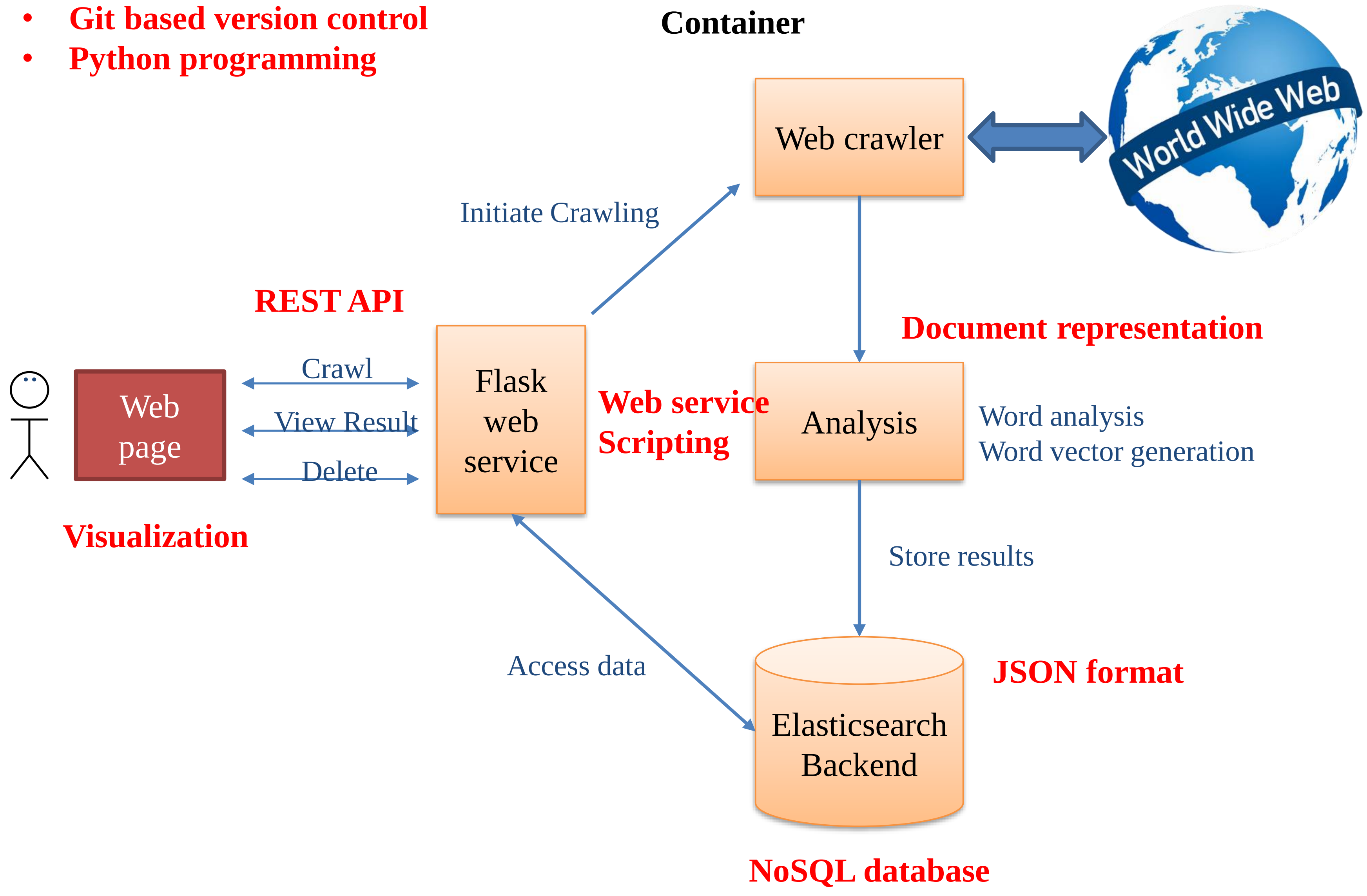
Open Source Programming

Lecture-06
Web Page

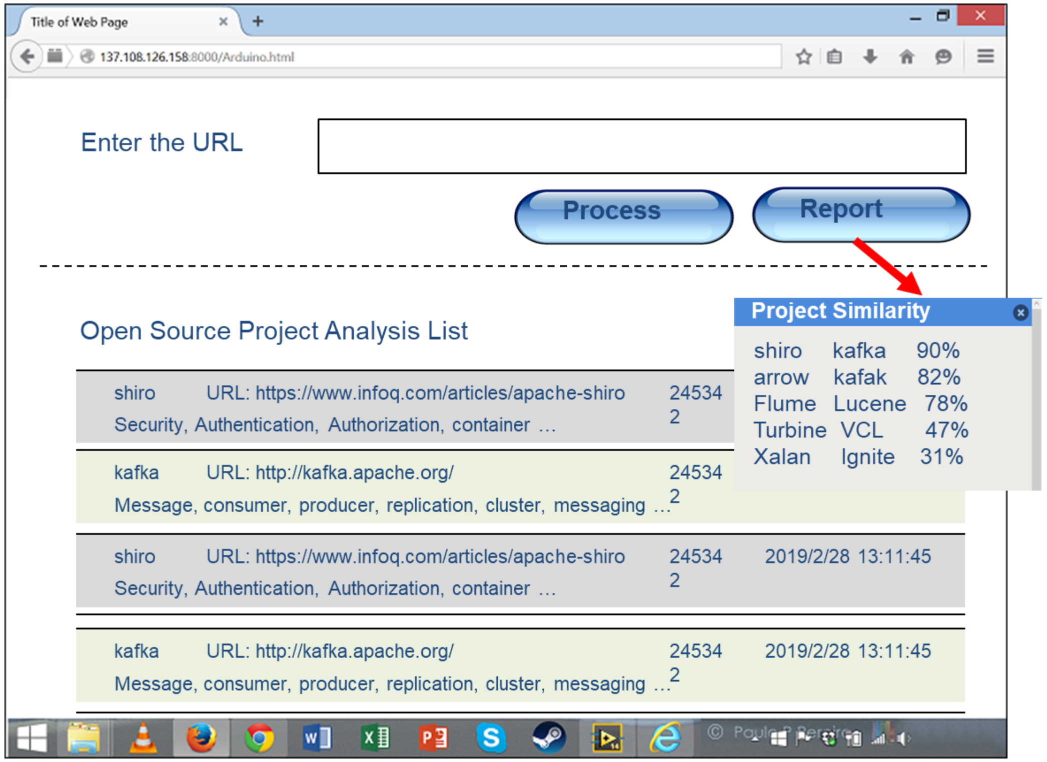
Components for the Project

- Git based version control
- Python programming



Project Example

- **Web Data Similarity Analysis**
  - Crawl web pages
  - Show a list of key words in the web page
  - Analyze and show the similarity of various web pages
  - Based on the well-known web framework, a system including a web page is built
  - Data must be systematically managed and maintained



Enter the URL

Open Source Project Analysis List

shiro	URL: https://www.infoq.com/articles/apache-shiro	24534 2	
Security, Authentication, Authorization, container …			
kafka	URL: http://kafka.apache.org/	24534 2	
Message, consumer, producer, replication, cluster, messaging …			
shiro	URL: https://www.infoq.com/articles/apache-shiro	24534 2	2019/2/28 13:11:45
Security, Authentication, Authorization, container …			
kafka	URL: http://kafka.apache.org/	24534 2	2019/2/28 13:11:45
Message, consumer, producer, replication, cluster, messaging …			

Project Similarity

shiro	kafka	90%
arrow	kafak	82%
Flume	Lucene	78%
Turbine	VCL	47%
Xalan	Ignite	31%

Project Examples

- 공모전 정보 서비스 시스템
- 관심 주식 정보 서비스 시스템
- 날씨/일기에 따른 음식 추천 시스템
- 유튜브 검색어 랭킹
- 세계뉴스 번역/요약 서비스

