



**CATHOLIC UNIVERSITY OF RWANDA**

P.O. Box 49 BUTARE / HUYE - RWANDA  
TEL: (00250) 0252 530 893 FAX: (00250) 0252 530 627  
E-mail: [administration@cur.ac.rw](mailto:administration@cur.ac.rw)  
Web: [www.cur.ac.rw](http://www.cur.ac.rw)

---

**FACULTY OF SCIENCE AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE**

**MODULE OF WEB APPLICATION & SITE  
ADMINISTRATION**

**‘SEMESTER I’**

**LEVEL TWO COMPUTER SCIENCE**

**WASA 2523, CREDITS: 15, HOURS: 75**

**ACADEMIC YEAR 2018-2019**

**LECTURER: NSENGIYUMVA Jean Marie Vianney**

**Masters in Information Systems**

**Email : [nsengiyumva@gmail.com](mailto:nsengiyumva@gmail.com)**

## **Learning Outcomes**

### **Learning and Teaching Strategies**

Lectures

Group work

Assignments

Practical works

## Contents

CHAP.1 INTRODUCTION TO INTERNET .....	5
1.1 THE INTERNET .....	5
1.2. HOW TO CONNECT TO THE INTERNET .....	8
1.3. INTERNET SERVICES .....	9
1.4. INTERNET ETIQUETTE.....	9
1.5. THE INTERNET PROTOCOLS.....	10
1.6. TYPES OF INTERNET ACCESS.....	12
1.8. THE WORLD WIDE WEB .....	15
1.8 CREATING A SEARCH STATEMENT .....	19
CHAP 2: HYPER TEXT MARKUP LANGUAGE (HTML) .....	24
2.1. DEFINITION.....	24
2.2 HTML TAGS.....	24
2.3. HTML DOCUMENTS .....	25
2.3 GETTING STARTED-THE TOOLS .....	25
2.4.HTML ELEMENTS .....	26
2.5. FORMATTING TAGS.....	28
2.6. HTML COMMENT TAGS.....	29
2.7 HTML FONTS.....	30
2.8. HTML Styles - CSS.....	30
2.9 .HTML LINKS .....	32
2.10 HTML TABLES.....	34
CHAP 3.INTRODUCTION TO CASCADING STYLE SHEETS .....	46
3.1 DEFINITIONS .....	46
3.2 THE BASIC CSS SYNTAX.....	46
3.3.FONTS.....	51
3.5.STYLING LINKS .....	54
3.6 LISTS .....	55
3.7 CSS TABLES .....	56
3.8 CSS BOX MODEL.....	58
3.9 IDENTIFICATION AND GROUPING OF ELEMENTS .....	63
3.10 CSS GROUPING AND NESTING SELECTORS.....	65

CHAP 4. JAVASCRIPTS .....	67
4.1. INTRODUCTION.....	67
4.2. BROWSER SUPPORT FOR JAVASCRIPT .....	69
4.3. STATEMENTS .....	71
4.4. JAVASCRIPT CODE .....	72
4.5 VARIABLES .....	73
CHAP 5. INTRODUCTION TO PHP .....	79
5.1 DEFINITION.....	80
5.2 WHAT CAN PHP DO?.....	80
5.3 HISTORY.....	81
5.4 PHP SCRIPTS .....	81
5.5 VARIABLES .....	82
5.6 VARIABLE VARIABLES .....	82
5.7 OPERATORS .....	82
5.8.CONTROL STRUCTURES.....	83
5.9 ARRAYS.....	84

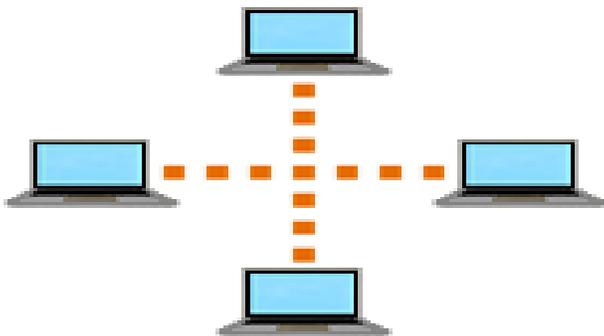
## CHAP.1 INTRODUCTION TO INTERNET

### 1.1 THE INTERNET

- The **Internet** is a worldwide collection of networks that links millions of businesses, government agencies, educational institutions, and individuals. Internet is a network of computer networks. Global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols. The internet is the largest computer network in the world, connecting millions of computers. A network is a group of two or more computer systems linked together. The Internet is similar to the roadway example. Remember, the Internet is a physical network (phone lines, etc). Data has to travel through that physical network. Too much traffic on the network means you may be unable to connect at a fast rate, or at all. The more open and wider your network is, the faster you can connect and surf.

#### 1.1.1 There are two main types of computer networks

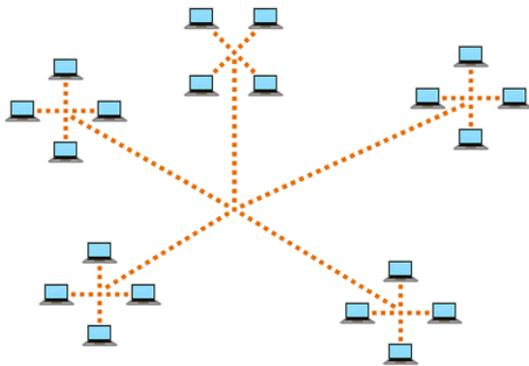
**Local Area Network (LAN):** A LAN is two or more connected computers sharing certain resources in a relatively small geographic location, often in the same building. Examples include home networks and office networks.



#### A Local Area Network (LAN)

**Wide Area Network (WAN):** A WAN typically consists of two or more LANs. The computers are farther apart and are linked by telephone lines, dedicated telephone lines, or radio waves. The internet is the largest Wide Area Network (WAN) in existence.

## **A Wide Area Network (WAN)**



### **1.1.2 .COMPONENTS OF THE INTERNET**

World Wide Web, E-Mail, Telnet, FTP, E-Mail Discussion Groups, Usenet News, FAQ, RFC, FYI, Chat & Instant Messaging.

### **1.1.3. Internet features**

Using Internet, organizations all over the world can exchange data, people can communicate with each other in a faster and effective way, Researchers can gather information in their respective area of research. With help of video conferencing over Internet, it has become possible that people can even see each other while communicating Even one can do all his shopping sitting back at home ( electronic commerce).

### **1.1.4. Internet History**

In 1969, the U.S. Defense Department funded a project to develop a network. Basically the idea was to develop a very secure network which can work even after a nuclear attack. This project was known as ARPANET. The proposed network was not supposed to have a central control which would be an obvious target

#### **Two main goals:**

- Allow scientists at different physical locations to share information and work together.

- Function even if part of the network were disabled or destroyed by a disaster.

Ten years of research brought Local Area Ethernet Networks (LANs) and workstations were developed to get connected to LAN. Computers connected to ARPANET used a standard or rule to communicate with each other with NCP (National Control Protocol). Protocol is a network term used to indicate the standard used by a network for communication. Rapid change in information technology suppressed NCP and brought TCP/IP (Transmission Control Protocol/Internet Protocol) in to the world of networking.

- **1986** NSF connects NSFnet to ARPANET and becomes known as the Internet
- **1995** NSFNet terminates its network on the Internet and resumes status as research network
- **1996** Internet2 is founded.
  - Each organization is responsible only for maintaining its own network
  - The World Wide Web Consortium (W3C) oversees research and sets guidelines and standards
  - Internet2 connects more than 200 universities and 115 companies via a high-speed private network.
  - It provides freedom of communications.
  - Internet is termed by some people as the world of largest democracy with no government. It has no state of head to control it.
  - The Internet is a rare example of a large democracy with no state of head, no official censors, no bosses, no board of directors. Nobody controls the Internet and in principle, any computer can speak to any other computer, as long as it obeys the technical rules of the TCP/IP protocol.
  - This freedom of Internet helped it to move out of its original base in military and research institutions, into elementary and high schools, colleges, public libraries, commercial sectors.

## 1.2. HOW TO CONNECT TO THE INTERNET

- Many home and small business users connect to the Internet via high-speed broadband Internet service
- Cable Internet service
- DSL
- Fiber to the Premises (FTTP)
- Fixed wireless
- Cellular Radio Network
- **Wi-Fi**
- Satellite Internet Service
- An **access provider** is a business that provides individuals and organizations access to the Internet free or for a fee
- **ISP (Internet service provider)**- Regional ISPs provide Internet access to a specific geographical area, National ISPs provide Internet access in cities and towns nationwide
- **Online service provider (OSP)** has many members-only features
- Popular OSPs include AOL (America Online) and MSN (Microsoft Network)
- **Wireless Internet service provider:** Provides wireless Internet access to computers and mobile devices
- May require a wireless modem
- An **IP address** is a number that uniquely identifies each computer or device connected to the Internet.
- A **domain name** is the text version of an IP address
- Top-level domain (TLD)
- A **DNS server** translates the domain name into its associated IP address



### 1.3. INTERNET SERVICES

- Internet mail is (e-mail or electronic mail), much faster as compared to normal postal mail.
- One can also send software and certain forms of compressed digital image as an attachment.
- News groups or discussion groups facilitate Internet user to join for various kinds of debate, discussion and new sharing.
- Long-distance computing was an original inspiration for development of ARPANET and does still provide a very useful service on Internet.
- Programmers can maintain accounts on distant, powerful computers, execute programs.
- File transfers service allows Internet users to access remote machines and retrieve programs, data or text.

### 1.4. INTERNET ETIQUETTE

- A core competency of public health informatics is to increase the effective use of information technology. Increasing, the internet is being used to disseminate health information and more and more people are turning to the internet for health information advice and services.
- It is important to ensure the presentation and receipt of that information is relayed appropriately.
- "Internet or Net Etiquette" is about the manners we use on the Internet/cyberspace. It is concerned with matters of courtesy in communications and guides us in proper behavior on the Internet.

- The concept began in the 1970's at Xerox, who developed e-mail etiquette to help employees learn to handle the new e-mail system. They successfully spread email culture throughout the company in the form of set guidelines and "best-practices" for using e-mail.

➤ **Core Rules to Netiquette**

Rule 1: Remember the human

Rule 2: Adhere to the same standards of behavior online that you follow in real life.

