555 to book

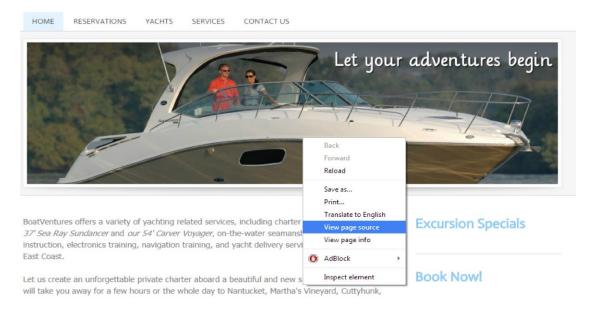


FIGURE 3.11 HOW TO SEE HTML SOURCE OF A WEBPAGE

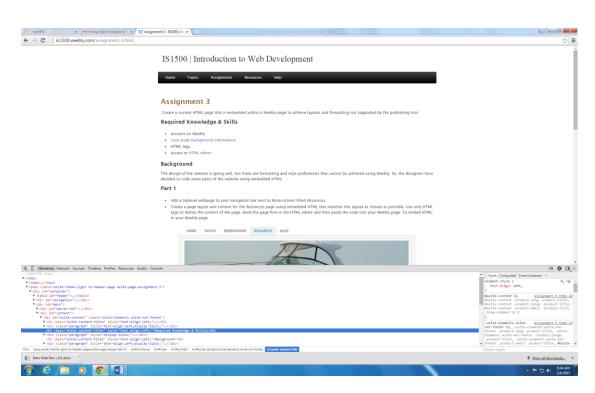


FIGURE 3.12 HOW TO SEE INSPECT ELEMENT PAGE

Here, we can a screenshot of the webpage http://is1500profs.weebly.com/. Like we discussed, when we select the "view page source", the web browser opens the HTML document, which it receives from the server (Figure 3.2 shows a part of the HTML). Go ahead and look up HTMLs of a few websites.

FIGURE 3.2 HTML SOURCE OF A WEBPAGE

HTML Markup Tags

As we have already seen a few examples of HTML code, we have observed that it's mostly written with some keywords surrounded by angular brackets. These are HTML markup tags and the keywords are the tag names. The HTML tags normally come in pairs like and (this tag renders the text enclosed in it as bold). The first tag in a pair is the start tag (or opening tag); the second tag is the end tag (or closing tag). The end tag is always written like the start tag, but with a forward slash before the tag name. We will look at a few tags in the next section and see what those tags mean.

A Simple HTML Page

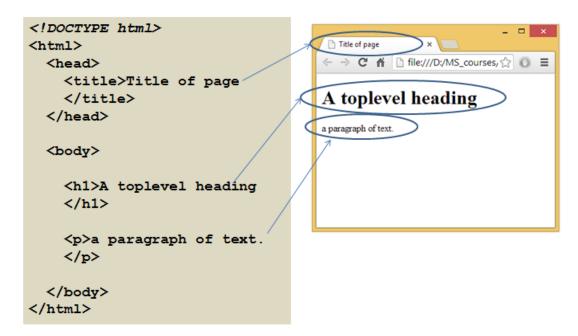


FIGURE 3.3 A SIMPLE HTML PAGE

The figure above shows a basic HTML page with the minimum number of tags. The first line is not an HTML tag. It's an instruction for web browsers about the version of HTML the following document is written in. This is a syntax of HTML5. This helps in rendering the document properly. Also, this declaration has to be always the first line in the HTML document. For HTML version like HTML4.01 other than HTML5, the first line is written like this:

```
<!DOCTYPE HTML PUBLIC "-//..//DTD HTML 4.01//EN" "http://...">
```

The next tag is the HTML tag. This is the tag at the highest level (root) i.e. every other tag has to be written inside it. There can be only one HTML tag per HTML page. At the next level, there are generally two tags namely <head> and <body>.

Head tag

The <head> element acts as a container for all head elements. This is the place where we generally prefer adding the external style sheets.

Title tag

The <title> tag is the only tag, which is a required tag inside a <head> element. Other optional tags are:

- <style> (to define style related information)
- <base> (base URL for all relative URLs in the document)
- (relationship between the document & an external source like css file)

- <meta> (meta data about the document which do not get displayed)
- <script> (used to define client-side scripts like JavaScript)
- <noscript> (alternate content for users who have disabled scripts on browser)

Body tag

The <body> tag defines the document's body i.e. the main contents being displayed. It acts as a container for all other html tags. We will look into the ones, which will help you understand and build simple HTML documents. Like HTML tag, Body tag can also be written one time per page.

Headings

Headings are used to provide titles to page sections. There are six heading levels: <h1> through <h6>. <h1> is for the most important heading and <h6> is for the least. The browser determines the default formatting and font style used for the headings. It also automatically adds empty space before and after each heading. Formatting can be changed using style sheets (described using CSS). As we can see in figure 3.4, the <h1> tag is by default the biggest in size and boldness and <h6> is the least. Heading should only be used for heading only and not for any other formatting.

h1 level heading h2 level heading h3 level heading h4 level heading h5 level heading h6 level heading

FIGURE 3.4 USE OF HEADING INSIDE <BODY> TAG

Paragraphs

Paragraphs are sections of content with whitespace after the content. It is similar to writing a paragraph from book. It starts at a new line and ends with a line break. Similarly, browsers automatically add empty line before and after every tag.

a paragraph of text.another paragraph.

a paragraph of text.
another paragraph.

FIGURE 3.5 EXAMPLE OF <P> TAG

Mostly, browsers will display the contents correctly even if you miss adding the end tag i.e. . But it's advisable to always have the end tags in place. This might cause problems when and if the HTML is parsed as strict versions like the XHTML (Extensible HTML).

You can also add text align attributes to the text in the paragraph but this is not supported in HTML5.

Most browsers will display the element with the following default values:

Example

```
p {
    display: block;
    margin-top: 1em;
    margin-bottom: 1em;
    margin-left: 0;
    margin-right: 0;
}
```

Pre tag

Browsers generally removes any extra space and lines within the tags. But sometimes, we need a paragraph to be displayed the way we write it. The tag is used in such cases. This tag displays the content as it is. So to display content with spaces & line breaks, we have to wrap it in a tag.

```
This paragraph contains a lot of spaces in the source code, but the browser ignores it.
```

FIGURE 3.6.1 DIFFERENCE BETWEEN <P> AND <PRE> TAGS

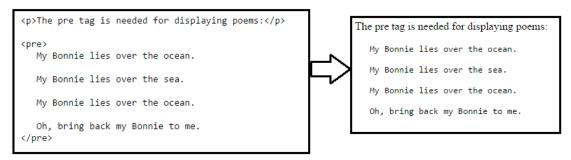


FIGURE 3.6.2 DIFFERENCE BETWEEN <P> AND <PRE> TAGS

The two examples above explain how the browser cancels the extra spaces & line breaks when rendering a tag, but doesn't do so while rendering content inside a tag.

This also have an attribute "width" which is not supported in HTML5.

Most browsers will display the element with the following default values:

```
Example
```

Formatting

HTML uses tags like and <i> for formatting output, like **bold** or *italic* text. Hence these elements are called as formatting elements or tags. Browsers treat and <e> same as and <i>, but they are don not have the same meaning. The and <e> tags are meant to elevate the importance of the text within them. It has a different

meaning than bold and italic when the HTML is rendered on a mobile or any handheld device. For the sake of this course, we will just stick with and <i> for now.

```
<b>This text is bold</b><u>This text is underlined</u><mall>This text is small</small><strong>This text is strong</strong><i>>This text is italic</i>
```

- Bold
- Underline
- Small
- Strong
- Italics
- · Emphasis

This text is bold

This text is underlined

This text is small

This text is strong

This text is italic

This text is emphasized

FIGURE 3.7.1 TAGS FOR FORMATTING TEXT INN HTML

```
<code>This is computer output</code>
This is<sub> subscript</sub> and
<sup>superscript
<ins>This text is inserted</ins>
<del>This text is deleted</del>
<mark>This text is marked</mark>
```

- Code
- Sub
- Super
- Ins
- Del
- Mark

This is computer output

This is subscript and superscript

This text is inserted

This text is deleted

This text is marked

FIGURE 3.7.2 TAGS FOR FORMATTING TEXT INN HTML

Figures 3.7.1 and 3.7.2 demonstrate the different types of formatting tags available. These help in displaying text in different formats and helps in segregating different elements in text. For example while displaying some mathematical formula, we might need <sup> and <sub> to make it look like a formula. The <ins> and tags are generally used by programmers in websites made for editing documents.

Nesting HMTL Tags

HTML allows us to nest tags within each other. The web browsers mostly are very lenient and might still display the content properly, but tags must be nested properly. The first example in Figure 3.8 is a good example of nesting where the start and closing tags of and are enclosed inside the start and closing tags of and .

FIGURE 3.8 NESTING IN HTML

The 2^{nd} example in figure 3.8 works because the browser is lenient. But we should never depend on the browser to correct it for us. It might produce unexpected results and/or errors if we do not nest tags properly.

Horizontal Lines

Horizontal lines can be added in a HTML document as a visual spacer. This is achieved by using the <hr/>tag. The <hr/>tag generally used to separate the content (e.g. change of topic) and to define a thematic break (HTML5) in an HTML document. Since the horizontal lines have no content, the </hr>
tag is a self-enclosing tag. There is no such thing as a vertical line in HTML. That can be achieved by CSS. For example one way is to define a <div> element and then define its border properties in the CSS. We will learn about CSS later in this chapter.

```
This is line one.

<hr />
This is line two.

This is line two.
```

FIGURE 3.9 EXAMPLE OF <HR> TAG IN HTML

Line Breaks

We can also start a new line of text by using the self-enclosing
br/> tag. Like the <hr/>tag the
tag the
tag doesn't have any closing tag. As we have discussed above, whitespace, spaces and line formatting is ignored by the browser. But if we need to add extra spaces, then we can use the element to add a single space. can be used multiple times as well to have multiple spaces together.

This is the same line
br />divided by and a line break.

This is the same line divided by and a line break.

FIGURE 3.10 EXAMPLE OF
 TAG AND &NBSP; IN HTML

Links

We can see links on nearly all websites. Links allow users to navigate from one page to another. Links in terms of HTML are hyperlinks. Hyperlinks could be either some text or even an image, which a user can click on and move to another page. Such links to other pages are defined using the anchor tag <a>>.

```
Link to <a
href="http://www.w3schools.com"> W3Schools
HTML Tutorial</a>
```

Link to W3Schools HTML Tutorial

FIGURE 3.11.1 EXAMPLE OF <A> TAG WITH TEXT IN HTML

As we can see above, the <a> tag encloses some text within it and that text is shown as a link on the browser. The "href" attribute of <a> tag indicates the link URL i.e. the destination address. Hence, clicking on that text on the browser will direct the browser to that web page. By this feature, you can also navigate from top to bottom just on a click in case of HTML page where you have to scroll through the page. For example in the figure above it will direct the browser to "http://www.w3schools.com".If you like, you can use this link tag in the form of button also and many other forms. One of the examples are:

<button> W3Schools HTML Tutorial
</button>

Opening a New Window/Tab

By default, the <a> tag redirects the same browser tab/window to the destination address. But if you want to open a linked resource in a new window or tab, use the *target* attribute of the <a> tag. Set the target to "_blank". This simply means as a command to the browser to open this link on a new blank tab.

```
<a href="http://www.cnn.com" target="_blank">
News of the Day
</a>
```

News of the Day

FIGURE 3.11.2 EXAMPLE OF <A> TAG WITH TARGET ATTRIBUTE IN HTML

We can see above that it looks like a normal link, but it opens the CNN webpage in a new window or tab.

Links within a page

At times we need links, which direct the user to a different location on the same page. One very good example is the Wikipedia website. Every Wikipedia page has a content section which has links directing to the different sections somewhere below on the same web page. These links can also be defined using the <a> tag, but in a different way.