…

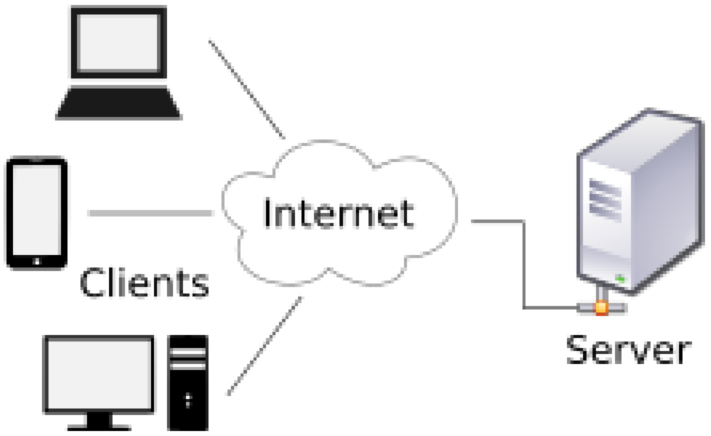
Contents

- WWW (World Wide Web)
- HTML
- XML
- JSON

WWW (World Wide Web)

How the Web Works

- **The Client-Server model**
  - Client and server operate on machines which are able to communicate through a network
  - The server waits for requests from a clients
  - Server receives a requests from a client
    1. Performs a the requested work
    2. Or lookup the requested data
    3. And send a response to the client
  - Servers:
    - file servers
    - web servers
    - name servers, etc.
  - Clients:
    - browsers
    - email clients, .etc.



URL Format

- **<scheme>://<server-domain-name>/<pathname>**

  - <scheme> which protocol to use
    - *http*: in general
    - *file*: which tells the client document is in a local machine
    - *ftp*: file transfer protocol
  - <server-domain-name> identifies the server system
    - i.e. **www.doc.gold.ac.uk**
  - <pathname> tells the server where to find the file

- **http://doc.gold.ac.uk/~username/index.html**

Web Servers & Browsers

- **Web Server**
  - Application which waits for client requests, fetches requested documents from disk and transmits them the client.
  - i.e Apache Server

- **Web Browser**
  - Program that can retrieve files from the WWW(world wide web) and render text, images, or sounds encoded in the files.
  - i.e. Chrome, Internet Explorer, Firefox, etc.

Edit Web Page

https://www.w3.org/

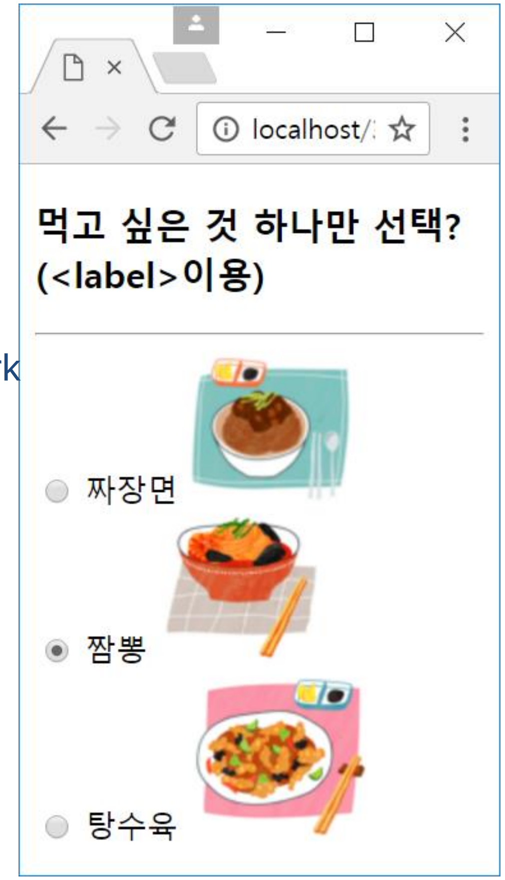
- Text files written in the Hypertext Markup Language(HTML)
- JavaScript code for dynamic behavior and CSS
- Images, videos, and multimedia files are also often embedded