Rule 3: Know where you are in cyberspace

Rule 4: Respect other people's time

Rule 5: Make yourself look good online

Rule 6: Share expert knowledge

Rule 7: Help keep flame wars under control

Rule 8: Respect other people's privacy

Rule 9: Don't abuse your power

Rule 10: Be forgiving of other people's mistakes

Rule 11: All Caps is considered "shouting"

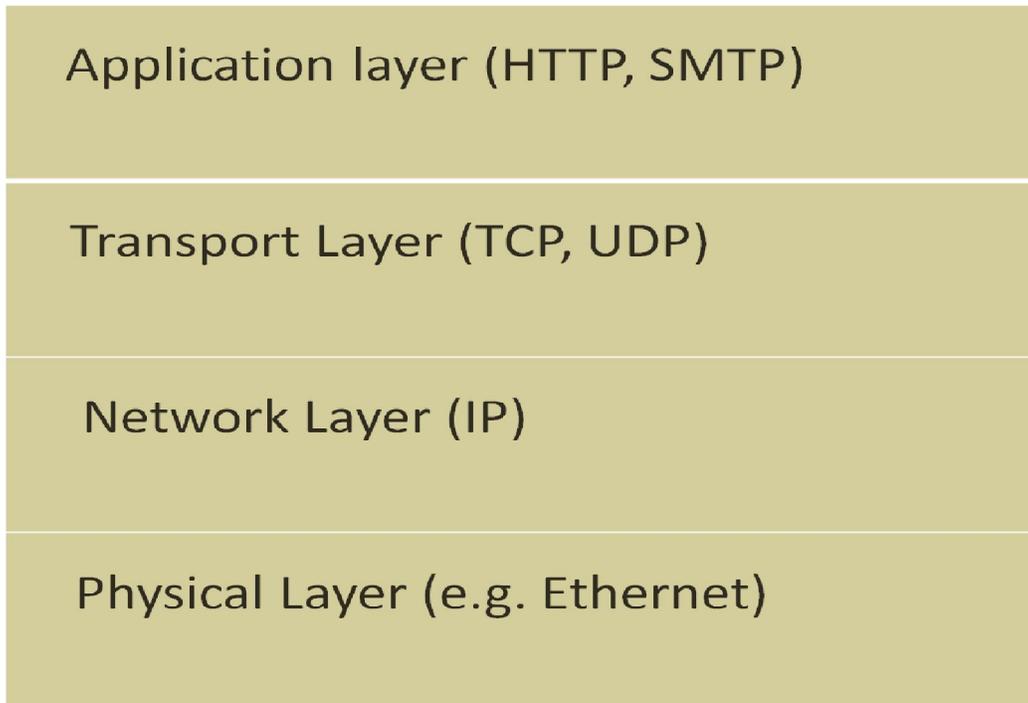
➤ **Violations of Internet Etiquette**

In the internet world, just as the real world, there are people who commit ultimate violations of guidelines of netiquette.

## **1.5. THE INTERNET PROTOCOLS**

The Internet Layered Architecture: as a network of connecting many small networks, consists of four layers:

- Application Layer (HTTP, SMTP.)
- Transport Layer (TCP, UDP)
- Network Layer (IP)
- Physical Layer



**FTP: (File transfer protocol):** is one of the most popular protocols to be used to move files on the Internet.

**TCP/IP: (Transmission Control Protocol and Internet Protocol)**

The low-level communications protocol that holds the Internet together. It provides means to allow two software on different machines on the Internet find each other, rendezvous, and transfer data.

It provides the essential service of making sure that each piece of data is transferred in the correct sequence and without error.

- **SMTP: (the e-mail message protocol):** A protocol to allow two users to communicate through e-mail messages over the Internet.

- **NNTP: (Net News Transfer Protocol):** A protocol, which can be used to access or transfer Usenet news over the Internet.

**Telnet:** - The traditional teletype-style communications protocol for communicating with text-based information services.

## **1.6. TYPES OF INTERNET ACCESS**

### **What Type of Access is Available to You?**

Many ISPs offer different levels of Internet access. The next few pages detail some different types of access.

#### **1.6.1. 56K Dial-up**

##### **Pros:**

- Uses existing phone lines.
- Lower cost-usually around \$22.00 per month.

##### **Cons:**

- Dial-up connection uses existing phone line, making it impossible to receive phone calls unless another line is installed. Uses existing phone lines.
- Slow -- Can be limited to speeds of 28.8K to 33.6K. Upload speeds can be limited to as low as 28.8, with download speeds approaching 56K under only perfect conditions.

Service can be somewhat unreliable (busy signals, spontaneously terminated sessions, etc).

#### **1.6.2. ISDN (Integrated Services Digital Network)**

Faster than 56K dial-up, but every ISP does not support ISDN.

##### **Pros:**

- Can provide 2-4 times the speed of a 56K modem.
- Uses digital rather than analog signals to transmit data.

##### **Cons:**

- Requires special equipment that can drive up your cost.
- Data is not compressed so transmission rates can be comparable to 56k-at more cost.

### **1.6.3 DSL (Digital Subscriber Line)**

Newer technologies such as DSL use an ISP to connect to the Internet and allow faster connections.

#### **Pros:**

- Can fall back to slower speeds if the line cannot handle the modem's fastest speed.
- Uses the existing phone line in most cases.
- Does not tie up existing phone line, leading to "always on" access.
- More bandwidth results in improved streaming audio/video, online games, application programs, telephone calling, video conferencing and other high-bandwidth services.
- Can connect multiple computers on a single line.
- Faster than a 56K dial-up modem.

#### **Cons:**

- Availability-limited to homes/businesses with a dedicated copper wire running between it and the phone company's nearest central office.
- Cost-includes installation fees and monthly charges (around \$50.00 per month).
- Need a firewall to ensure home network security. (additional cost)
- Service is not infallible.

### **1.6.4 Cable Modem**

Cable modems, another newer technology that uses an ISP to connect to the Internet, allows faster connections.

#### **Pros:**

- Uses coaxial and/or fiber-optic cable rather than phone lines for data transmission, so performance doesn't depend on your distance from a central cable office.
- Can fall back to slower speeds if the line cannot handle the modem's fastest speed.
- Convenient for homes already wired with cable access.
- Does not tie up existing phone line, leading to "always on" access.
- More bandwidth results in improved streaming audio/video, online games, application programs, telephone calling, video conferencing and other high-bandwidth services.
- Can connect multiple computers on a single line.
- Faster than a 56K dial-up modem.

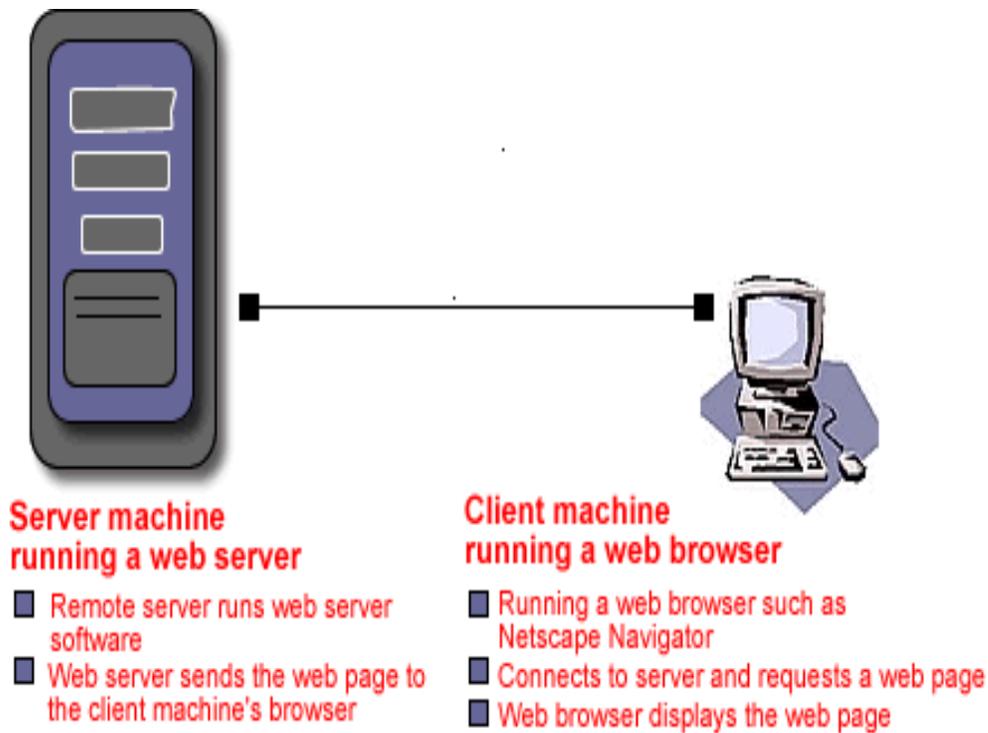
**Cons:**

- Availability-limited to homes/businesses with cable access.
- Cost-includes installation fees and monthly charges (around \$40.00 per month).
- Often requires commitment to a cable TV package.
- A firewall is necessary to ensure home network security (additional cost).

Service is not infallible

### **1.7. Servers and Clients Communicate**

- Your computer (client hardware) is running a web browser such as Internet Explorer (client software).
- When you want to surf the web, your browser connects to a remote server and requests a web page.
- The remote server (server hardware) runs web server software (server software).
- The web server sends the web page to your computer's web browser.
- Your web browser displays the page.



## 1.8. THE WORLD WIDE WEB

### 1.8.1. History

March, 1989, Tim Berners-Lee of Geneva's European Particle Physics Laboratory (CERN) circulated a proposal to develop a hypertext system for global information sharing in High Energy Physics community. The World Wide Web project began to take shape at the beginning of 1991. Before the end of 1991, CERN announced the Web to the High Energy Physics community in general. Essentially, 1992 was a developmental year. In March of 1993, WWW traffic clocked in at 0.1 percent of total Internet backbone traffic.

- **WWW** is the acronym for the World Wide Web.
- It is also commonly known as 'The Web'.
- The WWW is hypertext based information retrieval tool.
- One can easily surf the Web by jumping from one document to another using the links in those documents.

- These documents can be in many formats, such as text, graphics, animation, sound and latest is video.

What's the difference between the Internet and the World Wide Web? The Internet is a network of computers, cables, routers, and other hardware and software that interconnect and run on a network. The World Wide Web consists of documents that are transmitted across the Internet's hardware. The Web is made up of Web pages and Web sites. A Web page is a specially formatted document that can include text, graphics, hyperlinks, audio, animation, and video. A Web site is a collection of Web pages.

### **1.8.2. Web browsers**

A **browser** is a software program which interprets the HTML documents and displays it on the user's screen. The tool used to view the Web Pages on Internet is known as Internet browser or simply browser. It is a software program specifically developed to extract information on user request from the Internet and present them as a Web Page to the viewer. There are several browsers available in the market: The most popular are the following

Internet Explorer, Firefox, Opera, Safari, Google Chrome

- The process of using browser to view information on Internet is known as Browsing

#### **Web Addresses**

- Every page on the Internet has a unique address.
- This address is used to get the web page for user from Internet.
- The address on the Internet is known as URL (Uniform Resource Locator).
- A typical Internet address or URL would look like; <http://www.wda.gov.rw/curriculum/>
- The URL contains the components that specify the protocol, server, and pathname of an item

Uniform Resource Locators (URL)

- the addresses of Web resources.