First, add an 'id' attribute to the <a> tag which you want to be the destination section. Then define link to this <a> by specifying the 'href' as '#id'. For example in figure 3.12 we have defined 'id' as *tips* & the 'href' of the link as *#tips*.

```
<a id="tips">Useful
Tips Section</a>
Tip 1
Tip 2
<br />
<a href="#tips">Visit
the Useful Tips
Section</a>
```

```
Useful Tips Section

Tip 1

Tip 2

<u>Visit the Useful Tips Section</u>
```

FIGURE 3.12 USING THE ID ATTRIBUTE TO NAVIGATE WITHIN SAME PAGE

Images

Images can be defined in HTML using the tag. The image tag is an empty tag i.e. it only contains the attributes like width, height, etc. Hence it doesn't have an ending tag. The 'url' attribute defines the source of the image. It can be an image on your local hard disk or even an image link from the internet.

Some tips while using the tag are:

Always define the width and height attributes. If they are not provided then
the image is displayed in its original full size. Also, the browser takes a longer

time to render the image. Although we can specify them using the width and height attribute of the image tag, but it's better to use the style attribute instead.

 Define the 'alt' attribute. This specifies what alternate text will be displayed in place of the image if it cannot be displayed (mostly when it can't be located by the provided 'src').

```
<img src="http://www.w3.org/html/logo/downloads/HTML5_Badge_512.png"
alt="HTML5 Icon" style="width:128px;height:128px"/>

HTML5 Icon

<img src="HTML5_Badge_512.png"
alt="HTML5 Icon" style="width:128px;height:128px"/>
```

FIGURE 3.13 THE IMAGE TAG

Image Links

Many times we need to use an image as a link. This can be done by enclosing the image tag within an anchor tag. In the figure below we can see that the <a> tag is pointing to 'www.w3schools.com' but has a tag within it in place of text.

```
<a href="http://www.w3schools.com">
<img src="smiley.gif" width="100"
height="100" /></a>
```

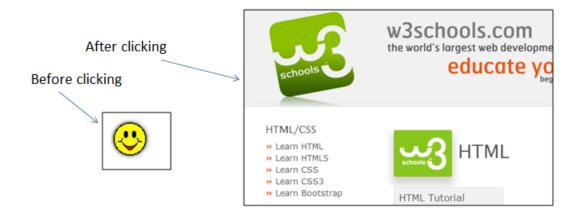


FIGURE 3.14 THE IMAGE TAG AS A LINK

Lists

HTML provides a wide array of lists. Those are unordered lists, ordered lists and the HTML description lists. The list items in all of these lists can contain other HTML elements such as images, links, etc.

Unordered list

An unordered list starts with the tag. And each list item is defined using the tag. Each list item is marked with a bullet. But the style attribute of the tag can be used to define the markers.

```
CoffeeTeaMilk
```

- Coffee
- Tea
- Milk

FIGURE 3.15 UNORDERED HTML LIST

Ordered list

It is similar to the unordered list. The only difference in the syntax is that is used instead of . By default the markers are numbers.

```
    Coffee
    Tea
    Milk
```

- 1. Coffee
- 2. Tea
- Milk

FIGURE 3.16.1 DEFAULT ORDERED HTML LIST

The marker can be changed by just using the 'type' attribute. It can hold values such as '1', 'A', 'a', 'I' and 'I'. The value '1' is the default type value. Below are some examples.

FIGURE 3.16.2 ORDERED HTML LIST OF TYPE UPPER CASE & LOWER CASE

```
    Coffee

 Coffee
                II. Tea
 Tea
                III. Milk
 Milk
i. Coffee
 Coffee
                ii. Tea
 Tea
                iii. Milk
 Milk
```

FIGURE 3.16.3 ORDERED HTML LISTOF TYPE ROMAN UPPER CASE & ROMAN LOWER CASE

Description Lists

Many times we need to add description to the list items. The description HTML list allows us to do that. Here the <dl> tag defines the list, the <dt> tag is for each list item and the <dd> tag is used to define the description of each list item.

```
<dl>
<dt>Coffee</dt>
<dd>- black hot
drink</dd>
<dt>Milk</dt>
<dd>- white cold
drink</dd>
</dl>
```

```
Coffee
- black hot drink
Milk
- white cold drink
```

FIGURE 3.17 DESCRIPTION HTML LIST

Nested Lists

All the HTML lists can be nested inside of another list. For example, in the figure below an ordered list of type '1' is nested inside an unordered list. It also works vice versa.

```
    Coffee
    Tea

            Black tea
            Green tea
            Milk
```

- Coffee
- Tea
 - Black tea
 - Green tea
- Milk

FIGURE 3.18 NESTED HTML LIST

Blocks of Text

Texts are mostly placed into blocks for special formatting. These can be classified into block level elements and inline elements. The block elements generally start and end with a new line (when displayed on a browser). The inline elements are displayed without any line breaks. The most common elements which come under these two categories are <div> and .

<div>

This is a block level element which acts as a container for other HTML elements (tags). The <div> tag in itself doesn't have any importance and doesn't have any attributes other than style, class & id, which are common for any tag in HTML. This tag is used to mostly segregate different sections of content on a web page.

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

FIGURE 3.19 EXAMPLE OF DIV

This is an inline element which is generally used a container for text content. Similar to the <div> element, the tag also doesn't have any attributes for itself. But unlike <div> element, the elements do not have any default special formatting for line breaks. In the example below we can see that the span element is displayed in the same line as the <h4> element.

<h4>My ImportantHeading</h4>



My ImportantHeading

FIGURE 3.20 EXAMPLE OF SPAN

Tables

Tables are like spreadsheets with rows and columns. They are defined using the tag are the tags which define the rows of the table. And each row can have table data which are defined with the . The actual data goes inside the tag.

```
    row 1, cell 1
  </t>
  <t >row 1, cell 2
  </t>
  </t>
  </t>
  </t>

    row 2, cell 1

  row 2, cell 2

  row 2, cell 2

    row 2, cell 2

    row 2, cell 2

    row 2, cell 2

    row 2, cell 2

    row 2, cell 2

    row 2, cell 2

    row 2, cell 2

    row 2, cell 2

    row 3, cell 2

    row 4, cell 2

    row 5, cell 2

    row 6, cell 2

    row 6, cell 2

    row 7, cell 2

    row 6, cell 2

    row 7, cell 2

    row 6, cell 2

    row 7, cell 2

    row 8, cell 2

    row 9, cell 2

    row
```

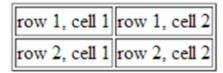


FIGURE 3.21 CREATING TABLE WITH <TABLE> TAG

In the figure above, we can see an example of a simple table with two rows, and each row containing two cells each. There are many attributes through which we can change the way the table looks on a browser.

Table border

One such attribute can be seen in figure 3.21 (i.e. the 'border' attribute). The border attribute defines that a border will be displayed around the table & the value specifies the thickness of it. For example the value '1' in the figure above means a thin line.

Cell spacing

Cell padding refers to the space between the content inside a cell and its borders. By default there is no padding in the table cells (tag). In figure 3.22 the *cellpadding* attribute is used & we can see that it produces spaces all around the text inside every cell in the table.

HTML & Markdown

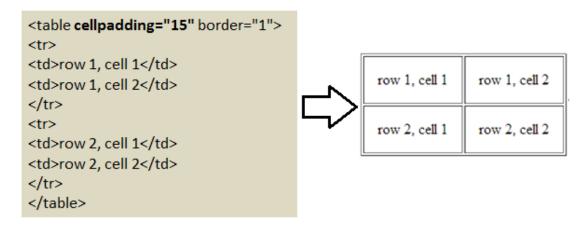


FIGURE 3.22 CELL PADDING

Cell spacing

Cell spacing defines the space between the borders of all cells. By default there is no cell spacing. In figure 3.23, the *cellspacing* attribute is used & we can see that it produces spaces between the cells. The value assigned to *cellspacing* attribute defines how big the space would be when displayed on a browser.

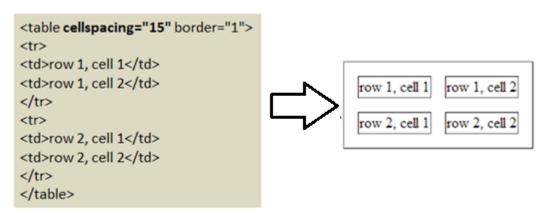


FIGURE 3.23 CELL SPACING

Table Headers

Table headers are used to define the headings for a table. It is defined using the tag. The text or image inside the tag is displayed as a header on the browser. The text inside a header tag is by default displayed in bold. We can always change the default styling using CSS.

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

FIGURE 3.24 TABLE HEADER

Table cell spans

Many times we need to specify cells which span across multiple cells. HTML tables do have this option. There are two types of spans, row span and column span. Both of the spans can be used individually or together depending on what we need.

Table Row Spans

As the name suggests, the rowspan attribute makes the cell span across multiple rows. The value defines the number of rows that particular cell will merge. In figure 3.25, the rowspan is applied to a tag containing text 'Telephone:', and the browser renders it in a way that now that cell spans across two rows.

```
            First Name:
        Hill Gates

        Ath>First Name:

        Ath>First Name:

        Ath>First Name:

        Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath>First Name:

            Ath First Name:
            Ath First Name:

            Ath First Name:
            Ath First Name:

            Ath First Name:
            Ath First Name:
            Ath First Name:

            Ath First Name:
            Ath First Name:
            Ath First Name:

            Ath First Name:
            Ath First Name:
            Ath First Name:
            Ath First Name:
            Ath First Name:
            Ath First Name:
            Ath First Name:
```

```
First Name: Bill Gates
Telephone: 555 77 854
555 77 855
```

FIGURE 3.25 ROW SPAN

Table Column Spans

Similar to rowspan, the column span allows a cell to span across multiple columns. The value defines the number the columns the cell will merge. In figure 3.26, the colspan attribute is applied to a tag containing text 'Telephone', and the browser renders it in a way that now that cell spans across two columns.

```
        \table border="1">
        \table border="1">
```

Name	Telephone	
Bill Gates	555 77 854 555 77 855	

FIGURE 3.26 COLUMN SPAN

Tables with CSS

The default styles of table elements can be modified using CSS. For instance, we can change the border width and color; the column width and height (same for table);

alignment of text inside cells; the background, etc. Below are some examples which show the simple modifications that can be made to the table's appearance. Note that all the CSS modifications shown in the examples below will be applied to every table in that document. But there are ways where we can assign CSS styles to particular elements as well. We will learn about that later.

Note: values entered as px (pixel) and % (percentage) differ. Pixel values don't change with the browser's current size, but elements styled with percentage values resize according to the browser size.

```
table {
   border-collapse: collapse;
}
table, th, td {
   border: 1px solid black;
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

FIGURE 3.27.1 TABLE WITH CSS

In figure 3.27.1, first the table's border has been removed & then a single line border of 1px (pixel) is added for tables, headers and all data cells.

```
table {
    width: 100%;
}
th {
    height: 50px;
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

FIGURE 3.27.2 TABLE WITH CSS

In the figure 3.27.2, the width is set to 100%, which means the table's width will expand to the current browser's width. Also we can see that the header row is set to 50px (pixels), which means that it's height will be fixed at 50 pixels.

```
td {
   height: 50px;
   vertical-align: bottom;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150

FIGURE 3.27.3 TABLE WITH CSS

In figure 3.27.3, along with the height, the cell content is vertically aligned to bottom. By default the alignment is middle.