Web Server

```
<!DOCTYPE html>
<html>
<head><title>캡션을 가진 라디오버튼</title></head>
<body>
<h3>먹고 싶은 것 하나만 선택?(&lt;label&gt;이용)</h3>
<hr>
<form>
    <label>
        <input type="radio" name="china" value="1">
        짜장면 <img src="media/jajang.png">
    </label><br>
    <label>
        <input type="radio" name="china" value="2" checked>
        짬뽕 <img src="media/jjambbong.png">
    </label><br>
    <label>
        <input type="radio" name="china" value="3">
        탕수육 <img src="media/tangsuyuk.png">
    </label>
</form>
</body>
</html>
```

Network

Web Client (Browser)



먹고 싶은 것 하나만 선택?(<label>이용)

짜장면

[✓ 짬뽕]

탕수육

Introduction to HTML

HTML File

- **HyperText Markup Language**
- **An HTML file is a *text file* containing small markup tags**
- **The markup tags tell the Web browser how to display the page**
- **An HTML file must have an .htm or .html file extension**
- **An HTML file can be created using a *simple text editor***

Markup Languages

- **Tags indicate what they are and how they should be formatted**
  - Marking-up the document
  - Paired: **<title> My Memories </title>**
  - A pair of tags plus their content constitute an element

- **HTML**
  - HTML places primary emphasis on structure
    - paragraphs, headings, lists, images, links, ….
  - HTML places secondary emphasis on style (CSS)
    - fonts, colors, ….
  - HTML does not label the meaning of the text (XML)
    - → Semantic Web (HTML5)

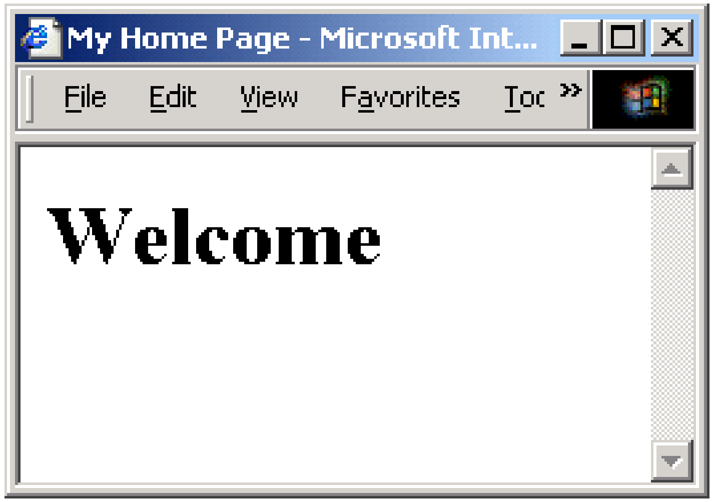
Basic HTML Document

- Every document should start with the following line

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

- There are three required elements, defined by the tags `<html>`, `<head>` and `<body>`

```
<html>
 <head>
  <title>My Home Page</title>
 </head>
 <body>
  <h1>Welcome</h1>
 </body>
</html>
```



Welcome

Basic Structure Elements

- **First and Last tags `<html> </html>`**

- **The HEAD section**
  - must come before the BODY section
  - contains generic information about the document
  - Elements specified in the HEAD section can include
    - **title**, **link**, **script**, **style**

- **The BODY section**
  - contains the content of the document (text, images, etc.)
  - this content is structured by other tags

Block Elements

- **Block elements define sections of text, usually preceded by a blank line**
  - `<p>...</p>` - paragraph
  - `<h1>...</h1>~<h6>...</h6>` - headings
  - `<blockquote>...</blockquote>` - indented text
  - `<div>...</div>` - division
    - used to identify a section of the document