Usually, an URL leads to a file, but that s not always the case.

- A URL can point you to a single record in a database, the front-end of an Internet program, or a result of a query.

*Example:*

*http://www.ibm.com/Features/Harlem/Harlem.html*



### **The World Wide Web: HTTP**

HTTP stands for Hypertext Transfer Protocol.

- It is a simple data transfer protocol that binds the Web together.
- It supports the communications between a web client (browser) and its web server.
- It consists of a set of messages and replies for both servers and browsers.
- It treats documents, files, menus, and graphics as objects.
- It relies on the Universal resource identifier (URI), enclosed in the universal resource locator (URL), to identify files.

- It uses the Internet's TCP/IP network protocol.

## **Search Engines**

- A Web search engine service consists of three components:
- **Spider:** Program that traverses the Web from link to link, identifying and reading pages
- **Index:** Database containing a copy of each Web page gathered by the spider
- **Search engine mechanism:** Software that enables users to query the index and that usually returns results in term relevancy ranked order
- A more sophisticated development in search engine technology is the ordering of search results by concept, keyword, site, links or popularity.
- Engines that support these features may be thought of as second generation search engines. These engines offer improvements in the ranking of results.
- **• Subject directory**
- – Classifies Web pages in an organized set of categories
- A search engine is helpful in locating items such as:
  - Images; Videos; Audio; Publications; Maps; People or Businesses; Blogs
- Directories are useful for general topics, for topics that need exploring, for in-depth research, and for browsing.
- There are two basic types of directories:

### **Academic and professional directories**

Often created and maintained by subject experts to support the needs of researchers

### **Commercial portals.**

That related to the general public and are competing for traffic. Be sure you use the directory that appropriately meets your needs.

### **Subject Directories**

- Internet Subject Directories.
- INFOMINE, from the University of California, is a good example of an academic subject directory
- Yahoo is a famous example of a commercial portal
- Examples of Specialized directories
- EXAMPLES OF SUBJECT-SPECIFIC DATABASES (i.e., VORTALS), Educator's Reference Desk (educational information)

### **1.8 CREATING A SEARCH STATEMENT**

- When structuring your query, keep the following tips in mind:
- Be specific

**EXAMPLE: CUR**

- Whenever possible, use nouns and objects as keywords

**EXAMPLE: Website**

Put most important terms first in your keyword list; to ensure that they will be searched, put a +sign in front of each one

**EXAMPLE: +hybrid +electric +gas +vehicles.**

- Use at least three keywords in your query

**EXAMPLE: web programming concepts**

- Combine keywords, whenever possible, into phrases

**EXAMPLE: "search engine tutorial"**

- Avoid common words, e.g., water, unless they're part of a phrase

**EXAMPLE: "bottled water"**

Think about words you'd expect to find in the body of the page, and use them as keywords

**EXAMPLE: Rwanda**

Write down your search statement and revise it before you type it into a search engine query box

### **Top Level Domains**

Only a few top-level domains are currently recognized, but this is changing. Here is a list of the domains generally accepted by all:

**.edu** : educational site (usually a university or college)

**.com** : commercial business site

**.gov** : U.S. governmental/non-military site

**.mil** :U.S. military sites and agencies

**.net** : networks, internet service providers, organizations

**.org** :U.S. non-profit organizations and others

**.museum** : restricted use by museums

**.name** :general use by individuals

**.pro** :restricted use by certified professionals and professional entities.

There is a large portion of the Web that search engine spiders cannot, or may not, index. It has been dubbed the "Invisible Web" or the "Deep Web" and includes, among other things, pass-protected sites, documents behind firewalls, archived material, the contents of certain databases, and information that isn't static but assembled dynamically in response to specific queries.

### **THE "INVISIBLE WEB"?**

There is a large portion of the Web that search engine spiders cannot, or may not, index. It has been dubbed the "Invisible Web" or the "Deep Web" and includes, among other things, pass-

protected sites, documents behind firewalls, archived material, the contents of certain databases, and information that isn't static but assembled dynamically in response to specific queries.

Data on the invisible Web is not inaccessible; the information is out there-it is stored on a Web server somewhere and can be accessed using a browser-but the data must be found using means other than the general-purpose search engines, such as Google and Yahoo!.

**Google.com indexes 40 billion public web pages.**

**100+ billion static web pages** are publicly-available. These pages can easily be found by Google and other search engines.

**11+ billion static pages are hidden from the public.** As private intranet content, these are the corporate pages that are only open to employees of specific companies.

**450+ billion database-driven pages are completely invisible to Google.** These invisible pages are not the regular web pages.

## **Home page**

A **home page**, **index page**, or **main page** is a page on a website. A Home Page usually refers to:

The initial or main web page of a website, sometimes called the "front page" (by analogy with newspapers).

The first page that appears upon opening a web browser program, which is also sometimes called the start page. This 'start page' can be a website or it can be a page with various browser functions such as the visual display of websites that are often visited in the web browser. The web page or local file that automatically loads when a web browser starts or when the browser's "home" button is pressed; this is also called a "home page". The user can specify the URL of the page to be loaded, or alternatively choose e.g. to re-load the most recent web page browsed. A personal web page, for example at a web hosting service or a university web site, that typically is stored in the home directory of the user.

In the 1990s the term was also used to refer to a whole web site, particularly a personal web site.

A home page can also be used outside the context of websites, such as to refer to the principal screen of a user interface, which is also referred to as a **home screen** on mobile devices such as cell phones.

### **Purpose of Home page**

Purpose of the website clearly stated on the homepage (e.g., mission statement, aim). Is just to introduce your company, let visitors know what you do, make a sale, sign them up, all of the preceding or something else. When visitors arrive at a website, they have some expectations, based on their experiences with other websites, as to how your website will work.

For example, they expect the home page to tell them the purpose of your website and what kinds of information are on the content pages.

Your home page will identify a number of paths that visitors may take to find that information by using menu bars, hyperlinks and other aids to searching.

### **Characteristics**

#### **1. Original, Fresh Content**

People visit websites for the primary purpose of finding content, so make sure you deliver.

Website content should be unique and up to date. Fresh content will keep visitors and search engines coming back for more.

#### **2. Target Audience**

A good website will have headlines and text that speaks to the target audience's needs and wants. Many websites simply list what their company does without saying how they can benefit their target audience.

#### **3. User-Friendly Navigation**

A good website has content that is easy to find. Pages should be organized and named in a way that the target audience will easily understand.

#### **4. Simple and Professional Design**

A good website will have an attractive layout that is easy on the eyes. Be sure your colors contrast well and your text doesn't require a magnifying glass to read.

Use design elements to draw attention to or to enhance the content of a page.

## **5. Speed**

Many factors can affect the loading time of a website including coding, number of graphics, the server speed, traffic volume on the website and the capabilities of a user's computer.

## **6. Search engine optimization**

Think about the keywords that users may search for to find a product or service you offer. Do some research to see how often those keywords are searched for through a tool like Google's Keyword Tool.

## **7. Link building**

Links are an important factor in determining where your website appears in search engine results.

## **8. Tracking**

A good website is a work in progress. A nice tool like Google Analytics will keep track of the number of people who come to your website, what pages they viewed, where they came from, what keywords they used in search engines, how many left after the first page and more.

## **. E-mail**

- **E-mail** is the transmission of messages and files via a computer network
- An **e-mail program** allows you to create, send, receive, forward, store, print, and delete e-mail messages
- A **mailing list** is a group of e-mail names and addresses given a single name
- **Subscribing** adds your e-mail name and address
- **Unsubscribing** removes your name

## **Asynchronous and Synchronous Communication**

**Synchronous** and **asynchronous transmissions** are two different methods of transmission synchronization. Synchronous transmissions are synchronized by an external clock, while asynchronous transmissions are synchronized by special signals along the transmission medium.

**Synchronous communication** is said to occur when two parties communicate in real-time. Examples of synchronous communication include telephone calls and two-way radio communication. In contrast, **asynchronous communication** is non real-time communication. Examples might be email, blog and message board postings, and especially text messaging. However due to the latency on digital networks, technically speaking, no communication is truly synchronous, as there will always be a small delay of at least a few milliseconds or more between parties. Whenever an electronic device transmits digital (and sometimes analogue) data to another electronic device, there must be a certain rhythm established between the two devices, i.e., the receiving device must have some way of knowing, within the context of the fluctuating signal that it's receiving, where each unit of data begins and where it ends.

## **CHAP 2: HYPER TEXT MARKUP LANGUAGE (HTML)**

### **2.1. DEFINITION**

- HTML is a language for describing web pages.
- HTML stands for Hyper Text Markup Language
- HTML is not a programming language, it is a markup language.
- A markup language is a set of markup tags
- HTML uses markup tags to describe web pages

### **2.2 HTML TAGS**

HTML markup tags are usually called HTML tags:

- HTML tags are keywords surrounded by angle brackets like <html>
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag
- Start and end tags are also called opening tags and **closing tags**

### 2.3. HTML DOCUMENTS

HTML documents describe web pages, HTML documents contain HTML tags and plain text, HTML documents are also called web pages. The purpose of a web browser (like Internet Explorer, Firefox, Google Chrome, safari, opera mini): Is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page:

#### Sample html document

```
<html><body><h1>My First Heading</h1><p>My first paragraph.</p></body> </html>
```

#### Example Explained

- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading

The text between <p> and </p> is displayed as a paragraph

### 2.3 GETTING STARTED-THE TOOLS

#### ➤ Editing HTML

HTML can be written and edited using many different editors like Dreamweaver and Visual Studio. However, in this class we use a plain text editor (like Notepad, edit plus2) to edit HTML. We believe using a plain text editor is the best way to learn HTML.

#### ➤ HTM or .HTML File Extension?

When you save an HTML file, you can use either the **.htm** or the **.html** file extension.

➤ **HTML Headings**

HTML headings are defined with the <h1> to <h6> tags.

➤ **HTML Paragraphs**

HTML paragraphs are defined with the <p> tag.

➤ **HTML Links**

HTML links are defined with the <a> tag. <a href="http://www.w3schools.com">This is a link</a> The link address is specified in the href attribute

➤ **HTML Images**

HTML images are defined with the <img> tag  The name and the size of the image are provided as attributes.

## **2.4.HTML ELEMENTS**

An HTML element is everything from the start tag to the end tag: The start tag is often called the opening tag. The end tag is often called the closing tag

➤ **html element syntax**

- An HTML element starts with a start tag / opening tag
- An HTML element ends with an end tag / closing tag
- The element content is everything between the start and the end tag
- Some HTML elements have empty content
- Empty elements are closed in the start tag
- Most HTML elements can have attributes

➤ **Nested HTML Elements**

- Most HTML elements can be nested (can contain other HTML elements).
  - HTML documents consist of nested HTML elements.