```
td {
   padding: 15px;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150

FIGURE 3.27.4 TABLE WITH CSS

In figure 3.27.4, the padding for the data cells is set to 15 pixels. This adds space between the text and the cell borders.

```
table, td, th {
   border: 1px solid green;
}

th {
   background-color: green;
   color: white;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

FIGURE 3.27.5 TABLE WITH CSS

In figure 3.27.5, along with border width of 1 pixel the color is also set to green. And then for the header row, the background is set to green and the text color is set to white. The default text color is black & the default background color is white.

Aligning Text within Tables

To create a menu style header, where one part is left aligned and the other right, place the text into a table. Note that by default a table uses less than the width of the page. Use the attribute *width="100%"* in your tag.

FIGURE 3.28 ALIGN TEXT INSIDE TABLE ELEMENT

Arranging Items in Columns

HTML5 has a different way of arranging items in columns instead of a table. The different <div> elements are stacked in columns in HTML5. As seen in the figure below, we can achieve the table effect with the 'float' attribute of the <div> tag. What the float attribute does is that it stacks the div element with its content from the specified side of the browser. Similarly, in figure 3.28 both the div elements have a list inside them and the browser displays them side by side due to the float attribute. In absence of the float attribute, the div elements would have been displayed one after another vertically.

```
<div style="float: left; width: 50%;">

Left Item 1
Left Item 2
</div>
</div>
<div style="float: right; width: 50%;">

Right Item 1
Left Item 2
Right Item 1
Left Item 2
Right Item 2

* Left Item 1
* Right Item 2
* Right Item 2

* Right Item 2

* Right Item 2
```

FIGURE 3.29 FLOAT ATTRIBUTE OF HTML5

Styles with CSS

We have been talking about styling HTML elements using CSS. CSS is a very powerful language, which helps in representing the look and formatting of a document written in a markup language like HTML.

CSS i.e. styling can be added to HTML elements in three ways

- Inline styles
- Internal (embedded) styles
- External styles

We will look into each of these three ways one by one.

Inline Styles

Inline styles are placed directly in the tag i.e. styling is added using the style attribute inside a tag itself. This is a way uniquely styling a single HTML element. The figures below, styling has been applied using the style attribute of the <h1> and tags. Similarly, inline styling can be applied to various elements. Inline style is generally not recommended.

FIGURE 3.30.1 INLINE STYLING IN HTML

```
<!DOCTYPE html>
<html>
<body>
<h1 style="text-align:center;">Center-aligned heading</h1>
This is a paragraph.
</body>
</html>

A heading

A paragraph.
```

FIGURE 3.30.2 INLINE STYLING IN HTML

Internal Style Sheets

Internal (embedded) style sheet specifications are placed in the <head> section of the HTML document. It is used to define styling for one or more elements together. Using an internal style sheet, one can style the whole HTML document i.e. all the elements. In the figure below, styling has been applied on the <body> and the tag, which means all the <body> tag and all the tags in the HTML document will have this particular styling.

```
<head>
<style type="text/css">
body {background-color:yellow;}
p {color:blue;}
</style>
</head>

A heading

A paragraph.
```

FIGURE 3.31 INTERNAL STYLE SHEET IN HTML

External Style Sheets

External style sheet specifications are written in a separate document and then attached to various HTML documents. This is very important while developing a website because we would want to have the same styling for elements across all pages of our website. If

we use inline or internal styling, then we will have to repeat the same styling code on each and every HTML document. In the figure below, the styling has been defined in a separate file called "mystyle.css". The HTML document links to this CSS file using the link> tag enclosed in the <head> element.

FIGURE 3.32 EXTERNAL STYLE SHEET IN HTML

Custom Style with External Style Sheets

External style sheets make it much easier to manage your style definitions across multiple html pages. Here's how you link from your HTML to the style sheet. Note that the style sheet is expected in the same folder (directory) as the html page.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
link href="style.css" rel="stylesheet" />
<title>Boatventures</title>
</head>
<body>
<h1>BoatVentures</h1>
Simple paragraph text
</body>
</html>
```

STYLE SHEET FOR THE SAME:

```
body {
  background-color: #d0e4fe;
}
h1 {
  color: orange;
  text-align: center;
}

p {
  font-family: "Times New Roman";
  font-size: 20px;
}
```



The resulting web page looks like this. Note that the background has changed and that headings and paragraphs have new styles.

CSS Style formats

We have discussed quite some code snippets about CSS and how it works along with HTML and defines the styling for HTML documents. CSS stands for Cascading Style Sheets. But what does the term 'cascading' mean in the context of styling? HTML and CSS work in a way that it is possible that there can be multiple properties affecting the same HTML element. And some of these might conflict with each other. For example, one CSS defines the color for tag to be green and then another one defined it to be red. So which ones do the browser chose? This decision is made based on cascade. The whole idea behind cascading is a bit complex and beyond the requirements of this course. But it's always good to know how something is named.

CSS were first added to HTML version 4.0 to solve the whole problem of storing formatting information. CSS could be stored as separate ".css" files which saved a lot of work.

In the following sections, we will look at some of the commonly used styling

Background Styles

CSS can be used to define backgrounds for the HTML elements. Some examples of CSS properties for backgrounds are:

Changing the background color.

```
background-color: "red";
```

Changing the background image.

```
background-image: url("paper.gif")
```

• Repeating background image. This will repeat the image as many times are required to fill the element's background area. The default value is "repeat-x" i.e. repeat vertically. Similarly "repeat-y" will repeat the image horizontally. Use "no-repeat" when you don't want to repeat the background.

```
background-repeat: repeat-x;
```

 Specifying the background position. The default value for this property is "left top".

```
background-position: right top;
```

Try out these properties with different values to get a picture of what these do to the background of the elements.

Specifying Color

With CSS, a color is most often specified by one of the following methods:

 HEXADECIMAL value like "#ff0000", are supported by all browsers. The format is "#RRGGBB" where RR, GG & BB are hexadecimal integers which specify the red, green & blue components of the color. The values for each of these three colors can range from 00 to ff.

- RGB value like "rgb(255,0,0)", are also supported by all browsers. The RGB values specify the intensity of the colors red, green and blue. The values can range from either 0 to 255 or can be specified using percentage i.e. rgb(100%,0%,0%).
- Color name like "red". Around 140 color names are predefined in the HTML & CSS specifications.

For choosing a color and it's hex value, use http://www.colorschemer.com/online.html

Text Formatting

Text can be formatted using one of the following properties:

- Text Color: The color property can be used to set the color of the text. By setting the color of a tag, the text enclosed in it will take up that color. The color can be set using the "color" attribute.
- Text Alignment: Text inside a tag can be horizontally aligned by using the "text-align" property. The default alignment is "left", but it can take values like "center", "right" and "justify". These hold the same interpretation as the ones used in text editors like MS-word.
- Text Decoration: The "text-decoration" property can used to set or remove any decorations from text. To remove any decoration it's value should be set to "none". To set any decoration the "text-decoration" property can be set to "overline", "line-through" and "underline".
- Text Transformation: The "text-transform" property is used to change the case
 of the letters in the text. The three possible values are "lowercase", "uppercase"
 and "capitalize". The last value changes the first letter of every word in the text
 to upper case.

Fonts

We are all familiar will fonts, especially if you have worked on any text editor like MS-Word, etc. They are nothing but a set of character symbols that determine how the text will be displayed on a browser. Since HTML has lots of text enclosed in its elements, CSS has some properties related to fonts. Some common ones are:

• Font family: This can be defined using the "font-family" attribute. Many times some browsers can't support all the fonts. Hence, it is advisable to include multiple fonts so that the browser can fall back to other ones in case it doesn't support the first one. You can specify multiple fonts in a comma separated list. Also, font family names with multiple words should be defined within quotation marks e.g. "Times New Roman".

```
p {
    font-family: "Times New Roman", Arial, Helvetica;
}
```

 Font style: This is defined using the "font-style" property, and is mostly used for specify italic text. The three possible values are normal, italic & oblique. (Italic & oblique are quite similar to each other, but italic is the preferred one due to certain design limitations)

```
p {
    font-style: italic;
}
```

Font size: This defines the size of the text and is defined using the "font-size" property. The font size can be both absolute and relative. Absolute size is specified using px (pixel) values. HTML tags have a default font size. For example the default size of a tag is 16px. So we can specify font size relative to that. This is done using the "em", where 1em is equal to the current font size (default text size in browsers is 16 px). This also improves the adjustability of text size on browsers. The best way to define font sizes is to have default size defined in the body tag and then use "em" on other element styles.

```
p {
font-size: 14px;
}

p {
font-size: 0.875em; /* 14px/16=0.875em */
}
```

Till now we have seen how to make text inside a paragraph the same style attributes. What if we wanted different parts of a paragraph to be of different font, color or style? This can be achieved using the tag. Remember that the span tag is an inline element, thus it doesn't put a line break before & after. So in this figure below, we can enclose text inside which have a certain style defined.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, ...

FIGURE 3.33 USE OF TO CHANGE STYLE OF TEXT

```
<html>
<head>
<style>
body {
    font-size: 100%;
                                   A heading 1
h1 {
    font-size: 2.5em;
}
p {
    font-size: 0.875em;
                                   A paragraph.
</style>
</head>
<body>
    <h1>A heading 1</h1>
    A paragraph.
</body>
</html>
```

FIGURE 3.34 BEST WAY TO DEFINE FONT-SIZE IN CSS

Fonts, Spans and Styles

A font is a collection of character symbols that determine how the text looks when it is displayed. To change the font of some block of text, you use the *style="font-family:font-name"* attribute in the tag block that encloses the text. This will work for , <div>, and blocks.

Here's a block of text in a paragraph () displayed in the browser's default (normal or standard) font:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Now, let's say that you want to change the color or font for this block of text. You can use the style attribute within the tag to accomplish this. Here's the HTML code for this:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, ...

And here's what it looks like:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, ...

But what if you wanted different parts of the paragraph to be a different font and/or a different color. Using the above strategy makes all of the text within the tags the same style. The solution is to use tags. Here's what that looks like:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, ...

The HTML code uses tags to change the style of parts of the paragraph. The has no other purpose; it simply encloses text for the purpose of changing styles.

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISICING ELIT, ...

Margin

The margin property is used to clear an area around an element (outside the border). The margin is completely transparent and doesn't have a background color. The top, bottom, left and right margins can either be defined separately or all margins can be specified at once. It's always recommended to give margin in % like 25%, which will keep it responsive across various devices of different screen sizes.

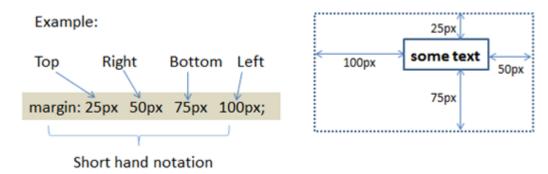


FIGURE 3.35.1 MARGINS SPECIFIED IN A SINGLE STATEMENT USING SHORTHAND "MARGIN"

```
p {
    margin-top: 25px;
    margin-bottom: 50px;
    margin-right: 75px;
    margin-left: 100px;
}
```

FIGURE 3.35.2 MARGINS SPECIFIED SEPARATELY

The first figure shows the shorthand margin where we can define all the four margins in a single statement. The second one is more detailed with separate statements for top, bottom, left and right margins.

Padding

Similar to padding that we learned in element, we can also specify padding for other elements as well. The padding property clears an area around an HTML element (inside the border). Unlike margins, padding does get effected by the element's background color.

```
padding-top: 25px;
padding-bottom: 75px;
padding-right: 50px;
padding-left: 100px;

p.padding {
    padding: 25px 50px 75px 100px;
}
```

FIGURE 3.36 PADDING: SHORTHAND AND DETAILED

Similar to margins, padding can also be specified in many ways. Either by using the shorthand and defining all the four paddings in single statement or in different statements.

Layout using CSS

Below is a good example of formatting an HTML document using CSS styling. In this particular example inline CSS has been used to define the styling. Check the different type of properties used to format the elements. For example, the width property on the <body> tag has restricted the whole HTML's width to 500px.

```
<!DOCTYPE html>
<html>
<body>
<div id="container" style="width:500px">
<div id="header" style="background-color:#FFA500;">
<h1 style="margin-bottom:0;">Main Title of Web Page</h1></div>
<div id="menu" style="background-color:#FFD700;height:200px;width:100px;float:left;">
<b>Menu</b><br />
HTML<br />
CSS<br />
JavaScript</div>
<div id="content" style="background-color:#EEEEEE;height:200px;width:400px;float:left;">
Content goes here</div>
<div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
Copyright © W3Schools.com</div>
</div>
</body>
</html>
```

FIGURE 3.37.1 SAMPLE HTML WITH CSS LAYOUT (CODE)



FIGURE 3.37.2 SAMPLE HTML WITH CSS LAYOUT (VIEW ON BROWSER)

Adding Custom Fonts

CSS also gives the designers the option of using their own fonts. This can be done using the "@font-face" property. Inside the rule, you have to specify a name (font-family) & then give the location (src) for the font file.