Paragraphs

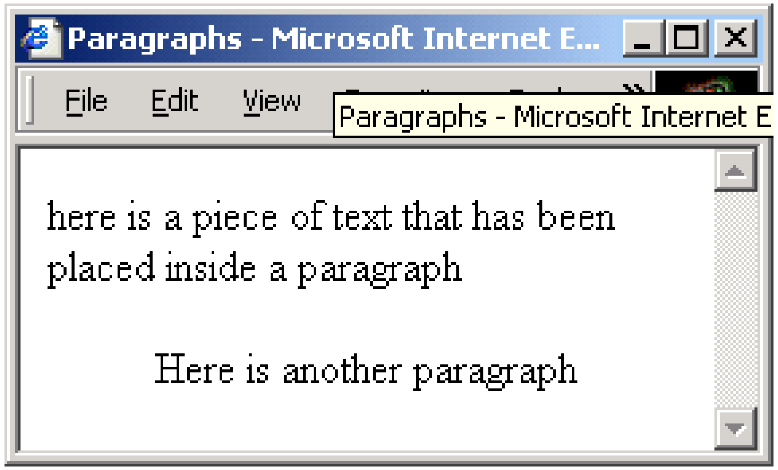
Paragraphs: `<p>...</p>`

- force a break between the enclosed text and the text surrounding it
- the tagged region of text may be subject to special formatting

`<p align="center">Here is another paragraph</p>`

- **`align`** is an attribute of the paragraph tag
- **`center`** is the value of the align attribute

```
<p>here is a piece of
text that has been
placed inside a
paragraph</p>
<p align="center">Here
is another
paragraph</p>
```


Paragraphs - Microsoft Internet E...
File
Edit
View
Paragraphs - Microsoft Internet E

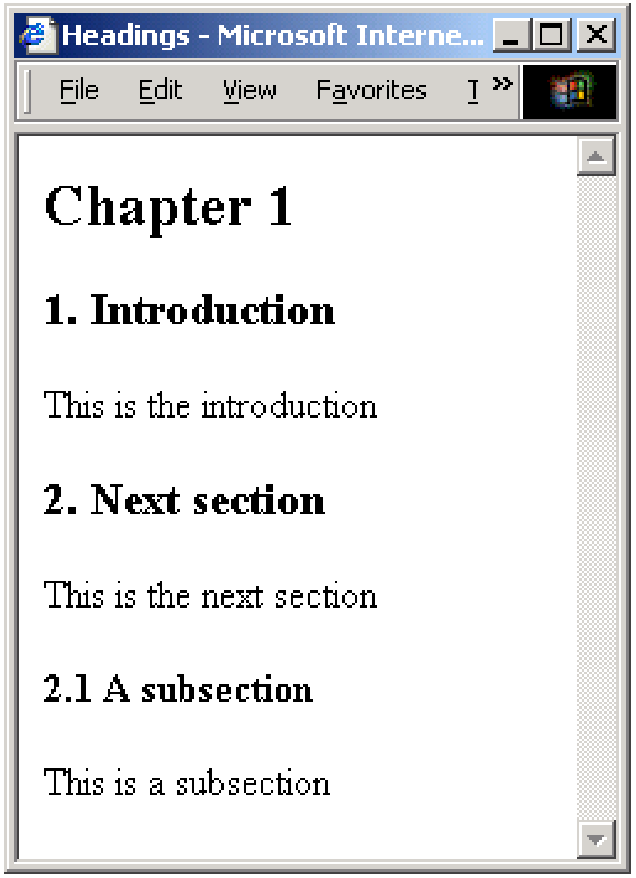
here is a piece of text that has been placed inside a paragraph

Here is another paragraph

Headings

- Six levels of importance `<h1>~<h6>`
- use headings to divide document into sections

```
<html>
 <head>
  <title>Headings</title>
 </head>
 <body>
  <h2>Chapter 1</h2>
  <h3>1. Introduction</h3>
   This is the introduction
  <h3>2. Next section</h3>
   This is the next section
  <h4>2.1 A subsection</h4>
   This is a subsection
 </body>
</html>
```