**<html>**

**<body>**

**<p>This is my first paragraph.</p>**

**</body>**

**</html>**

- The **<html>** element defines the whole HTML document.
  - The element has a start tag **<html>** and an end tag **</html>**.
  - The element content is another HTML element (the body element).
- The **<body>** element defines the body of the HTML document.
  - The element has a start tag **<body>** and an end tag **</body>**.
  - The element content is another HTML element (a p element).
- The **<p>** element defines a paragraph in the HTML document.
  - The element has a start tag **<p>** and an end tag **</p>**.
  - The element content is: This is my first paragraph.

### **Don't Forget the End Tag**

Some HTML elements might display correctly even if you forget the end tag:

**<p>This is a paragraph </p>**

**<p>This is a paragraph </p>**

The example above works in most browsers, because the closing tag is considered optional.

Never rely on this. Many HTML elements will produce unexpected results and/or errors if you forget the end tag .

### ➤ **Empty HTML Elements**

HTML elements with no content are called empty elements.

`<br>` is an empty element without a closing tag (the `<br>` tag defines a line break).

### ➤ **HTML Tip: Use Lowercase Tags**

HTML tags are not case sensitive: `<P>` means the same as `<p>`. Many web sites use uppercase HTML tags. The World Wide Web Consortium (W3C) **recommends** lowercase in HTML 4, and **demands** lowercase tags in XHTML.

### ➤ **HTML Attributes**

HTML elements can have **attributes**

- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value"**

#### **Attribute Example**

HTML links are defined with the `<a>` tag. The link address is specified in the **href** attribute:

```
<a href="http://www.w3schools.com">This is a link</a>
```

## **2.5. FORMATTING TAGS**

- HTML uses tags like `<b>` and `<i>` for formatting output, like bold or italic text.
- These HTML tags are called formatting tags
- HTML Lines

- The **<hr>** tag creates a horizontal line in an HTML page.
- The **hr** element can be used to separate content:
- Example

**<p>**This is a paragraph.**</p>**

**<hr>**

<b>Tag</b>	<b>Description</b>
<b>&lt;b&gt;</b>	Defines bold text
<b>&lt;em&gt;</b>	Defines emphasized text
<b>&lt;i&gt;</b>	Defines a part of text in an alternate voice or mood
<b>&lt;small&gt;</b>	Defines smaller text
<b>&lt;strong&gt;</b>	Defines important text
<b>&lt;sub&gt;</b>	Defines subscripted text
<b>&lt;sup&gt;</b>	Defines superscripted text
<b>&lt;ins&gt;</b>	Defines inserted text
<b>&lt;del&gt;</b>	Defines deleted text
<b>&lt;mark&gt;</b>	Defines marked/highlighted text

## **2.6. HTML COMMENT TAGS**

You can add comments to your HTML source by using the following syntax:

**<!--** Write your comments here **-->**

Comments are not displayed by the browser, but they can help document your HTML. With comments you can place notifications and reminders in your HTML:

### Example

```
<!-- This is a comment -->  
<p>This is a paragraph.</p>  
<!-- Remember to add more information here -->
```

## 2.7 HTML FONTS

- The HTML `<font>` Tag Should NOT be Used
- The `<font>` tag is deprecated in HTML 4, and removed from HTML5.
- The World Wide Web Consortium (W3C) has removed the `<font>` tag from its recommendations.
- In HTML 4, style sheets (CSS) should be used to define the layout and display properties for many HTML elements.
- The example below shows how the HTML could look by using the `<font>` tag:

```
<p>  
<font size="5" face="arial" color="red">  
This paragraph is in Arial, size 5, and in red text color.  
</font>  
</p>
```

## 2.8. HTML Styles - CSS

- CSS is used to style HTML elements.
- CSS was introduced together with HTML 4, to provide a better way to style HTML elements.
- CSS can be added to HTML in the following ways:

- in **separate style sheet files** (CSS files)
  - in the **style element** in the HTML head section
  - in the **style attribute** in single HTML elements
- **Using the HTML Style Attribute**
    - It is time consuming and not very practical to style HTML elements using the style attribute.
  - **The preferred way to add CSS to HTML, is to put CSS syntax in separate CSS files.**
    - However, in this HTML lecture we will introduce you to CSS using the style attribute. This is done to simplify the examples. It also makes it easier for you to edit the code and try it yourself.
  - The background-color property defines the background color for an element:

```
<html>
<body style="background-color:yellow;">
<h2 style="background-color:red;">This is a heading</h2>
<p style="background-color:green;">This is a paragraph.</p>
</body>
</html>
```

- The font-family, color, and font-size properties defines the font, color, and size of the text in an element:

```
<html>
<body>
<h1 style="font-family:verdana;">A heading</h1>
<p style="font-family:arial;color:red;font-size:20px;">A paragraph.</p>
</body>
```

</html>

- The font-family, color, and font-size properties make the old <font> tag obsolete.

#### ➤ HTML Style Example - Text Alignment

- The text-align property specifies the horizontal alignment of text in an element:

<html>

<body>

<h1 style="text-align:center;">Center-aligned heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

- The text-align property makes the old <center> tag obsolete

## 2.9 .HTML LINKS

- Links allow users to click their way from page to page.
- A hyperlink (or link) is a **word, group of words, or image** that you can click on to jump to a new document or a new section within the current document. When you move the cursor over a link in a Web page, the arrow will turn into a little hand.
- Links are specified in HTML using the <a> tag.
- The <a> tag can be used in two ways:
  - To create a link to another document, by using the href attribute
  - To create a bookmark inside a document, by using the name attribute

<a href="http://www.cur.ac.rw/">Visit Catholic University of Rwanda</a>

- The " Visit Catholic University of Rwanda " doesn't have to be text. It can be an image or any other HTML element.

#### ➤ HTML Links - The target Attribute

- The target attribute specifies where to open the linked document.
- The example below will open the linked document in a new browser window or a new tab

```
<a href="http://www.cur.ac.rw/" target="_blank"> Visit Catholic University of Rwanda </a>
```

- Create a link to the "Useful Tips Section" from another page:

```
<a href="http://www.w3schools.com/html_links.htm#tips">Visit the Useful Tips Section</a>
```

### ➤ **HTML Images**

#### **The <img> Tag and the Src Attribute**

- In HTML, images are defined with the <img> tag.
- The <img> tag is empty, which means that it contains attributes only, and has no closing tag.
- To display an image on a page, you need to use the src attribute.
  - Src stands for "source". The value of the src attribute is the URL of the image you want to display.
  -  The URL points to the location where the image is stored.

#### **Syntax for defining an image:**

An image named "boat.gif", located in the "images" directory on "www.w3schools.com" has the URL: <http://www.w3schools.com/images/boat.gif>. The browser displays the image where the <img> tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

### ➤ **The Alt Attribute**

- The required alt attribute specifies an alternate text for an image, if the image cannot be displayed.
- The value of the alt attribute is an author-defined text:

****

- The alt attribute provides alternative information for an image if a user for some reason cannot view it because of:
  - slow connection,
  - an error in the src attribute,
  - or if the user uses a screen reader

#### ➤ **Set Height and Width of an Image**

- The height and width attributes are used to specify the height and width of an image.
- The attribute values are specified in pixels by default

****

## **2.10 HTML TABLES**

- Tables are defined with the <table> tag.

A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). td stands for "table data," and holds the content of a data cell. A <td> tag can contain text, links, images, lists, forms, other tables, etc.

## Table Example

```
<table border="1">
  <tr>
    <td>row 1, cell 1</td>
    <td>row 1, cell 2</td>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
  <tr>
    <td></td>
    <td></td>
    <td></td>
  </tr>
</table>
```

- **Border Attribute**
- If you do not specify a border attribute, the table will be displayed without borders.
- To display a table with borders, specify the border attribute:

```
<table border="1">...</table>
```

### ➤ HTML Table Headers

- Header information in a table are defined with the <th> tag.
- All major browsers will display the text in the <th> element as bold and centered

```
<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
</table>
```

➤ **An HTML Table with a Border Attribute**

If you do not specify a border for the table, it will be displayed without borders.

A border can be added using the border attribute:

**Example**

```
<table border="1" style="width:300px">
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>
```

**2.11 HTML Lists**

- The most common HTML lists are ordered(numbered) and unordered (non-numbered) lists:

### ➤ HTML Unordered Lists

- An unordered list starts with the <ul> tag.
- Each list item starts with the <li> tag.
- The list items are marked with bullets (typically small black circles).

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

### ➤ HTML Ordered Lists

- An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.
- The list items are marked with numbers.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

### ➤ HTML Definition Lists

- A definition list is a list of items, with a description of each item.
  - The <dl> tag defines a definition list.
  - The <dl> tag is used in conjunction with <dt> (defines the item in the list) and <dd> (describes the item in the list):

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

- How the HTML code above looks in a browser:

**Coffee**

**- black hot drink**

**Milk**

**- white cold drink**

- Inside a list item you can put text, line breaks, images, links, other lists, etc

## 2.12. HTML Forms and Input

- HTML Forms are used to select different kinds of user input.
- HTML forms are used to pass data to a server.
- A form can contain input elements like
  - text fields,
  - checkboxes,
  - radio-buttons,
  - Submit buttons and more.
  - A form can also contain select lists, textarea, fieldset, legend, and label elements.
- The <form> tag is used to create an HTML form:
- The most important form element is the input element.
  - The input element is used to select user information.
  - An input element can vary in many ways, depending on the type attribute. An input element can be of type text field, checkbox, password, radio button, submit button, and more.

The most used input types are described below.

### ➤ Text Fields

`<input type="text" />` defines a one-line input field that a user can enter text into:

```
<form>First name: <input type="text" name="firstname" /><br />
Last name: <input type="text" name="lastname" />
</form>
```

### ➤ Password Field

`<input type="password" />` defines a password field:

```
<form>
Password: <input type="password" name="pwd" />
</form>
```

- **Note:** The characters in a password field are masked (shown as asterisks or circles).

### ➤ Radio Buttons

- `<input type="radio" />` defines a radio button.
  - Radio buttons let a user select ONLY ONE one of a limited number of choices:

```
<form>
<input type="radio" name="sex" value="male" /> Male<br />
<input type="radio" name="sex" value="female" /> Female
</form>
```

### ➤ Checkboxes

- `<input type="checkbox" />` defines a checkbox.
  - Checkboxes let a user select ONE or MORE options of a limited number of choices.

```
<form>
<input type="checkbox" name="vehicle" value="Bike" /> I have a bike<br />
<input type="checkbox" name="vehicle" value="Car" /> I have a car
</form>
```

### ➤ **Submit Button**

`<input type="submit" />` defines a submit button.