Different browsers support different font file formats. Some common ones are:

- .eot IE4+
- .svg Iphone, Safari
- .woff FF, Opera, Chrome (supported by almost all browsers)

For more details about all the file formats & the browsers which support them, look up the "Browser Support" section in http://www.w3schools.com/cssref/css3_pr_font-face_rule.asp.

```
@font-face {
   font-family: 'MyFont';
   src: url('fonts/myFont.eot'),
      url('fonts/myFont.svg'),
      url('fonts/myFont.woff');
}
p{
   font-family: MyFont, Tahoma;
}
```

FIGURE 3.38 @FONT-FACE EXAMPLE

HTML Entities

While writing text in HTML, we should take care of the reserved characters in HTML. These kinds of reserved characters and some characters which might not be available on a keyboard can be defined using entities. For example if we use arithmetic operators like greater (<) or smaller (>), the browser might mistake them for a tag. There are hundreds of pre-defined entities which are rendered by browsers accordingly. Some examples can be seen in the table below.

Result	Description	Entity Name
<	Less than	<
>	Greater than	>
&	Ampersand	&
©	Copyright	&сору;

FIGURE 3.39 HTML ENTITIES.

For more details, refer http://www.w3schools.com/html/html and http://www.w3schools.com/html/html symbols.asp.

IFrames

IFrames are used to embed a web page within a web page. We all might have seen web pages, which a small frame within it which has another we page in it. The <iframe> tag is used to define iframes. Below is an example of using an <iframe> tag. It needs a "src" attribute which specifies the URL of the iframe page. Height and width attributes can be specified to define the size of the <iframe> tag. There are other such attributes, which can be used to modify the appearance of an <iframe> tag http://www.w3schools.com/tags/tag_iframe.asp). They are mostly used to embed social networking tags, blogs, maps, etc. from other sites.

Following is an embedded iframe
 <iframe src="demo_iframe.htm" width="200" height="200"></iframe>

Following is an embedded iframe

This page is displayed in an iframe

FIGURE 3.40.1 HTML IFRAME

iFrame as a Target for a Link

iframes can be used as targets for links. This can be done by first giving the iframe tag a name suing the name attribute of the <iframe> tag. Then set the target attribute of the <a> tag to the name of the <iframe> tag. The figure below shows an example of this. On page load, the iframe will display whatever is specified in its own "src" attribute. But when the associated link is clicked, it displays the web page specified in the link's "src" attribute.

<iframe width="100%"
height="300px"
src="demo_iframe.htm"
name="iframe_a"></iframe>
<a
href="http://www.w3schools.com"
target="iframe_a">W3Schools.com/
a>

When the target of a link matches the name of an iframe,
the link will open in the iframe.



FIGURE 3.40.2 HTML IFRAME AS A TARGET FOR A LINK

Embedding YouTube Video in iFrame

IFrames can also be used to display videos from any video sharing website like YouTube. For using a video from YouTube, click on "share" and click on the "embed" button. (Do not use the URL from the browser address bar). Copy the iframe code and use it in your HTML.

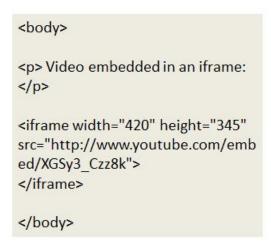




FIGURE 3.41 HTML IFRAME DISPLAYING YOUTUBE VIDEOS

Embed using <embed> tag

Videos can also be displayed using the <embed> tag. This tag is generally used to display content from some external application. This tag has been recently standardized by HTML5. Hence, the <embed> tag might behave differently in browsers which don't support HTML5.

```
<body>
Video embedded using embed tag:

<embed width="420" height="345"
src="http://www.youtube.com/embed/
XGSy3_Czz8k">
</embed>
</body>
```



FIGURE 3.42 HTML EMBED TAG FOR DISPLAYING YOUTUBE VIDEOS

<iframe> v/s <embed>

Why does HTML offer two different tags for the same purpose? The differences below will answer this question.

IFrame

- Importing another webpage in the main page.
- Primarily used to include resources from other domains but can be used to include content from the same domain as well.
- The embedded code is "live" which makes it possible to communicate with the parent HTML.

Embed

- Primarily used to integrate an external application or some interactive media which is typically non-HTML.
- Used to embed content for browser plugins (Exceptions include SVG)

HTML 5

HTML5 is the latest HTML standard for HTML documents. It works the best with CSS3 which is the latest standard for CSS. Following are some major advances which makes the HTML5 interesting:

- Cross-platform, designed to work on various types of hardware (PCs, Tablets, Phones, TVs, etc.)
- New semantic elements which improves the readability of the HTML document. (e.g. <header>, <footer>, <section>)
- New form input types like number, date, time, etc.
- Full CSS3 Support
- Video and Audio tags to standardize embedding audio and video contents.
- 2D/3D Graphics support with <svg> and <canvas> tags.
- With "Local Storage", web pages can store data within the browser. Earlier this
 was done using cookies which had lots of disadvantages like less storage limit
 and data had to be included in every server request.

Adding New Elements to HTML

HTML5 has introduced a new feature where users can create custom tags to the HTML. Gone are those days where developers had to stick with the limited HTML vocabulary. With HTML5, you can tags such that its name gives meaning to its content.

My First Heading My First Hero

FIGURE 3.43 CUSTOM TAGS WITH HTML5

Audio Element

HTML versions before HTML didn't have any standard tag for playing audio files on HTML pages. HTML5 has introduced the new <audio> tag specifying a standard way to embed audio in a web page.

Below is an example of using the<audio> tag. Any text between the starting and closing tags will be displayed if the browser doesn't support it. The "controls" attribute of the <audio> tag adds the audio controls like play, pause and volume. The "autoplay" attribute play the audio file on page load. Multiple <source> tag inside the <audio> </audio> tags contain links to different audio files (generally of different formats). The browsers will use the one it recognizes first.

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
  </audio>
```



FIGURE 3.44 HTML5 < AUDIO > TAG

Video Element

Similar to audio, HTML5 has introduced the <video> tag to specify the standard way to embed videos on web pages. The <video> tag is very similar to the <audio> tag.

```
<video width="400px" controls>
  <source src="mov_bbb.mp4"

type="video/mp4">
  <source src="mov_bbb.ogg"

type="video/ogg">
  Your browser does not support HTML5
video.
  </video>

Video courtesy of
  <a
    href="http://www.bigbuckbunny.org/"
   target="_blank">Big Buck Bunny</a>.
```



FIGURE 3.45 HTML5 < VIDEO > TAG

HTML 5 Graphics

HTML5 has also introduced tags for specifying graphics on a HTML document. The two such tags are canvas and SVG.

Canvas

It is used to draw graphics, on the fly, on a web page via scripting (usually JavaScript). It is only a container for graphics. You have to use a script to actually draw the graphics.

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;"> </canvas>
```

FIGURE 3.46.1 HTML5 < CANVAS> TAG

The canvas element has several JavaScript methods for drawing paths, boxes, circles, text, and adding images. Once the canvas is defined in the HTML, it can be used in the JavaScript to draw over it.

SVG

SVG stands for Scalable Vector Graphics and it is used to define vector-based graphics for the Web. SVG defines the graphics in XML format. SVG tags can be directly embedded in the HTML. All the svg related xml tags are available and can be used directly. The figure

below shows the use of the $\langle svg \rangle$ tag. Notice that inside the $\langle svg \rangle$ tag, there is a $\langle polygon \rangle$ tag which has various attributes. All the space separated number pairs represent the (x,y) points inside the $\langle svg \rangle$ with the left top corner being (0,0).