- A submit button is used to send form data to a server.
- The data is sent to the page specified in the form's action attribute.
- The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="html_form_action.asp" method="get"> Username: <input type="text" name="user" /> <input type="submit" value="Submit" /></form>
```

### ➤ **Textarea**

textarea is, basically, a large, multi-line textbox.

The anticipated number of rows and columns can be defined with rows and cols attributes.

```
<textarea rows="5" cols="20">A big load of text</textarea>
```

### ➤ **Select**

The select tag works with the option tag to make drop-down select boxes.

Example:

```
<select>
<option>Option 1</option>
<option>Option 2</option>
<option value="third option">Option 3</option>
</select>
```

## **2.13 HTML Frames**

- With frames, several Web pages can be displayed in the same browser window.
- With frames, you can display more than one HTML document in the same browser window.

- Each HTML document is called a frame, and each frame is independent of the others. The disadvantages of using frames are:
  - Frames are not expected to be supported in future versions of HTML
  - Frames are difficult to use. (Printing the entire page is difficult).
  - The web developer must keep track of more HTML documents

#### ➤ **The HTML frameset Element**

- The frameset element holds one or more frame elements.
  - Each frame element can hold a separate document.
- The frameset element states HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them

#### ➤ **The HTML frame Element**

- The <frame> tag defines one particular window (frame) within a frameset.
- In the example below we have a frameset with two columns.
  - The first column is set to 25% of the width of the browser window. The
  - second column is set to 75% of the width of the browser window.
  - The document "frame\_a.htm" is put into the first column,
  - and the document "frame\_b.htm" is put into the second column:

```
<frameset cols="25%,75%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
</frameset>
```

**Note:** The frameset column size can also be set in pixels (cols="200,500"), and one of the columns can be set to use the remaining space, with an asterisk (cols="25%,\*").

#### ➤ **Useful Tips**

- **Tip:** If a frame has visible borders, the user can resize it by dragging the border.
  - To prevent a user from doing this, you can add noresize="noresize" to the <frame> tag.
- **Note:** Add the <noframes> tag for browsers that do not support frames.

- **Important:** You cannot use the <body></body> tags together with the <frameset></frameset> tags! However, if you add a <noframes> tag containing some text for browsers that do not support frames, you will have to enclose the text in <body></body> tags!

### ➤ HTML Iframes

- An iframe is used to display a web page within a web page.

#### Syntax for adding an iframe:

<iframe src="URL"></iframe> The URL points to the location of the separate page.

#### Set Height and Width

- The height and width attributes are used to specify the height and width of the iframe.
- The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

#### Example

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

#### Remove the Border

- The frameborder attribute specifies whether or not to display a border around the iframe.
- Set the attribute value to "0" to remove the border:

#### Example

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

#### Use iframe as a Target for a Link

- An iframe can be used as the target frame for a link.
- The target attribute of a link must refer to the name attribute of the iframe:

#### Example

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>  
<p><a href="http://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

### ➤ HTML Layouts

- Web page layout is very important to make your website look good.
  - Design your webpage layout very carefully.

### ➤ Website Layouts

- Most websites have put their content in multiple columns (formatted like a magazine or newspaper).
- Multiple columns are created by using <table> or <div> tags.
  - Some CSS are normally also added to position elements, or to create backgrounds or colorful look for the pages.

### ➤ HTML Layouts - Using Tables

- The simplest way of creating layouts is by using the HTML <table> tag.
- The following example uses
  - a table with 3 rows and 2 columns
  - the first and last row spans both columns using the colspan attribute:

```
<html>  
<body>
```

```
<table width="500" border="0">  
<tr>  
<td colspan="2" style="background-color:#FFA500;">  
<h1>Main Title of Web Page</h1>  
</td>  
</tr>
```

```
<tr valign="top">  
<td style="background-color:#FFD700;width:100px;text-align:top;">  
<b>Menu</b><br />  
HTML<br />  
CSS<br />  
JavaScript  
</td>  
<td style="background-color:#EEEEEE;height:200px;width:400px;text-align:top;">  
Content goes here</td>
```

```
</tr>
<tr>
<td colspan="2" style="background-color:#FFA500;text-align:center;">
Copyright © 2016 W3Schools.com</td>
</tr>
</table>
</body>
</html>
```

```
<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 © 2011 W3Schools.com</div>
```

```
</div>
```

```
</body>
</html>
```

#### ➤ MARQUEE BEHAVIOR

- <td><MARQUEE BEHAVIOR=ALTERNATE SCROLLDELAY=55><FONT COLOR="#660033" SIZE="24">Welcome to CUR</FONT></MARQUEE> </td>
- <MARQUEE BEHAVIOR=SCROLL/SLide >Texte d'animation </MARQUEE>.

➤ **UPLOAD YOUR PICTURE**

<HTML>

<BODY bgcolor="blue">

<form action="upload\_photo.php" method="post" enctype="multipart/form-data">

<p>

Upload your picture :<br />

<input type="file" name="my\_picture" /><br />

<input type="submit" value="send picture" />

</p>

</form>

</BODY>

</HTML>

## CHAP 3.INTRODUCTION TO CASCADING STYLE SHEETS

### 3.1 DEFINITIONS

- Cascading Style Sheets were created to provide a powerful, yet flexible means for formatting HTML content. CSS works much like style sheets in a word processing program.
- CSS is a style language that defines layout of HTML documents. For example, CSS covers fonts, colours, margins, lines, height, width, background images, advanced positions and many other things.

CSS was a revolution in the world of web design. The concrete benefits of CSS include: Control layout of many documents from one single style sheet; more precise control of layout; apply different layout to different media-types (screen, print, etc.); numerous advanced and sophisticated techniques.

### 3.2 THE BASIC CSS SYNTAX

#### ➤ Style Rules

All style rules follow the same basic format:

```
selector { property1: value1; property2: value2; ...  
propertyN: valueN; }
```

Using **HTML** we could have done it like this:

```
<body bgcolor="#FF0000">
```

With **CSS** the same result can be achieved like this:

```
body {background-color: #FF0000;}
```

Note that the formatting of CSS rules is very exact and follows these guidelines:

- ✚ The selector is followed by the formatting property definitions, which are enclosed in braces (`{ }`).
- ✚ A colon separates each property/value pair. Note that values that include spaces should be enclosed in double quotes, as in the following example: **font-family: "Times New Roman";**
- ✚ Each property/value pair ends with a semicolon.

The *selector* is the elements that the style should be used on. The *properties* are all formatting properties of the selected elements that should be set to the associated *values*. A very simple example of a style rule follows:

```
h1 { color: Red; }
```

### ➤ **Applying CSS to an HTML document**

There are three ways you can apply CSS to an HTML document.

#### **Method 1: In-line (the attribute style)**

One way to apply CSS to HTML is by using the HTML attribute style

#### **Example:**

```
<html>
<head> <title>Example</title>
</head>
<body style="background-color: #FF0000;">
<p>This is a red page</p> </body>
</html>
```

#### **Method 2: Internal (the tag style)**

### ➤ **Using the <style> element**

The <style> element behaves like other HTML elements. It has a beginning and ending tag and everything in between is treated as a style definition.

A document's <style> section must appear inside the document's <head> section, although multiple <style> sections are permissible.

The <style> tag has the following, minimal format:

```
<style type="text/css">
... style definitions ...
</style>
```

### Example:

```
<html> <head> <title>Example</title> <style type="text/css"> body {background-color: #FF0000;} </style> </head> <body> <p>This is a red page</p> </body> </html>
```

### Method 3: External (link to a style sheet)

The recommended method is to link to a so-called external style sheet.

When creating a separate style sheet file, you do not need to include the <style> tags, only the definitions. For example, the following is an example style sheet file named mystyles.css:

```
/* mystyles.css - Styles for the main site */  
  
h1, h2, h3, h4 { color: blue; }  
  
h1 { font-size: 18pt; }  
  
h2 { font-size: 16pt; }  
  
h3 { font-size: 14pt; }  
  
h4 { font-size: 12pt; }
```

#### ➤ How to link an external CSS sheet to HTML

To link an external style sheet with a document, use the <link> tag in the <head> of the document to which you want the styles applied. The <link> tag has the following format when used to link a style sheet:

```
<link rel="stylesheet" type="text/css"  
href="url_to_style_sheet" />
```

For example, let's say that your style sheet is named **style.css** and is located in a folder named **style**. The situation can be illustrated like this:

The trick is to create a link from the HTML document (default.htm) to the style sheet (style.css). Such link can be created with one line of HTML code:

```
<link rel="stylesheet" type="text/css" href="style/style.css" />
```

The line of code must be inserted in the header section of the HTML code i.e. between the <head> and </head> tags. Like this:

```
<html> <head> <title>My document</title> <link rel='stylesheet' type='text/css' href='style/style.css' /> </head> <body> ... </html>
```

This link tells the browser that it should use the layout from the CSS file when displaying the HTML file. The really smart thing is that several HTML documents can be linked to the same style sheet. In other words, one CSS file can be used to control the layout of many HTML documents.

## 1. Colors and backgrounds

### Foreground color: the 'color' property

The color property describes the foreground color of an element.

For example, imagine that we want all headlines in a document to be dark red. The headlines are all marked with the HTML element `<h1>`. The code below sets the color of `<h1>` elements to red.

```
h1 { color: #ff0000; }
```

Colors can be entered as hexadecimal values as in the example above (`#ff0000`), or you can use the names of the colors ("red") or rgb-values (`rgb(255,0,0)`).

### ➤ The 'background-color' property

The background-color property describes the background color of elements.

The element `<body>` contains all the content of an HTML document. Thus, to change the background color of an entire page, the background-color property should be applied to the `<body>` element. You can also apply background colors to other elements including headlines and text.

#### Example:

```
body { background-color: #FFCC66; }
```

```
h1 { color: #990000;
```

```
background-color: #FC9804; }
```

### Background images [background-image]

The CSS property background-image is used to insert a background image. To insert the image of the butterfly as a background image for a web page, simply apply the background-image property to `<body>` and specify the location of the image.

```
body { background-color: #FFCC66;
        background-image: url("myimage.gif"); }
h1 { color: #990000;
      background-color: #FC9804; }
```

### Repeat background image [background-repeat]

Value	Description
background-repeat: repeat-x	repeated horizontally
background-repeat: repeat-y	repeated vertically
background-repeat: repeat	both X,Y
background-repeat: no-repeat	is not repeated

### Lock background image [background-attachment]

The property background-attachment specifies whether a background picture is fixed or scrolls along with the containing element.

Value	Description
Background-attachment: scroll	unlocked
Background-attachment: fixed	locked

```
body { background-color: #FFCC66;
        background-image: url("image.gif");
        background-repeat: no-repeat;
        background-attachment: fixed; }
h1 { color: #990000;
      background-color: #FC9804; }
```

### **Place background image [background-position]**

By default, a background image will be positioned in the top left corner of the screen. The property background-position allows you to change this default and position the background image anywhere you like on the screen.

#### **Value**

background-position: 2cm 2cm

background-position: 50% 25%

background-position: top right

### **Compiling [background]**

With background you can compress several properties and thereby write your style sheet in a shorter way which makes it easier to read.

**background: #FFCC66 url("butterfly.gif")no-repeat fixed right bottom;**

## **3.3.FONTS**

font-family

font-style

font-variant

font-weight

font-size

font

### **Font family [font-family]**

The property font-family is used to set a prioritized list of fonts to be used to display a given element or web page.

When you list fonts for your web site, you naturally start with the most preferred font followed by some alternative fonts.

It is recommended to complete the list with a generic font family. Example:

```
h1 { font-family: arial, verdana, sans-serif; }
```

```
h2 { font-family: "Times New Roman", serif; }
```

➤ Font style [font-style]

The property font-style defines the chosen font either in normal, italic or oblique.

The property font-variant is used to choose between normal or small-caps variants of a font. A small-caps font is a font that uses smaller sized capitalized letters (upper case) instead of lower case letters.

➤ Font weight [font-weight]

The property font-weight describes how bold or "heavy" a font should be presented. A font can either be **normal** or **bold**.

**EX:** p {font-family: arial, verdana, sans-serif;}

td {font-family: arial, verdana, sans-serif;

**font-weight: bold;}**

➤ Font size [font-size]

The size of a font is set by the property font-size.

h1 {font-size: **30px**;}

h2 {font-size: **12pt**;}

h3 {font-size: **120%**;}

p {font-size: **1em**;}

The units 'px' and 'pt' make the font size absolute, while '%' and 'em' allow the user to adjust the font size as he/she see fit. **To make your website accessible** for everybody, you should use adjustable units such as '%' or 'em'.

➤ **Compiling [font]**

p { **font: italic bold 30px arial, sans-serif; }**

### **3.4.Text**

text-indent

text-align

text-decoration

letter-spacing

## text-transform

### ➤ **Text indentation [text-indent]**

The property text-indent allows you to add an elegant touch to text paragraphs by applying an indent to the first line of the paragraph.

### ➤ **Text alignment [text-align]**

ext can either be aligned to the **left**, to the **right** or **centred**. In addition to this, the value **justify** will stretch each line so that both the right and left margins are straight.

Example:

```
th { text-align: right; }
```

```
td { text-align: center; }
```

```
p { text-align: justify; }
```

### ➤ **Text decoration [text-decoration]**

The property text-decoration makes it is possible to add different "decorations" or "effects" to text.

```
a { text-decoration:none;}
```

It can also be used to decorate text:

### **Example:**

```
h1 { text-decoration:underline;}
```

```
h2 { text-decoration:line-through;}
```

```
h3 { text-decoration:underline;}
```

### ➤ **Letter space [letter-spacing]**

The spacing between text characters can be specified using the property letter-spacing. The value of the property is simply the desired width.

```
h1 { letter-spacing: 6px; }
```

```
p { letter-spacing: 3px; }
```

### ➤ **Text transformation [text-transform]**

The text-transform property controls the capitalization of a text. You can choose to **capitalize**, use **uppercase** or **lowercase** regardless of how the original text is looks in the HTML code.



The background-color property specifies the background color for links:

### Example

```
a:link {background-color:#B2FF99;}
a:visited {background-color:#FFFF85;}
a:hover {background-color:#FF704D;}
a:active {background-color:#FF704D;}
```

## 3.6 LISTS

In HTML, there are two types of lists:

**unordered lists** - the list items are marked with bullets

**ordered lists** - the list items are marked with numbers or letters.

With CSS, lists can be styled further, and images can be used as the list item marker.

### Different List Item Markers

The type of list item marker is specified with the list-style-type property:

### Example

```
ul.a {list-style-type: circle;}
ul.b {list-style-type: square;}

ol.c {list-style-type: upper-roman;}
ol.d {list-style-type: lower-alpha;}
```

### An Image as The List Item Marker

To specify an image as the list item marker, use the list-style-image property:

### Example

```
ul
{
list-style-type: none;
padding: 0px;
margin: 0px;
}
ul li
{
background-image: url(sqpurple.gif);
background-repeat: no-repeat;
```

```
background-position: 0px 5px;
padding-left: 14px;
}
```

### Example explained:

For ul:

- Set the list-style-type to none to remove the list item marker
- Set both padding and margin to 0px (for cross-browser compatibility)

For all li in ul:

- Set the URL of the image, and show it only once (no-repeat)
- Position the image where you want it (left 0px and down 5px)

Position the text in the list with padding-left

### All CSS List Properties

Property	Description
----------	-------------

list-style : Sets all the properties in one declaration

list-style-image: an image as the list-item marker

list-style-position: Specifies if the list-item markers should appear inside or outside the content flow

list-style-type: Specifies the type of list-item marker

### 3.7 CSS TABLES

#### ➤ Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for table, th, and td elements:

#### Example

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

#### Collapse Borders

The border-collapse property sets whether the table borders are collapsed into a single border or separated:

### **Example**

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

#### ➤ **Table Width and Height**

Width and height of a table is defined by the width and height properties.

### **Example**

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

#### ➤ **Table Text Alignment**

The text in a table is aligned with the text-align and vertical-align properties.

### **Example**

```
td
{
text-align:right;
}
```

The vertical-align property sets the vertical alignment, like top, bottom, or middle:

## Example

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

## Table Padding

To control the space between the border and content in a table, use the padding property on td and th elements:

## Example

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

## Table Color

The example below specifies the color of the borders, and the text and background color of th elements:

## Example

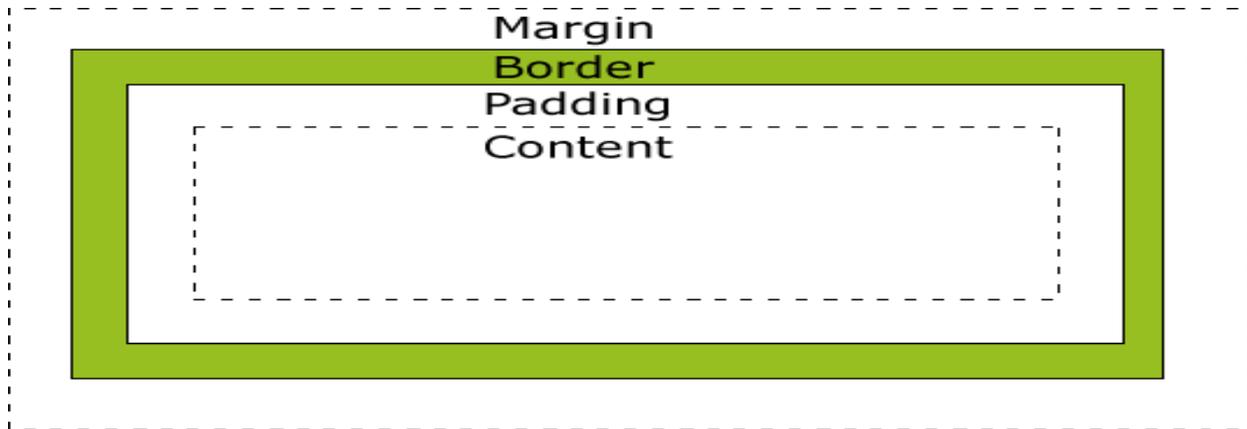
```
table, td, th
{
border:1px solid green;
}
th
{
background-color:green;
color:white;
}
```

## 3.8 CSS BOX MODEL

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to place a border around elements and space elements in relation to other elements.



- **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent
- **Border** - A border that goes around the padding and content. The border is inherited from the color property of the box
- **Padding** - Clears an area around the content. The padding is affected by the background color of the box
- **Content** - The content of the box, where text and images appear

➤ **Set the margin in an element**

An element has four sides: right, left, top and bottom. The margin is the distance from each side to the neighboring element (or the borders of the document).

**Example:**

```
body { margin-top: 100px;  
      margin-right: 40px;  
      margin-bottom: 10px;  
      margin-left: 70px; }
```

**or**

```
body { margin: 100px 40px 10px 70px; }
```

➤ **Set padding in an element**

Padding can also be understood as "filling". This makes sense as padding does not affect the distance of the element to other elements but only defines the inner distance between the border and the content of the element.

```
h1 { background: yellow;
      padding: 20px 20px 20px 80px; }
```

```
h2 { background: orange;
      padding-left:120px; }
```

### ➤ Borders

Borders can be used for many things, for example as a decorative element or to underline a separation of two things.

border-width

border-color

border-style

border

### ➤ The width of borders

The border-width property is used to set the width of the border.

The width is set in pixels, or by using one of the three pre-defined values: **thin, medium, or thick.**

- The "border-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.

### Example

```
p.one
{
border-style:solid;
border-width:5px;}
```

```
p.two
{
border-style:solid;
border-width:medium;
}
```

### The color of borders

The property border-color defines which color the border has.

## Example

```
p.one  
{  
border-style:solid;  
border-color:red;}
```

```
p.two  
{  
border-style:solid;  
border-color:#98bf21;  
}
```

## Types of borders

There are different types of borders to choose from.

The values none or hidden can be used if you do not want any border.



### Example

```
p
{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:dotted;
border-left-style:solid;
}
```

The border-style property can have from one to four values.

- **border-style:dotted solid double dashed;**

top border is dotted

right border is solid

bottom border is double

left border is dashed

- **border-style:dotted solid double;**

top border is dotted

right and left borders are solid

- **border-style:dotted solid;**

top and bottom borders are dotted

right and left borders are solid

- **border-style:dotted;**

all four borders are dotted

➤ **Border - Shorthand property**

The border property is a shorthand for the following individual border properties:

border-width

border-style (required)

border-color

bottom border is double

### 3.9 IDENTIFICATION AND GROUPING OF ELEMENTS

Sometimes you want to apply a special style to a particular element or a particular group of elements, use class and id to specify properties for selected elements.

➤ **Grouping elements with class**

**Example:**

```
<p>Grapes for white wine:</p>
```

```
<ul> <li><a href="ri.htm" class="whitewine">
```

```
Riesling</a></li>
```

```
<li><a href="ch.htm" class="whitewine">
```

```
Chardonnay</a></li> <li><a href="pb.htm"
```

```
class="whitewine">Pinot Blanc</a></li> </ul>
```

```
<p>Grapes for red wine:</p> <ul> <li>
```

```
<a href="cs.htm" class="redwine">
```

```
Cabernet Sauvignon</a></li> <li>
```

```
<a href="me.htm" class="redwine">Merlot</a></li>
```

```
<li><a href="pn.htm" class="redwine">
```

```
Pinot Noir</a></li> </ul>
```

```
class="whitewine">Pinot Blanc</a></li> </ul>
```

```
<p>Grapes for red wine:</p> <ul> <li>
```

```
<a href="cs.htm" class="redwine">
```

```
Cabernet Sauvignon</a></li> <li>
```

```
<a href="me.htm" class="redwine">Merlot</a></li>
```

```
<li><a href="pn.htm" class="redwine">
```

```
Pinot Noir</a></li> </ul>
```

➤ **Identification of element using id**

you might need to identify one unique element. This is done by using the attribute id.

What is special about the attribute id is that there cannot be two elements in the same document with the same id. Each id has to be unique.

**Example:**