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-
rule:evenodd;" />
  </svg>
```



FIGURE 3.46.2 HTML5 <SVG> TAG

SVG tags have many advantages. One of them is that the graphics produced by SVG do NOT lose any quality if they are zoomed or resized i.e. they are scalable.

Comparison of SVG with an image when scaled

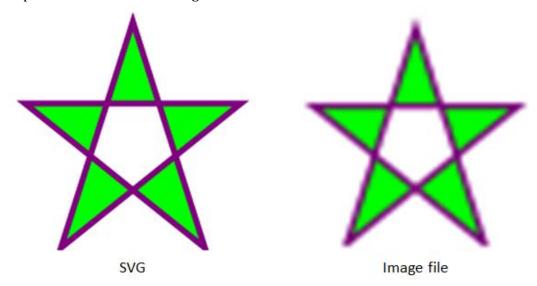


FIGURE 3.46.3 HTML5 <SVG> TAG: COMPARISON OF SVG WITH AN IMAGE WHEN SCALED

HTML 5 Semantic Elements

Semantic elements are elements with meanings. HTML5 offers new semantic elements which clearly define different parts of a web page. This helps in describing meaning both to browsers and the developers. It thus increases the readability of the HTML as well. Earlier versions of HTML used several non-semantic elements such as <div> and , which doesn't tell anything about the contents enclosed inside them. Developers had to use id & class attributes to designate meanings to different elements (e.g. <div id="header">). Some of the new semantic elements offered by the HTML5 are:

 <header>: to define header content of a web page or section. It should be used as a container to define header or introductory content.

- <nav>: to define navigation links. Generally used as container for common navigation links throughout the website pages. (e.g. links which websites generally have at the top or right menu to navigate to the different pages).
- <section>: to define a section in a webpage. It is generally used to split the HTML document into sections like introduction, content, footer, etc. Typically it should contain all related elements inside it.
- <article>: as the name suggests, it is used to define independent and selfcontained content. It is generally used to specify articles which make sense on its own, like news articles in a news website.
- <aside>: these are used to define some content aside from the content it is placed in. It is generally placed like a sidebar giving out some information about something related to the surrounding content.
- <footer>: to define the footer content of a webpage or section. It should always contain information related to the content of the section it is enclosed in. You will see that this tag generally contains information like author, copyright information, contact links, etc.

Although these elements should be used for specific purposes are mentioned above, developers are not restrained to nest any element inside another. The new semantic elements are just a way for the developers to design meaningful HTML code. The figure below gives us an idea of how the elements above might look like.

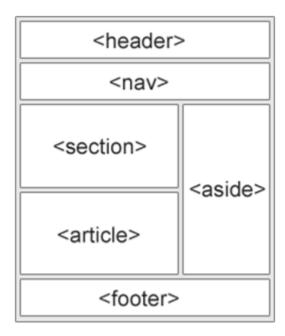


FIGURE 3.47 HTML5 SEMANTIC ELEMENTS

HTML 5 Semantic Elements

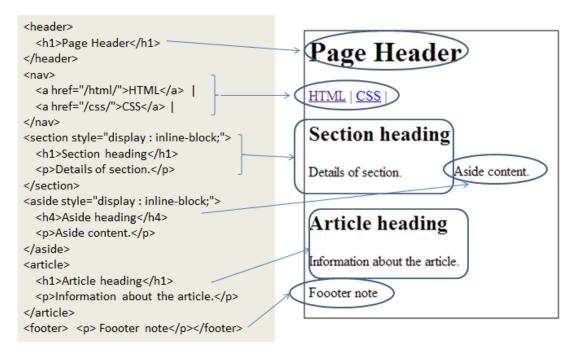


FIGURE 3.48 HTML5 SEMANTIC ELEMENTS EXAMPLE CODE

The figure above is an example of a simple document created using the new HTML5 semantic elements. Look closely that the elements inside these are still the same ones we have been talking about till now e.g. <h1>, , etc. The only difference here is that instead of using the meaningless <div> & elements, we can now use meaningful tags which say something about the content without even looking at the contents inside.

Creating HTML Pages with Notepad

Building "pure" HTML pages requires the use of a text editor that does not add anything to the text file when it is saved. On Windows, the *notepad* program works well. On a Mac you need to find and install a text edit; the *textedit* program does not work unless you follow the instructions for editing plain text files (also sometimes called ASCII files) at http://support.apple.com/kb/TA20406.

The notepad program can be found under Programs/Accessories. Here's a quick video (http://www.screenr.com/W2ZH) on where it is and how to create and save a file in notepad. The key is to load notepad, create the HTML file, and then save it. When saving, specify a .html extension. To view the file, just click on it from your file folder where you saved it. This will preview the file in your default browser. To edit the file again requires that you go to the file folder, click the right mouse button, and then choose "Open with", select your own program and pick notepad. Here's a tutorial (http://www.screenr.com/r2ZH) on how to do that.

To "link" pages is simple: just build each of your pages and then place <a> tags into the page where you want to link to another page. The pages are independent; the linking is based on how to add links from one page to another.

Frequently Asked Questions

Do I need the <!DOCTYPE> and <html> tags when I embed HTML content in my web tool?

• No. You are embedded the HTML within an HTML page, so your web-publishing tool (Weebly, Zoho, Wix) already provides these tags.

My content shows up fine in my web-publishing tool, but when I publish it, it doesn't look right.

• This is most commonly due to your HTML being not well formed. Most likely you forgot to close a tag or you nested them incorrectly. For example, be sure to close your tags and tags.

Everything is indented after I create my table. How come?

• You forgot to close some tag, such as or or . Remember that every tag must have a matching closing tag. Check your HTML carefully. This process is called "debugging your code". It can be fun, but also really frustrating; it's like solving a puzzle. Some people like this and some really hate it. If you like it, consider majoring in Computer Science.

How can I get multiple columns?

Use embedded CSS or HTML tables. See the slide deck on how to use <div> to get
multi-column lists. Here's another (better, but more complicated) way: <u>Approach</u>
 1 | <u>Approach 2 with CSS</u> | <u>Approach 3 with DIV</u>

My bullets don't show up in the list when I publish it in Weebly but shows up when I edit the code and in the HTML editor?

• You need to use as your bulleted list. It appears that Weebly is modifying the style to have a negative margin, which causes the bullets to be cut off.

Why can't I build my own CSS style sheet in Weebly?

• Weebly already provides a style sheet, so the only way you can modify the style is with embedded styles.

How can I make my header image stretch across the entire page width?

• Specify width="100%" for you tag. If the height is too big, use height="125px" or something that suits your image. Note that width and height can be specified as % or px. % means that it's some percentage of the width or

height of the page while px is a fixed width or height in pixels (dots). Here's some sample code for an image that stretches across the entire page:

Summary

We discussed most of the HTML elements & CSS styling, which will help you, construct your own HTML document. HTML has a vast collection of elements and CSS properties to learn, but this would certainly be a starting step for you to build a simple HTML page.

Integrating Multimedia Content

At the end of this chapter, you will be able to:

- ✓ Link images in an HTML page using the tag
- Differentiate between the different image file formats, including GIF, JPG, TIFF, and PNG
- ✓ Use custom RGB colors
- ✓ Render text in non-standard fonts
- Embed videos and audio streams

Images

An Image can be defined as a visual perception that depicts the appearance of a subject such a person or a physical object. Images can be two-dimensional, such as a photograph, and as well as three-dimensional, such as a hologram.

Importance of images

Images in various forms enhance a website due to many reasons. One reason would be that images help in delivering messages visually. We all have heard about the phrase, an image is worth a thousand words. This stands true when used in website design as well. Images on a website certainly make it more personal to the website owner. For example, amazon, the e-commerce company, needs images of its logo and products to make its website relate to its business. Users are more inclined towards images than reading a whole lot of text describing the same content. All companies have now understood the importance of website and the role of images on them. And thus almost every company spends lots of effort and revenue in building their website and employ designers who can create images which increase their credibility and increase the count of their customers. This is because it has been observed that images can draw attention of the users can trigger a prompt action. Images also improve the aesthetics of websites to a great extent.

Now that we have discussed about the importance of images, we should also know about how to use them. Images should be used in such a manner that they enhance the visual appearance rather than just pasting them on the web page. They should be used creatively. Some useful creativity like the following can be implemented:

- Use background images in websites to give an immediate impression of the website.
- Tilt or border images to create additional whitespace around them.
- Use dithering or other enhancement techniques.

Label all clickable images on your website.

Sources for Images

Images can be created in many ways. One way is to create your own images using the following:

- Take photos with a digital camera. Digital cameras with low resolutions, like
 1.2 megapixels is good enough for using it on websites.
- Scan images or photos with a scanner. A scanner gives you the options of saving the scanned image in various different formats. We shall talk about them later in this chapter.
- Draw an illustration of a photograph or some real world subject. This gives a free hand to the designer to create and visualize objects in a creative manner.

Examples of Images

As discussed above, below are some examples of images of different types. Figure 4.1 is a hand drawn illustration which can be used in a website developed for a coffee shop. Figure 4.2 shows an image taken with a digital camera and figure 4.3 is an example of a scanned image.



FIGURE 4.1 HAND DRAWN ILLUSTRATION:

[IMAGE FROM FREEDESIGNFILE.COM UNDER A CREATIVE COMMONS LICENSE.]



FIGURE 4.2 PHOTO TAKEN WITH A DIGITAL CAMERA



FIGURE 4.3-SCANNED IMAGE FROM A CATALOG

Using Images from the Web

Apart from using self-created images, images can also be used directly from the web. We have already discussed about embedding images in a HTM document in previous chapter. Once we have located an image (its URL) on the web, we can use its URL in our HTML. It seems to be fairly simple process, but there are certain restrictions when using images from another source. The following are a few points to consider while using images from the internet.

Check for copyright information of the image. Copyright laws were created to enable the authors to protect their creation, in these case images. It attaches along with every image upon creation. Authors generally use "circle c" to illustrate that the image is copyrighted.

Purchase a license for an image, but watch license restrictions. Buying a license for an image may not involve complex legal documentation, but it might not be straight forward either. But before going ahead, there are certain things to consider. For example, what way will the image be used, the duration of the license, any real people on the image, whether the image is an artistic work, etc.

Follow the Creative Commons license type. The CC license provides a legal understanding to what kind of restrictions are applied to images. It is a public copyright license, which allows you to use, tweak images for your work as long as you give credit to the owner. Visit their website @ https://creativecommons.org/ to get detailed information.

Once the licensing part is taken care of, the correct usage of images should be given equal thought. For efficiency, always use image at the right size and resolution. Having large images can effect loading time and hence can counteract a positive experience for the users. We might think that internet connections today are very fast and we can use large images. But, people might also access your website from mobile or from a low-speed connection. Hence, it's important to optimize the images used in websites.

Creative Commons License

As discussed above, a Creative Commons (CC) license is one of several public copyright licenses that allow the free distribution of copyrighted work.

A CC license issues a free use of a work. The licensor and licensees can both have unlimited permissions. Typically the licensors apply for such license to give permission to others to copy or change their work which otherwise might infringe the copyright laws. Hence, any person can apply for this license and anyone can use the licensed work according to the conditions.

There are several types of CC licenses. But all of them provide the basic rights i.e. right to use them without modification for non-commercial purposes. The various versions of the CC license comprise a selection out of four conditions:

- **Attribution (BY):** gives licensees the right to copy, display, distribute and make derivatives based on it only if they give credits to the licensor.
- **Share-alike (SA):** under this the licensees can distribute only under a license identical to the license that governs the original image.
- Non-commercial (NC): allow only non-commercial uses of their own work to licensees.
- **No Derivative works (ND):** same as attribution condition except that licensees cannot produce derivatives of the image.

The licensing industry keeps updating the rules and newer versions keep rolling out. Thus it is important to keep track of the new and deprecated rules.

Scanning Images

Use a flatbed, sheet fed, or drum scanner to scan images. While scanning images for websites we should consider certain things, which might affect the loading time.

- Scan the images at 72dpi (dots per inch). Higher dpi makes the file size bigger.
 72 dpi is enough for websites.
- Use higher resolution if images will be manipulated or printed, e.g., 150dpi.
 This is recommended because manipulating images with lower dpi is harder and doesn't produce good results.
- Scan illustrations at 256 colors. Although having more colors increases the file size, it does not improve the looks, especially in case of illustrations.
- Scanning provides the user to save the scanned image in various formats. But it is advisable to save images in TIFF to retain all colors and quality. We will discuss about TIFF and other formats later in this chapter.

Screen Resolution

We have been talking about certain factors while creating images like resolution. What does it mean? Resolution is the detail an image hold. Higher resolution means that the image would hold more detail and would look clearer. There are various ways to define resolution. One of the most common ways to do it is by defining resolution in terms of pixel. Pixel is the unit of a digital image. Resolution in pixels is defined in by the number of pixels it can hold vertically and horizontally. It is usually depicted with a set of two positive numbers, where the first defines the number of pixel columns and the second to describe the number of pixel rows e.g. 1280 by 800. Figure 4.4 shows how pixels are represented in an image. The total number of pixels in an image is the product of the pixel rows and columns. For example a 1280 by 800 pixel resolution would have 1280x800 = 102400 number of pixels.

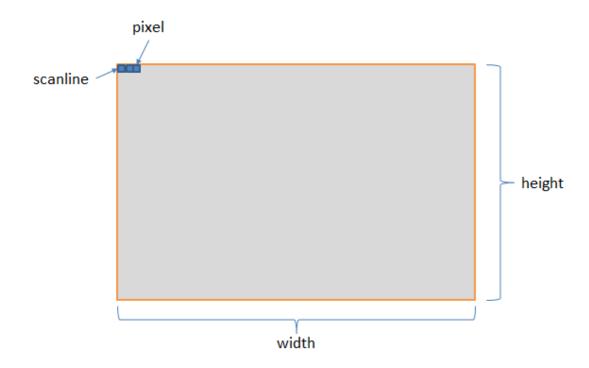


FIGURE 4.4 SCREEN RESOLUTIONS IN TERMS OF PIXEL

Figure 4.5 gives an illustration of how an image would look in different pixel resolutions.

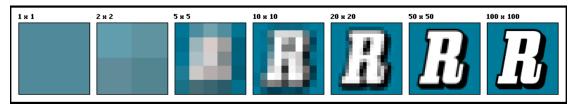


FIGURE 4.5 RESOLUTION ILLUSTRATIONS

[IMAGE SOURCE:

 $\label{lem:http://en.wikipedia.org/wiki/image_resolution\#mediaviewer/file: $$ RESOLUTION_ILLUSTRATION.PNG$$]$

Pixels and Color

As we discussed above, pixels are individual dots on the screen and in an image. The density of pixels defines the detail of an image. Every pixel represents a color. One way of storing images is by using the 8-bit color rule. Every pixel is one 8-bit byte. Hence a maximum of 256 colors can be displayed at a time.

Bit 7 6 5 4 3 2 1 0 Data R R R G G G B B

FIGURE 4.6 8-BIT IMAGE FORMAT