```
<h1>Chapter 1</h1> ...
```

```
<h2>Chapter 1.1</h2> ...
```

```
<h2>Chapter 1.2</h2> ...
```

```
<h1>Chapter 2</h1> ...
```

```
<h2>Chapter 2.1</h2> ...
```

```
<h3>Chapter 2.1.2</h3> ...
```

The above could be headings of any document split into chapters or paragraphs. It would be natural to assign an id to each chapter as follows:

```
<h1 id="c1">Chapter 1</h1> ...
```

```
<h2 id="c1-1">Chapter 1.1</h2> ...
```

```
<h2 id="c1-2">Chapter 1.2</h2> ...
```

```
<h1 id="c2">Chapter 2</h1> ...
```

```
<h2 id="c2-1">Chapter 2.1</h2> ...
```

```
<h3 id="c2-1-2">Chapter 2.1.2</h3> ...
```

Let us say that the headline for chapter 1.2

must be in red. This can be done accordingly with CSS:

```
#c1-2
```

```
{ color: red; }
```

➤ **Grouping of elements (span and div)**

The elements `<span>` and `<div>` are used to group and structure a document and will often be used together with the attributes `class` and `id`.

## 1. Grouping with <span>

The element <span> is what you could call a neutral element which does not add anything to the document itself. But with CSS, <span> can be used to add visual features to specific parts of text in your documents.

Example:

```
<p>Early to bed and early to rise  
makes a man <span class="benefit">healthy</span>,  
<span class="benefit">wealthy</span>  
and <span class="benefit">wise</span>.</p>
```

The CSS belonging to it:

```
span.benefit {  
    color:red; }
```

You may also use id to add style to the <span>-elements. Just as long as you remember, that you'll have to apply a unique id to each of the three <span>-elements.

## 2.Grouping with <div>

Whereas <span> is used within a block-level element as seen in the previous example, <div> is used to group one or more block-level elements.

### 3.10 CSS GROUPING AND NESTING SELECTORS

#### ➤ Grouping Selectors

In style sheets there are often elements with the same style.

Example:

```
H1  
{  
color:green;  
}  
h2  
{  
  
color:green;  
}  
p  
{
```

```
color:green;
}
```

To minimize the code, you can group selectors.

Separate each selector with a comma.

### **Example**

```
h1,h2,p
{
color:green;
}
```

#### **➤ Nesting Selectors**

It is possible to apply a style for a selector within a selector.

In the example below, one style is specified for all p elements, one style is specified for all elements with class="marked", and a third style is specified only for p elements within elements with class="marked":

### **Example**

```
p
{
color: blue;
text-align: center;
}
```

```
.marked
{
background-color: red;
}
```

```
.marked p
{
color: white;}
}
```

## CHAP 4. JAVASCRIPTS

### 4.1. INTRODUCTION

JavaScript is the scripting language of the Web. It is used in billions of Web pages to add functionality, validate forms and communicate with the server,

#### What is javascript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language. A scripting language is a lightweight programming language
- usually embedded directly into HTML pages
- is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without **purchasing a license**

#### Uses of Javascript

1. JavaScript gives HTML designers a programming tool - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages.
2. JavaScript can react to events - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.
3. JavaScript can read and write HTML elements - A JavaScript can read and change the content of an HTML element.
4. JavaScript can be used to validate data - A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
5. JavaScript can be used to detect the visitor's browser - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
6. JavaScript can be used to create cookies - A JavaScript can be used to store and retrieve information on the visitor's computer

## How To USE JAVASCRIPT

- The HTML `<script>` tag is used to insert a JavaScript into an HTML page.
  - To insert a JavaScript into an HTML page, use the `<script>` tag.
- Inside the `<script>` tag use the type attribute to define the scripting language.
- The `<script>` and `</script>` tells where the JavaScript starts and ends:

### Example

```
<html>  
<body>  
<h1>My First Web Page</h1>  
<script type="text/javascript">  
    document.write("<p>" + Date() + "</p>");  
</script>  
</body>  
</html>
```

**NOTE:** Try to avoid using `document.write()` in real life JavaScript code. The entire HTML page will be overwritten if `document.write()` is used inside a function, or after the page is loaded.

### Changing HTML Elements

- The example below writes the current date into an existing `<p>` element.

```
<html>  
<body>  
<h1>My First Web Page</h1>  
<p id="demo"></p>  
<script type="text/javascript">  
    document.getElementById("demo").innerHTML=Date();  
</script>  
</body>  
</html>
```

- **Note:** To manipulate HTML elements JavaScript uses the DOM method `getElementById()`. This method accesses the element with the specified id.
- In this case the browser will replace the content of the HTML element with `id="demo"`, with the current date.

Note that the JavaScript is placed at the bottom of the page to make sure it is not executed before the `<p>` element is created.

#### 4.2. BROWSER SUPPORT FOR JAVASCRIPT

- Browsers that do not support JavaScript, will display JavaScript as page content.
- To prevent them from doing this, and as a part of the JavaScript standard,
  - the HTML comment tag should be used to "hide" the JavaScript.
- Just add
  - an HTML comment tag `<!--` before the first JavaScript statement,
  - and a `-->` (end of comment) after the last JavaScript statement, like this

```
<html>
<body>
<script type="text/javascript">
<!--
      document.getElementById("demo").innerHTML=Date();
//-->
</script>
</body>
</html>
```

The two forward slashes at the end of comment line (`//`) is the JavaScript comment symbol. This prevents JavaScript from executing the `-->` tag.

#### Where To Put

- JavaScripts can be put in the `<body>` and in the `<head>` sections of an HTML page

#### JavaScript Functions and Events

- JavaScripts in an HTML page will be executed when the page loads. This is not always what we want.

- Sometimes we want to execute a JavaScript when an **event** occurs, such as when a user clicks a button.
  - When this is the case we can put the script inside a **function**.
- Events are normally used in combination with functions (like calling a function when an event occurs).

### JavaScript in <head>

- The example below calls a function when a button is clicked:

- **Example**

```

<html>
<head>
<script type="text/javascript">
function displayDate()
{
    document.getElementById("demo").innerHTML=Date();
}
</script>
</head>
<body>
<h1>My First Web Page</h1>
<p id="demo"></p>
<button type="button" onclick="displayDate()">Display Date</button>
</body>
</html>

```

Scripts in <head> and <body>

- You can place an unlimited number of scripts in your document,
- and you can have scripts in both the body and the head section at the same time.
- It is a common practice to put all
  - functions in the head section,
  - or at the bottom of the page.
- This way they are all in one place and do not interfere with page content.

## Using an External JavaScript

- JavaScript can also be placed in external files.
- External JavaScript files often contain code to be used on several different web pages.
- External JavaScript files have the file extension .js.
- **Note:** External script cannot contain the <script></script> tags!
- To use an external script, point to the .js file in the "src" attribute of the <script> tag:
- **Example**

```
<html>
<head>
<script type="text/javascript" src="xxx.js"></script>
</head>
<body>
</body>
</html>
```

### 4.3. STATEMENTS

- JavaScript is a sequence of statements to be executed by the browser.  
JavaScript Statements
- A JavaScript statement is a command to a browser. The purpose of the command is to tell the browser what to do.
- This JavaScript statement tells the browser to write "Hello Dolly" to the web page:
  - document.write("Hello Dolly");
- It is normal to add a semicolon at the end of each executable statement.
  - The semicolon is optional (according to the JavaScript standard),
  - the browser is supposed to interpret the end of the line as the end of the statement.
- Because of this you will often see examples without the semicolon at the end.
- **Note:** Using semicolons makes it possible to write multiple statements on one line.

#### 4.4. JAVASCRIPT CODE

- JavaScript code (or just JavaScript) is a sequence of JavaScript statements.
- Each statement is executed by the browser in the sequence they are written.
- This example will write a heading and two paragraphs to a web page:

##### Example

```
<script type="text/javascript">  
document.write("<h1>This is a heading</h1>");  
document.write("<p>This is a paragraph.</p>");  
document.write("<p>This is another paragraph.</p>");  
</script>
```

##### JavaScript Blocks

- JavaScript statements can be grouped together in blocks.
- Blocks start with a left curly bracket {, and end with a right curly bracket }.
- The purpose of a block is to make the sequence of statements execute together.
- This example will write a heading and two paragraphs to a web page:
- Example

```
<script type="text/javascript">  
{  
    document.write("<h1>This is a heading</h1>");  
    document.write("<p>This is a paragraph.</p>");  
    document.write("<p>This is another paragraph.</p>");  
}  
</script>
```

- Normally a block is used to group statements together in a function
- or in a condition (where a group of statements should be executed if a condition is met).

##### JavaScript Comments

- Comments can be added to explain the JavaScript, or to make the code more readable.
- Single line comments start with //.
- The following example uses single line comments to explain the code:

### Example

```
<script type="text/javascript">
// Write a heading
document.write("<h1>This is a heading</h1>");
// Write two paragraphs:
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

### JavaScript Multi-Line Comments

- Multi line comments start with `/*` and end with `*/`.

### Using Comments to Prevent Execution

**Comments can be used to prevent the execution of a single code line (can be suitable for debugging) or a block of code**

## 4.5 VARIABLES

- Variables are "containers" for storing information
- variables can be used to hold values ( $x=5$ ) or expressions ( $z=x+y$ )

### Rules for JavaScript variable names:

- Variable names are case sensitive ( $y$  and  $Y$  are two different variables)
- Variable names must begin with a letter or the underscore character

### Declaring (Creating) JavaScript Variables

- Creating variables in JavaScript is most often referred to as "declaring" variables.
  - You declare JavaScript variables with the **var** keyword:

```
var x;
```

```
var carname;
```

- After the declaration shown above, the variables are empty (they have no values yet).
- However, you can also assign values to the variables when you declare them:

```
var x=5;
```

```
var carname="Volvo";
```

- After the execution of the statements above, the variable **x** will hold the value **5**, and **carname** will hold the value **Volvo**.
- **Note:** When you assign a text value to a variable, use quotes around the value.

- **Note:** If you redeclare a JavaScript variable, it will not lose its value.

## 4.6. JAVASCRIPT VARIABLES AND SCOPE

### Local JavaScript Variables

- A variable declared within a JavaScript function becomes LOCAL and can only be accessed within that function. (the variable has local scope).
- You can have local variables with the same name in different functions,
  - because local variables are only recognized by the function in which they are declared.
- Local variables are destroyed when you exit the function.