The most commonly used color scheme is the 24-bit. In this pattern, colors are always combination of triplets of red, green and blue. And each color is represented with 8 bits. Hence, a total of 24 bits per pixel is used. This means that there can be a total of 2^{24} i.e. 16.7 million different colors. Hence it is often referred to as true color.

Image Scaling

Image scaling is the process of resizing an image. The web browsers scale the images to the right size automatically.

You might think by now that using larger images with high pixel density would enhance the aesthetics of the website. But in fact it would not necessarily lead to a higher display quality. Larger images will take more time to render because the browser will take time to download it first. It will depend on the resolution of the screen on which the website would be displayed. So, having a large image for a smaller screen resolution would mean that the browser has to download it and then reduce its size to fit the smaller screen.

Images can also be scaled through an image processing program, such as Photoshop and even Paint.

Scaling above an image's native (original) size leads to "blurriness". The figure below shows how this is true.





FIGURE 4.7 IMAGE SCALING BEYOND ORIGINAL SIZE INDUCES BLURRINESS

Hence, it is important to be aware of the various screen resolutions the website would be viewed on. And this means that the number of larger images in websites should be kept as low as possible.

Image File Formats

Image file formats are used to organize and store image data on the disk. The image files consist of digital data in various formats. Different file formats store the data i.e. the bit

information in different ways i.e. compressed, uncompressed, etc. Computers can read from the file according to the format and display it on the screen. We will discuss about the compression in the next topic.

There are hundreds of image file formats, but the most commons ones are JPG, GIF, PNG, TIFF and EPS.

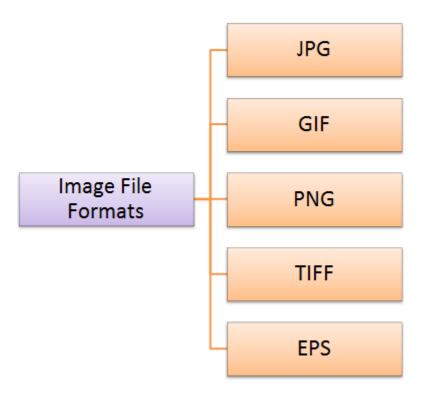


FIGURE 4.8 COMMON IMAGE FILE FORMATS

Image Compression

Some of the file formats store image data by compressing them to save storage space and reduce the network download time. This is important since we would want the browsers to download the file quickly in case of metered networks (Metered networks are those which charge the users by the amount of data sent and receiveed.).

Compression techniques can be either "lossless" or "lossy".

- **Lossless:** the compression algorithms used reduce the file size but preserve the original image pixels. Formats using lossless methods generally result in larger files.
- Lossy: the compression algorithms remove the pixels that won't affect image quality very much. This results in better compression, but at the cost of quality. Most lossless compression achieves smaller file sizes than lossless compression.

The figure below illustrates the different file formats which fall under lossless and lossy compression techniques.

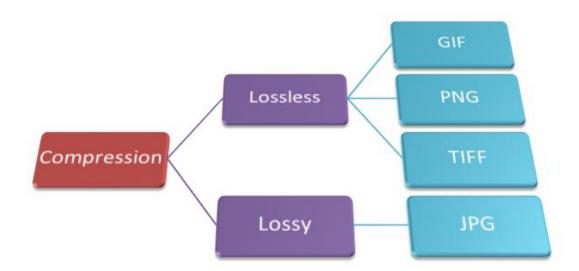


FIGURE 4.9 IMAGE FILE FORMATS AND COMPRESSION

We will not discuss briefly about all these image file formats in the following sections.

GIF

GIF (Graphics Interchange Format) was introduced in 1987. This format supports 8 bits per pixel of the image. Hence, an image with this format can have up to 256 different colors. Hence it has lower vibrancy.

GIF supports animation and allows a separate palette of 256 colors for each frame of the animation.

GIF images allow Interlacing, which makes for quicker viewing while downloading. Interlacing is a type of image reconstruction on screens, in which the rendering is not sequential. This allows a partial display of the image while the whole image data is being downloaded.

GIFs also have a property of having separate background for each frame of the animation. This is useful when you would like to change the background without changing the animation effects on the foreground.

As we have seen in the earlier sections that GIF is a lossless compression format. Hence it produces larger files sizes but the quality is retained.

Animated GIFs

GIFS support animation i.e. multiple GIF images in a single file can are displayed with programmable delay. In simple words, an animated gif file has a number of frames which are displayed in succession with some delay.

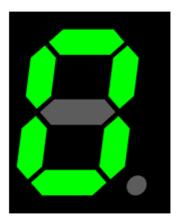


FIGURE 4.10 ANIMATED GIF: EXAMPLE OF NUMERIC COUNTER

[IMAGE SOURCE:

HTTP://WWW.FREEALLIMAGES.COM/WP-CONTENT/UPLOADS/2014/09/ANIMATED-GIF-IMAGES-2.GIF]

PNG

PNG (Portable Network Graphics) is a format, which supports lossless compression. It was developed as an improvement for GIF, and is now widely used lossless image format over the internet. It was solely designed for transferring images over the internet. Thus it doesn't support any non-RGB color-scheme.

It can have either 256 colors with transparency or 16.7 million colors. This file format also results in larger file sizes because it's based on lossless compression. But one improvement over GIF is that it's smaller than GIF. Similar to GIF, PNG format also uses interlacing, but with a more complex algorithm. Image data is loaded in several passes.

One very important feature of PNG is that it has the capability of embedding text within images so that search engines can index images. Having this feature improves its chances of showing up on a web search.

TIFF

TIFF (Tagged Image File Format) is also a lossless compression format. It is popular for its high color-depth and thus widely used as a scanned image format. It was first developed in in mid-1980s when it supported only binary images because the scanners and desktops at that time could handle that much only. Eventually with the increase in desktop space and advancement in scanner technology, TIFF was improved to support grayscale and color images. Images in TIFF format have the highest quality and hence take up more size on storage disk.

TIFF versus JPG

Notice in the figure below how the TIFF file is much clearer compared to the JPG file, but it's also much larger. JPG/JPEG is one of the most widely supported formats, which offer smaller file sizes. This is because it uses lossy compression and hence loses some data in the process. But TIFF proves to be an excellent choice when file size does not matter.





FIGURE 4.10 TIFF VS JPG

TIFF versus GIF

The difference between TIFF and GIF is even more pronounced than TIFF vs JPG as GIF reduces the number of colors as well as uses compression. No matter how many colors the original mage may have, GIF always reduces it to 256 colors. Hence, it loses quality and isn't recommended for images with high range of colors.



TIFF



GIF

FIGURE 4.10 TIFF VS GIFF

File Size Differences

We can observe a difference in file sizes of the same image stored in different formats. This is because every format uses a different compression technique. In figure 4.11 it's

clearly visible that compared to TIFF format all other formats have quite a reduction in file size but with the loss of data. This can be seen in figure 4.12 where TIFF format has the highest quality and GIF has the lowest. At last figure 4.13 illustrates the recommended usage of file formats for different requirements.

File Type	Size	% of TIF	Comment
TIF	14.5kb	0%	
PNG	11.3kb	78%	Lossless compression; full color
JPG	4.89kb	34%	No transparency; full color
GIF	3.87kb	26%	Fewer colors

FIGURE 4.11 FILE SIZES IN DIFFERENT IMAGE FORMATS



FIGURE 4.12 NORTHEASTERN LOGO AT A RESOLUTION OF 80×76

	Use	Don't Use
GIF	small graphics with limited color	photos
PNG	graphics and small files for web	photos for widespread use
JPEG	photos on the web	editing images
TIFF	editing and storage	online images

FIGURE 4.13 RECCOMMENDED USAGES OF FILE FORMATS

[IMAGE SOURCE:

HTTP://WWW.SMARTIMAGE.COM/WHATS-THE-DIFFERENCE-BETWEEN-GIF-PNG-JPEG-AND-TIFF/

In Practice

In practice, it's either TIFF or JPG, rarely GIF, for photo images. GIFs are mostly used as a loading image (a small running circle with "loading" text) while the browser loads the larger image. But background images are sometimes in PNG as it allows transparency. For absolute details about all major file formats, visit http://www.digitalmemoriesonline.net/scan/output/jpeg_vs_tiff.htm.

Thumbnails

Thumbnails are smaller versions of an image used for previewing. It replaces text from websites very efficiently and thus increasing the readability. We can see a lot of websites using thumbnails in place of text to display lists of items.

It is advisable to use separate thumbnail files instead of scaling larger images. Simply scaling them doesn't reduce the file size and websites generally have multiple thumbnails. This will cause a higher loading time.

The Tag

Now, how do we effectively use images so that users don't feel distracted when the browser has to deal with downloading images from the web? The tag in HTML has an attribute called "lowsrc" which when specified loads that image first and then the image specified with the "src" attribute. By doing so, the browser can load the smaller image quickly for the user and then load the actual bigger image. No doubt that the LOWSRC image is of lower quality, but it effectively engages the user till the actual image is loaded and displayed.

Streaming Video and Audio

With increasing internet speeds, websites nowadays have a lot of videos and audios.

Animation

Along with videos and audios, animations are also widely used. Animations are gaining a lot of importance in web design these days as they have proven to increase the credibility of websites and increases user engagement to a large extent. Animation can be created using the following:

- **Animated GIFs:** As discussed before, GIFs are greatly used in animation.
- **Flash:** flash animations can be created using Adobe Flash or other similar software, and often found in SWF file format.
- Java Applets: Java also has an applet API which is specifically made for creating animation programmatically. This is obviously an open source API and is being used very widely in the IT industry.
- Microsoft Silverlight: This is Microsoft's answer to animation software.
 Although it first started with streaming media on browsers, latest versions support animations as well.

Downloading vs. Streaming

We all must have heard about downloading and streaming media over the internet. But, what is the difference between them? Audio and video files on websites can be downloaded and/or streamed. Both allow users to play them but with different techniques:

Downloading: in this case, the media file must be first downloaded (saved in storage) fully before it can be played (listen or/and watch).

Streaming: Streamed media starts playing after a local buffer is filled. The buffer is kept full while the media is playing. The buffer is constantly updated as the media is played. Slower internet speeds will experience a delay while faster internet connections would hardly notice a lag.

	Advantages	Disadvantages
Downloading	Content can be accessed offline Uses HTTP and does not require special media servers	 Can take a long time to download Content is not accessible until fully downloaded No previewing of content Requires local storage Content is distributable and digital rights are difficult to enforce
Streaming	 Seeking and random access Does not require local storage Previewing of content possible Does not require special software on browser Digital rights more easily enforceable 	 High bandwidth requirements for continuous streaming Requires a media server on the web server Re-playing content requires re-sending of the same content

FIGURE 4.14 DOWNLOADING VS. STREAMING

Audio

Digital audio is stored in various formats like mp3, 3gp, wma, etc. But audio has to be stored (encoded) in such formats so that web browsers can play them. Formats with lower file size (lossy compression) such as mp3 gained a lot of popularity because it enable sharing audio files over the web even with slower Internet connections. Quality and the recording format affect the file sizes the most. The following sections will discuss about the various parameters, which affect the audio file sizes

Mono vs Stereo

Mono (Monophonic) is referred to audio in a single channel and is thus inexpensive to record and takes less file size. Stereo (Stereophonic) on the other hand uses multiple independent audio channels. This creates an impression of sound heard from different directions i.e. similar to natural hearing.

Sampling rate

Sampling is the process of extracting value(s) from a continuous signal at periodic intervals. Audio files are stored with respect to sampling rates. Sampling rate is the number of audio samples taken per second. It is measured in Hertz (Hz). For example if 8000 audio samples per second can be represented as 8000Hz or 8kHz. Higher sampling rates means more samples are stored which leads to better audio quality. Following are some common sample rates used for various purposes:

- 8kHz is fine for telephone speech i.e. voice.
- 22kHz is good enough for web audio (mp3.)
- 44.1kHz is typically used for audio CDs and the highest frequency, which is audible to human ears.
- 48kHz is for higher quality audio like in DVDs.

Bit depth

Bit depth is defined by the number of bits used to represent each sample. For example if it is encoded to 4-bit then each sample can have one of $2^4 = 16$ possible values.

8 bit audio vs 16 bit vs 128 bit

Higher bit value will result in more audio samples and hence better quality. 8-Bit resolution is good enough for low quality audio only. 16-Bit can be matched with CD quality audio i.e. higher quality. 32-Bit is generally used for the best quality of audio. This is because the number of possible samples is 2^{32} makes it possible to store more details of the audio.

128-Bit is still not used yet due to hardware limitations. Since the 32-Bit format is used in recording studios to store the best quality audio, higher bit depths have not yet gained popularity.

Codecs

Codecs are compression algorithms that can greatly reduce the audio file size. It is a computer program which compresses and decompresses digital audio data to a specified file format. Codecs were developed to store high quality audio with the minimum number of bits while retaining the quality.

But, most compression algorithms are "lossy", so quality is reduced to achieve smaller file sizes. Encoding with a codec requires a player that supports the codec when playing back. This means that the player should be able to decode the file and then play them.

Streaming Audio

Streaming audio from the web requires a streaming media server. The audio files play back as they download, so audio starts almost immediately. The audio data from the server is stored in the buffer, which is updated as the playback proceeds. Majority of the

web browsers support audio streaming. There are various formats for streaming audio. Windows started with its first media player ActiveMovie that could stream media. Later is introduced Windows Media Format (WAV) which was a much improved version. Then other companies also introduced streaming media formats like QuickTime (Apple Product) and RealPlayer to enable streaming media directly from the web.

Having different propriety formats require users to download all respective applications. This problem was sorted when a single and unified streaming format could be now developed using Adobe Flash. Such formats could be played in Flash-based players. Video sharing website YouTube also uses a flash-based player for its video streaming.

Stream Hosting

Most of the websites use hosting platforms to handle streaming and media formats. These hosting platforms allow to users for making their website and media available via the web. Web hosting companies offer such services on sale or on rent. Some popular media hosting websites are:

Video

- YouTube
- Vimeo
- Viddler

Audio

EdgeCast

Dynamic Websites

Upon completion of this chapter, you will be able to:

- ✓ Zoho creator
- ✓ Build database tables using the Zoho form builder
- Create dynamic pages using Deluge

Static and Dynamic Websites

A user might not always be able to differentiate between a static and dynamic websites. Static websites are those which have only content pages. Every page in such websites is a separate HTML page. None of the contents in any of these pages is created on the fly or on user demand. Even if two pages contain a part of identical content (e.g. header), they both contain two versions. So, if we have to update the header, it has to be done twice, once on each page.

Boston Bruins Goalies



Tuuka Rask



Malcolm Subban

FIGURE 5.1 STATIC WEBPAGE

Assume we have a static website which lists the information about the goalies in a team. Now, if additional goalies are to be added, a web developer has to modify the page and republish the website. This seems fairly straightforward and was actually the way websites were built in the early days. But this is actually very time consuming, costly and prone to error. Static websites have to be updated regularly since data may have changed. The solution is dynamic websites which can solve the problems we discussed.

Why Dynamic Websites?

Dynamic websites are those which collect data and display webpages that are generated right before it is sent to the browser. It is also known as generating webpages on the fly. Instead of having web pages with static information on it, store the data in a database and write server side code (e.g. ASP, PHP) to extract the information and build the page programmatically. The server side code may also consume other information such as the user's time zone, its IP address, etc. to generate the web page contents. Once built, the code doesn't need to change frequently to modify the data displayed. Updating the data in the database will modify the page content.

To sum it up the benefits of having a dynamic website are:

- Easy design updates
- Flexible data
- Easy content updates

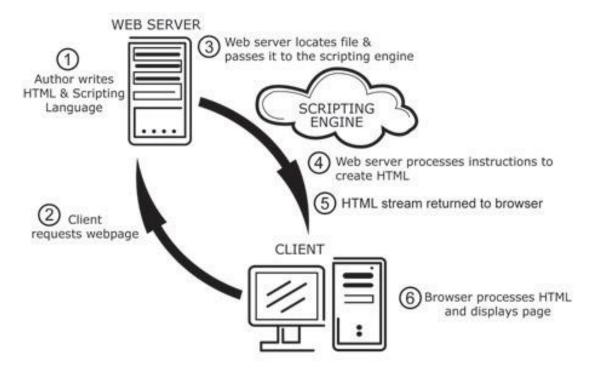


FIGURE 5.2 DYNAMIC WEBPAGE GENERATION

The figure above illustrates the generation of dynamic websites. Step 1 means that the HTML and server side code is uploaded on the web server. Then the user requests for a web page. The webserver responds to the request by fetching the required data. The scripting engine then decodes the script and generates the HTML which is then sent to the browser. The web browser simply displays the received HTML.

Scripting Engines

Scripting engines are used for implementing scripts in scripting languages. Scripting languages are different from other programming languages in a way that they are generally built on platforms that would allow more automation.

There are several server-side scripting engines that use different programming languages through which scripts can be embedded with HTML tags. This script runs on the server and handles the client requests before the server responds to the request. Some commonly used scripting engines are:

- PHP as the scripting language accessing data from a relational database, most commonly MySQL.
- Ruby on Rails using MySQL.
- ASP.NET using SQL Server or Microsoft Access.
- Zoho Reports using Deluge and ZohoDB.

In this course, we will be working with Zoho.

Common Technology Stacks

Technology stacks consists of layers of components that are used to provide a software solution. The common technology stacks used to develop server-side scripting for website development are LAMP and WAMP.

LAMP is an acronym for:

- Linux or Windows as the server operating system
- Apache as the web server
- MySQL as the database
- **P**HP as the dynamic page scripting language

WAMP is the same as LAMP except that 'W' is for Windows.

Our Stack

In this course, we are using a simplified stack requiring less programming experience. In place of LAMP/WAMP, we will be using the following:

- Hosted server (AwardSpace/Weebly)
- Hosted web development platform (Weebly)
- Database (Zoho DB)
- Dynamic page scripting language (Zoho Deluge)

Weebly is a service which allows users to build websites by using the drag-and-drop feature on their website builder.

ZohoDB is a service provided by Zoho to build databases online. This is similar to weebly as it is allows the users to drag and drop and it has over 20 different types of fields.

Zoho Deluge works and looks very similar to PHP and ASP.NET. But it is simpler to program since it provides a very simple visual interface for users to drag and drop from the tools available on screen.

Dynamic Page Generation in Zoho

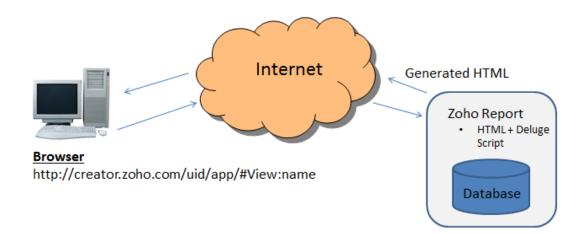


FIGURE 5.3 DYNAMIC WEBPAGE GENERATION USING ZOHO

Once the webpage has been created using Zoho, it can be accessed from a web browser over the internet. The URL for the website is provided by Zoho after the website is created. We will look into the details of creating and publishing a website using the stack we mentioned above.

Zoho Creator

Once an account is created on Zoho, you can start building your own website. The steps to create dynamic pages in Zoho Creator are as follows:

Give your application a name (a meaningful name which would give an idea about its purpose).

- **Define database:** Firstly, we have to define the database using the form builder of Zoho.
- Create Report(s)/ Page(s): Then, create report(s) and/or pages to display the data which has been defined using the form builder.

We will now look into all the steps in detail.

Data: Definition & Generation

A database is nothing but an organized collection of data. Typically the data is organized using tables. You can think of a table as a huge excel sheet where the columns represent the field and every row in that excel sheet represents a record.

The Zoho creator uses forms to create the database. It allows the user to drag and drop the fields and construct the form. Figure 5.4 illustrates the form builder which you will see while creating the form on Zoho creator. It is divided into three main sections. The middle section is the main form. The leftmost column on the form contains the templates for the fields which can be added to the form. The user has to just click on a field and drag to the main section. The right section has two sub-sections – Form properties and field properties. As the names suggest, we can modify the fields and form properties using that sections.

Once you have added all the fields you want, click on "Access this application" button on the top right corner of the webpage. Zoho will create a table in the database for you with the fields you have provided and direct you to a page which can be used to add data to the database.

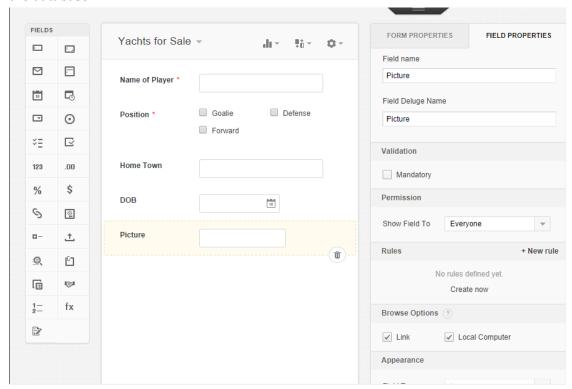


FIGURE 5.4 ZOHO CREATOR FORM BUILDER

Role of a Zoho Creator Form

To sum it up, the Zoho form builder has the following roles:

- Define the structure of the database tables.
- Create the tables.
- Provide a data entry form for users to enter data into the database directly through Zoho Creator.

Adding Data to the Database

Once you have created the form using the form builder, the next step is to add relevant data. Zoho provides two ways of adding data:

- Running the form and then adding the data using the web form which Zoho creates for you. This option using the webpage, allows users to only add data to the database.
- Add records using the built-in view provided by Zoho. This built-in view allows the users to add, update and delete data records.

Adding Data to ZohoDB by running the form

After you have completed the form, click on the "Access this application" button and Zoho will automatically direct you to a webpage which will have all the fields you have mentioned in your form. Figure 5.6 shows an example of the webpage created by Zoho. All you have to do is fill in the values and click on submit. Zoho will add the record to the database tied to this form. You can add any number of data records using this form.

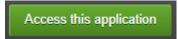


FIGURE 5.5 THE ACCESS APPLICATION BUTTON

You can have multiple forms in your website and can add data to all those with the same process we discussed above.



FIGURE 5.6 ADDING DATA USING THE WEBFORM GENERATED BY ZOHO

Adding Data to ZohoDB using built-in view

Zoho also provides you with a built-in view to add update and delete data. This is available as a "<Form Name> Report" link on the top left of the same webpage which

Zoho created for you. In Figure 5.7 we can see the report link circled in red. Clicking on this link takes you to the view page which is shown in Figure 5.8.

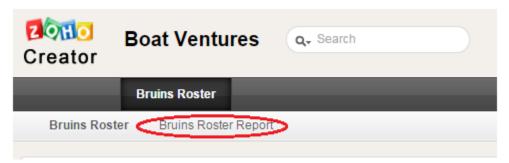


FIGURE 5.7 LINK TO THE BUILT-IN VIEW

This built-in view is very user friendly and you can modify the data records easily by using the buttons and checkboxes provided on the page. For example to add a new record, click on the "Add" button. You will see a new section pop-up on the screen which allows the user to fill in data one record at a time.



FIGURE 5.8.1 ADDING DATA USING THE BUILT-IN VIEW GENERATED BY ZOHO

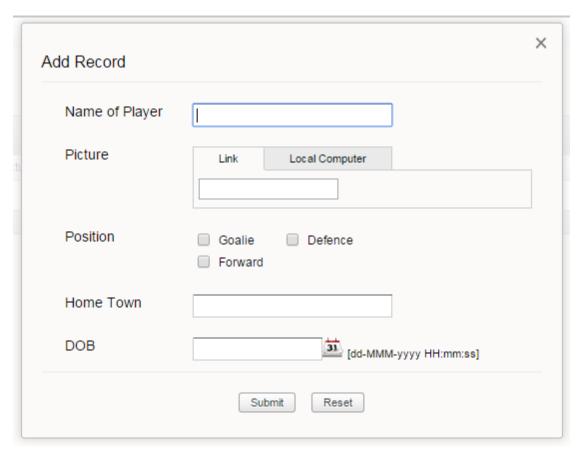


FIGURE 5.8.2 ADDING DATA USING THE BUILT-IN VIEW GENERATED BY ZOHO

Similarly, there are editing options for editing the existing data records. Remember that the previous method didn't have the editing options.

Retrieving the Data from ZohoDB

Once you have added data to ZohoDB using the form and/or the built-in view, it can be retrieved using the following ways:

- A report using a built-in layout. A default report is created whenever you first run the form by clicking on "Access this application". We can create multiple additional reports as per our needs. We will look into the creation of custom reports using Zoho in the next sections.
- A Deluge script with embedded HTML to create a custom layout. Deluge is Zoho's proprietary language for building dynamic web pages, similar to PHP and ASP.

Built-In Layouts / Reports

Just below the Zoho Creator dashboard button, select "Create New" and then "Report". You will see a form with the details of the report like the name, the report type, which data table (form) this report will use, etc. Once you are done providing details, click on "Create and Customize". This will redirect you to a webpage where you can customize

the report. You can change the display names of the fields, create groups of fields, apply sorting to one or many fields, etc. Once you are done applying customizations. Click on "Access the application" button and Zoho will redirect you to the reports page.

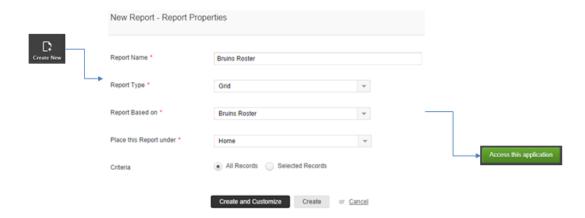


FIGURE 5.9 CREATING REPORTS USING BUILT-IN LAYOUT

Figure 5.10 is a report that shows all records and does not have any filters (As we chose to do in figure 5.9). Similarly, you can add any number of reports with different customizations or you can update the same filter whenever you need to change the report.

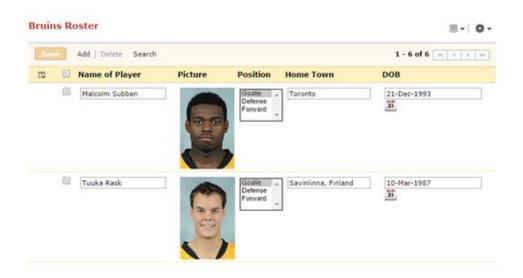


FIGURE 5.10 ZOHO CREATOR REPORT

Custom Deluge Pages

To create dynamic pages that use a custom layout, use the "New Page" option by selecting "Create New" \rightarrow "Page". Enter the name for your page and click on "Create". Zoho creates a page for you with a blank script to start with.

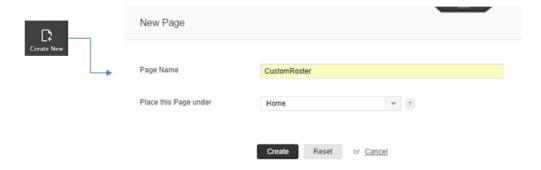


FIGURE 5.11.1 ZOHO CREATOR REPORT

Figure 5.11.2 shows the template which is created by Zoho. No we can write scripts and HTML code on this page to customize it.

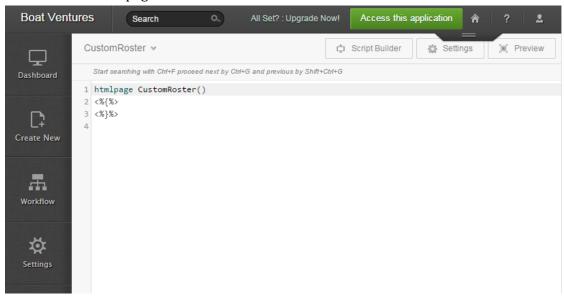


FIGURE 5.12.2 ZOHO CREATOR REPORT

The pages created can be accessed anytime by selecting "Dashboard" \rightarrow "pages" tab(As shown in figure 5.13).

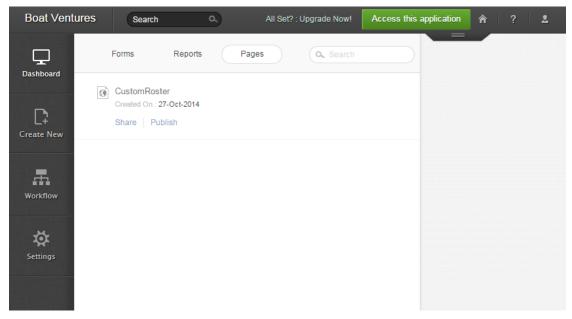


FIGURE 5.13 ACCESSING PAGES OF YOUR WEBSITE

Editing Pages in Zoho

Once we have created a page and understood how to access it from the dashboard, we can now start with writing some simple code to display the information using Deluge. It is very simple to use and allows the user to write HTML tags along with some scripting code to manage the data from the database. Below is an illustration of page creation and running the page code to get the display. We will look into some basic Deluge code and understand how to extract data from the database and display it using plain HTML and some scripting.

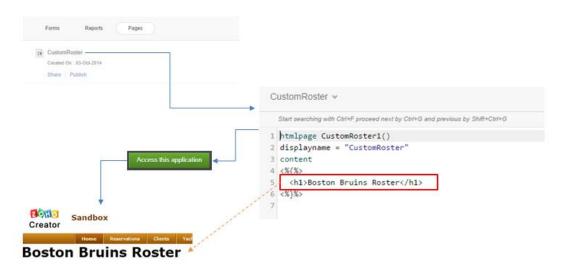


FIGURE 5.14 ACCESSING PAGES OF YOUR WEBSITE

Writing Deluge Code

There are few important points before we dive into the coding part. These tips will help you to use Zoho creator efficiently.

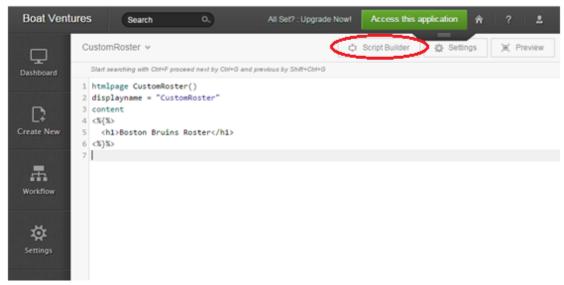


FIGURE 5.15.1 PAGE SCRIPT EDITOR

Firstly, always use the "Script Builder" option. The option of "Free Flow Script" may be enticing and seem to be easier, but the interface and tools provided for that in Zoho is not too user friendly and is bound to confuse you. Another advantage of using the script builder is that you will learn how to write basic HTML and embedded code.

After you have written your code, make sure to click on "Save Script" before you move away from this script editing window.



FIGURE 5.15.2 PAGE SCRIPT EDITOR

Accessing Data

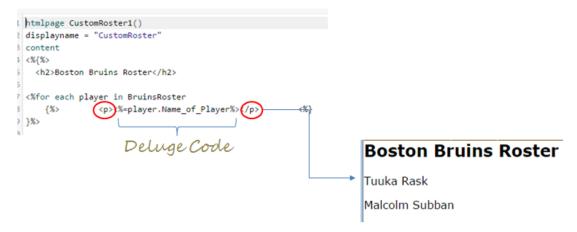


FIGURE 5.16 ACCESSING DATA USING DELUGE

Figure 5.16 shows a basic embedded script in deluge where we are using the data we entered using the form builder. In this script, we have used HTML tags as we have discussed in previous modules to display data on the web page. And the data from the database is accessed using the script.

Basic Deluge Data Access

Now, let's learn the basics of deluge statements. The first thing to remember is that every deluge script is enclosed in "<%" and "%>" which are called the scripting tags. All code inside these closing and ending scripting tags states that it would be rendered as a deluge script. Every deluge script can be thus considered as <% *deluge script* %>.

In figure 5.16, we have used a "for each" statement which is nothing but an iterator which basically iterates through each record in the provided form-name. So, if we have 10 records in the referenced table, then the "for each" statement would iterate 10 times. The template of a "for each" statement is: for each <row-variable> in <formname>. So we can observe that in the example in figure 5.16, the <row-variable> is "player" and the <formname> is "BruinsRoster". The <row-variable> can be named anything and it will be assigned a single record in every iteration of the "for each" loop. The <formname> has to be an existing name of a form which we have already created to insert data.

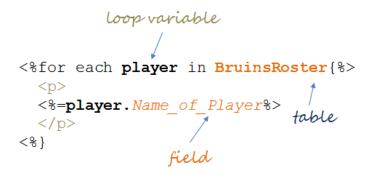


FIGURE 5.17 DELUGE FOR EACH LOOP

Along with the script, we need the HTML tags to display the data on the webpage when it is rendered. So, we have mix the deluge script with appropriate HTML tags. Look how we have use the paragraph tag to display the name of players.

Access data from database

To access the data from the ZohoDB, we would need the reference of the loop variable or the *<row-variable>* "player". In every "for each" iteration, the "player" variable will have all data related to a single player and to access each field we need to use a dot (.) and the field-name. For example "player.Name_of_Player" in figure 5.16 fetches the name of the player. We can see that in the output the player names are displayed.

Finding Matching Records

We can also find a particular record from the data table. This statement in figure 5.18 finds the position of the player whose name is "Tuuka Rask" and places that value in the variable *playerPos*. Remember that each player record we inserted had a "Position" field.



FIGURE 5.18 FINDING MATCHING RECORDS

Alternately, we can retrieve Tuuka Rask's whole record as well. To do so, we can just say *player* = BruinsRoster[Name_of_Player == "Tuuka Rask"]. The variable *player* will now have the whole record with information about Tuuka.

After we have the position value in variable *playerPos*, we can use it in the script that follows the statement. For example, if we have to display this on th webpage, we can simply embed it inside a HTML tag like: <h2><%=playerPos%></h2>.

Deluge Mode vs HTML

The Zoho Creator scripts have two modes:

- **HTML mode:** Allow you to use HTML tags, e.g., , , <h3>, etc. Switch to HTML mode by exiting Deluge mode. Exit Deluge mode with %>.
- **Deluge mode:** Allow you to write scripting code, e.g. for, {, }, if, etc. Enter Deluge mode with <%.

So, basically Zoho has only these two modes and you can create dynamic pages by using them.

A Complete Example