### Global JavaScript Variables

- Variables declared outside a function become GLOBAL, and all scripts and functions on the web page can access it.
- Global variables are destroyed when you close the page.
- If you declare a variable, without using "var", the variable always becomes GLOBAL.
- If you assign values to variables that have not yet been declared, the variables will automatically be declared as global variables.
- These statements:

`x=5;`

`carname="Volvo";`

will declare the variables x and carname as global variables (if they don't already exist).

## 4.7 OPERATORS

- The assignment operator = is used to assign values to JavaScript variables.
- The arithmetic operator + is used to add values together.

• `y=5;`

`z=2;`

`x=y+z;` The value of x, after the execution of the statements above, is 7.

JavaScript Arithmetic Operators

- Arithmetic operators are used to perform arithmetic between variables and/or values.
- Given that `y=5`, the table below explains the arithmetic operators

## JavaScript Assignment Operators

- Assignment operators are used to assign values to JavaScript variables.

### The + Operator Used on Strings

- The + operator can also be used to add string variables or text values together.
- To add two or more string variables together, use the + operator.

```
txt1="What a very";
```

```
txt2="nice day";
```

```
txt3=txt1+txt2;
```

After the execution of the statements above, the variable txt3 contains "What a verynice day".

- To add a space between the two strings, insert a space into one of the strings:

- txt1="What a very ";

```
txt2="nice day";
```

```
txt3=txt1+txt2; or insert a space into the expression:
```

## Adding Strings and Numbers

- The rule is: **If you add a number and a string, the result will be a string!**
- **Example**

```
x=5+5;
```

```
document.write(x);
```

```
x="5"+"5";
```

```
document.write(x);
```

```
x=5+"5";
```

```
document.write(x);
```

```
x="5"+5;
```

```
document.write(x);
```

### Comparison Operators

- Comparison operators are used in logical statements to determine equality or difference between variables or values.
- If **x=5**

`==` is equal to     `x==8` is false   `x==5` is true

*=== is exactly equal to (value and type) x===5 is true: x==="5" is false*

*!= is not equal x!=8 is true*

*> is greater than x>8 is false*

*< is less than x<8 is true*

*>= is greater than or equal to x>=8 is false*

*<= is less than or equal to x<=8 is true*

### **How Can it be Used**

- Comparison operators can be used in conditional statements to compare values and take action depending on the result:

```
if (age<18) document.write("Too young");
```

### **Logical Operators**

- Logical operators are used to determine the logic between variables or values.
- Given that **x=6 and y=3**, the table below explains the logical operators:

- Operator    Description    Example
  - &&            and    (x < 10 && y > 1) is true
  - ||             or     (x===5 || y===5) is false
  - !              not    !(x==y) is true

### **Conditional Operator**

- JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

### **Syntax**

```
variablename=(condition)?value1:value2
```

### **Example**

```
greeting=(visitor=="PRES")?"Dear President ":"Dear ";
```

- If the variable **visitor** has the value of "PRES", then the variable **greeting** will be assigned the value "Dear President " else it will be assigned "Dear".

### **Conditional statements**

- Conditional statements are used to perform different actions based on different conditions.

### **Conditional Statements**

- Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.
- In JavaScript we have the following conditional statements:
  1. **if statement** - use this statement to execute some code only if a specified condition is true
  2. **if...else statement** - use this statement to execute some code if the condition is true and another code if the condition is false
  3. **if...else if...else statement** - use this statement to select one of many blocks of code to be executed
  4. **switch statement** - use this statement to select one of many blocks of code to be executed

### **If Statement**

```

<script type="text/javascript">
//Write a "Good morning" greeting if
//the time is less than 10
var d=new Date();
var time=d.getHours();
if (time<10)
{
  document.write("<b>Good morning</b>");
}
</script>

```

- Notice that there is no `..else..` in this syntax. You tell the browser to execute some code **only if the specified condition is true.**

## If...else Statement

```
<script type="text/javascript">
//If the time is less than 10, you will get a "Good morning" greeting.
//Otherwise you will get a "Good day" greeting.
var d = new Date();
var time = d.getHours();
if (time < 10)
{
    document.write("Good morning!");
}
else
{
    document.write("Good day!");
}
</script>
```

## If...else if...else Statement

```
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
{
    document.write("<b>Good morning</b>");
}
else if (time>10 && time<16)
{
    document.write("<b>Good day</b>");
}
else
{
    document.write("<b>Hello World!</b>");
}
```

```
}  
</script>
```

### **Switch Statement** : How it works

This is how it works:

- First we have a single expression  $n$  (most often a variable), that is evaluated once.
- The value of the expression is then compared with the values for each case in the structure.
- If there is a match, the block of code associated with that case is executed.
- Use **break** to prevent the code from running into the next case automatically.

### **Example**

```
    <script type="text/javascript">  
var d=new Date();  
var theDay=d.getDay();  
switch (theDay)  
{  
case 5:  
    document.write("Finally Friday");  
    break;  
case 6:  
    document.write("Super Saturday");  
    break;  
case 0:  
    document.write("Sleepy Sunday");  
    break;  
default:  
    document.write("I'm looking forward to this weekend!");  
}  
</script>
```

## 5.1 DEFINITION

- PHP is server side scripting system, Hypertext Preprocessor", Syntax based on Perl, Java, and C, Very good for creating dynamic content, Powerful, but somewhat risky!
- If you want to focus on one system for dynamic content, this is a good one to choose. Here we would like to show the very basics of PHP in a short, This text only deals with dynamic web page creation with PHP, though PHP is not only capable of creating web pages.

PHP stands for "**PHP: Hypertext Preprocessor**"

PHP enabled web pages are treated just like regular HTML pages and you can create and edit them the same way you normally create regular HTML pages.

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours. Although PHP's development is focused on server-side scripting, you can do much more with it.

## 5.2 WHAT CAN PHP DO?

PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more. There are three main areas where PHP scripts are used.

- Server-side scripting. This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through

the server. All these can run on your home machine if you are just experimenting with PHP programming. Command line scripting. You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks..

### 5.3 HISTORY

- ▶ Started as a Perl hack in 1994 by Rasmus Lerdorf (to handle his resume), developed to PHP/FI 2.0
- ▶ By 1997 up to PHP 3.0 with a new parser engine by Zeev Suraski and Andi Gutmans
- ▶ Version 5.2.4 is current version, rewritten by Zend ([www.zend.com](http://www.zend.com)) to include a number of features, such as an object model
- ▶ Current is version 5
- ▶ php is one of the premier examples of what an open source project can be

### 5.4 PHP SCRIPTS

```
<P>
```

```
<?php $myvar = "Hello World!";
```

```
echo $myvar;
```

```
?>
```

```
</P>
```

- ▶ Typically file ends in .php--this is set by the web server configuration
- ▶ Separated in files with the <?php ?> tag
- ▶ php commands can make up an entire file, or can be contained in html--this is a choice....
- ▶ Program lines end in ";" or you get an error
- ▶ Server recognizes embedded script and executes

- ▶ Result is passed to browser, source isn't visible

## 5.5 VARIABLES

Typed by context (but one can force type), so it's loose

Begin with "\$" (unlike javascript!)

Assigned by value

- `$foo = "Bob"; $bar = $foo;`

Assigned by reference, this links vars

- `$bar = &$foo;`

Some are preassigned, server and env vars

- For example, there are PHP vars, eg. `PHP_SELF`, `HTTP_GET_VARS`

## 5.6 VARIABLE VARIABLES

- Using the value of a variable as the name of a second variable)

```
$a = "hello";
```

```
$$a = "world";
```

- Thus:

```
echo "$a ${$a}";
```

- Is the same as:

```
echo "$a $hello";
```

- But \$\$a echoes as "\$hello"....

## 5.7 OPERATORS

- ▶ **Arithmetic (+, -, \*, /, %)** and **String (.)**

- ▶ Assignment (=) and combined assignment

```
$a = 3;
```

```
$a += 5; // sets $a to 8;
```

```
$b = "Hello ";
```

```
$b .= "There!"; // sets $b to "Hello There!";
```

- ▶ Bitwise (&, |, ^, ~, <<, >>)

- `$a ^ $b` (Xor: Bits that are set in `$a` or `$b` but not both are set.)

- `~ $a` (Not: Bits that are set in `$a` are not set, and vice versa.)

- ▶ Comparison (==, ===, !=, !==, <, >, <=, >=)

## 5.8.CONTROL STRUCTURES

- Wide Variety available
  - if, else, elseif
  - while, do-while
  - for, foreach
  - break, continue, switch
  - require, include, require\_once, include\_once

### ➤ SWITCH

Switch, which we've seen, is very useful

These two do the same things....

```
if ($i == 0) {  
  
    echo "i equals 0";  
  
} elseif ($i == 1) {
```

```
    echo "i equals 1";
} elseif ($i == 2) {
    echo "i equals 2";
}
switch ($i) {
case 0:
    echo "i equals 0";
    break;
case 1:
    echo "i equals 1";
    break;
case 2:
    echo "i equals 2";
    break;
}
```

## 5.9 ARRAYS

- ▶ You can create an array with the array function, or use the explode function (this is very useful when reading files into web programs...)

```
$my_array = array(1, 2, 3, 4, 5);
```

```
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
```

```
$pieces = explode(" ", $pizza);
```

- ▶ An array is simply a variable representing a keyed list
  - A list of values or variables
  - If a variable, that var can also be an array
  - Each variable in the list has a key
  - The key can be a number or a text label
  
- Arrays are lists, or lists of lists, or list of lists of lists, you get the idea--Arrays can be multi-dimensional
- Array elements can be addressed by either by number or by name (strings)
- If you want to see the structure of an array, use the `print_r` function to recursively print an array inside of pre tags

eg:

```
echo "$my_text_array[third]";
```

```
$my_text_array = array(first=>1, second=>2, third=>3);
```

```
echo "<pre>";
```

```
print_r($my_text_array);
```

```
echo "</pre>";
```

## ➤ **Multidimensional Arrays**

- A one dimensional array is a list, a spreadsheet or other columnar data is two dimensional...

- Basically, you can make an array of arrays

```
$multiD = array
```

```
(
```

```
"fruits" => array("myfavorite" => "orange", "yuck" => "banana", "yum" => "apple"),
```

```
"numbers" => array(1, 2, 3, 4, 5, 6),
```

```
"holes" => array("first", 5 => "second", "third")
```

```
);
```

- The structure can be built array by array, or declared with a single statement

- You can reference individual elements by nesting:

```
echo "<p>Yes, we have no " . $multiD["fruits"]["yuck"] . " (ok by me).</p>";
```

- `print_r()` will show the entire structure, but don't forget the pre tags