```
1 htmlpage CustomRoster1()
2 displayname = "CustomRoster"
3 content
4 <%{%>
5 <h3>Boston Bruins Goalies</h3>
6 
7 
   <%for each player in BruinsRoster</pre>
9
    {%>
  10
  <%=player.Picture%>
11
12 
   <%=player.Name_of_Player%>
13
   14
   15
                     <%}%>
16
   17
18
19 <%}%>
```

FIGURE 5.19 EXAMPLE OF A COMPLETE DELUGE SCRIPT

Boston Bruins Goalies



Tuuka Rask



Malcolm Subban

FIGURE 5.20 DISPLAYED AS WEBPAGE

Embedding the Dynamic Page

Following the sections till now you can easily create a working dynamic web page. Zoho also provided the option to embed these dynamic pages into any HTML code or Weebly webpage.

This can be done by getting the embedded code from Zoho. For doing so, select "Settings" and "Publish", then select the embed code. Figure 5.21 is an example embedded code.

<iframe height='500px' width='100%' frameborder='0' allowTransparency='true'
scrolling='auto' src='https://creator.zohopublic.com/mjs01776/sandbox/viewembed/CustomRoster1/dSRRO4tTzWvXaUhrrOMktMP2697eq2dsXCDPYhafBJbt9
593CkQnekrfBw0s3TZKYsbAUv5HrEEuS19QBV216zdV4vrPjNXBnM6f'></iframe>

FIGURE 5.21 EMBED < IFRAME > FOR PUBLISH

The embed *<iframe>* provides a direct link to the dynamic page that you created. So, you don't need to re-embed the *<iframe>* if you make a change to the dynamic page; it is automatically regenerated from the source *<iframe>*.

Dynamic Pages in Summary

The figure below summarizes the whole module in one piece. First, start with form builder and create the structure for your database. Then insert data into the database using the reports. Finally, create scripts and HTML using Deluge scripting to produce a dynamic webpage. This page is dynamic because the HTML seen by a user on a browser is never static. The page is always generated by using the data from the database.



FIGURE 5.22 FULL CYCLE OF DYNAMIC PAGE GENERATION USING ZOHO CREATOR

References & Bibliography

These are excellent resources to learn more about business analysis and requirements management. Be sure to read the author's blog at www.arsdoceo.com.