Chapter 1

1. Introduction

This is the introduction

2. Next section

This is the next section

2.1 A subsection

This is a subsection

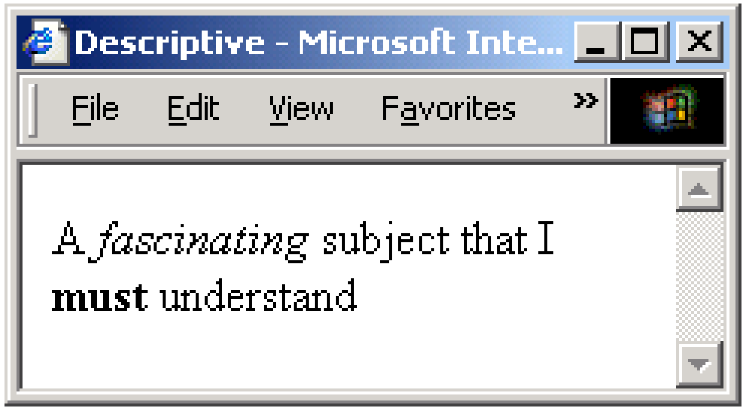
Element Relationships

- **The elements marked by tags form a hierarchy**
- **The root element is html (marked by `<html> </html>`)**
- **It usually has two children: head and body**
  - each of these are further subdivided
- **There are rules for which elements can contain other elements**
  - e.g. headers cannot contain headers
  - see `http://www.w3.org/` for a full list of rules
- **Elements must not overlap each other**
  - we cannot have: `<h1>...<a..> ... </h1>...</a>`
  - we can have: `<h1>...<a..> ... </a>...</h1>`

Inline Descriptive Elements

- **Descriptive elements affect the appearance of text depending on how the text is described**
  - `<em>...</em>` emphasis, usually with italics
  - `<strong>...</strong>` strong, usually with bold
  - `<cite>...</cite>` citation, usually in italics
  - `<code>...</code>` usually results in monotype spacing

```
<body>
A <em>fascinating</em>
subject that I
<strong>must</strong>
understand
</body>
```



A *fascinating* subject that I **must** understand

Inline Explicit Style Elements

- `<boldface>...</boldface>`
- `<big>...</big>` bigger font than surrounding text
- `<small>...</small>` smaller font than surrounding text
- `<i>...</i>` italics
- `<s>...</s>` strikethrough
- `<sub>...</sub>` subscripts
- `<sup>...</sup>` superscripts
- `<span>...</span>` delimits text for stylesheet control
- `<div>...</div>` delimits blocks of text for stylesheet control

Inline Explicit Style Elements

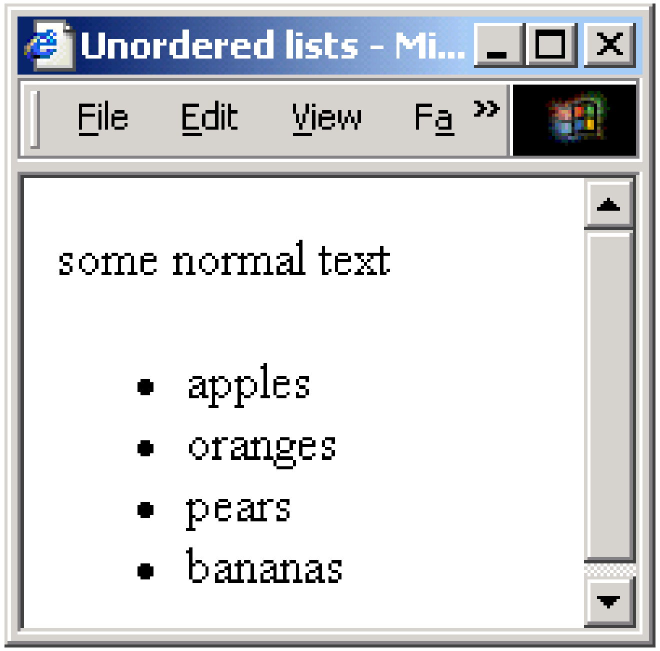
- `<font>` **attributes**
  - `face` - name of font (must be installed)
    - **`"arial", "times", "verdana", "helvetica"`**
  - `size` - absolute size (1-7), or relative to previous text
    - **`"2", "5", "7", "+1", "-2"...`**
  - `color` - hexadecimal RGB, or a named color
    - **`"3399dd", "blue", "red"`**
  - `weight` - boldness from 100, 200, ..., 900
    - **`"100", “200", "900"`**
- **e.g.**

```
<font face="arial" size="+1" color="pink" weight="300">
```

Unordered List

- Unordered lists `<ul>...</ul>`
- `<li>...</li>` for the list elements
- each item has a bullet

```
some normal text
<ul>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
<li>bananas</li>
</ul>
```



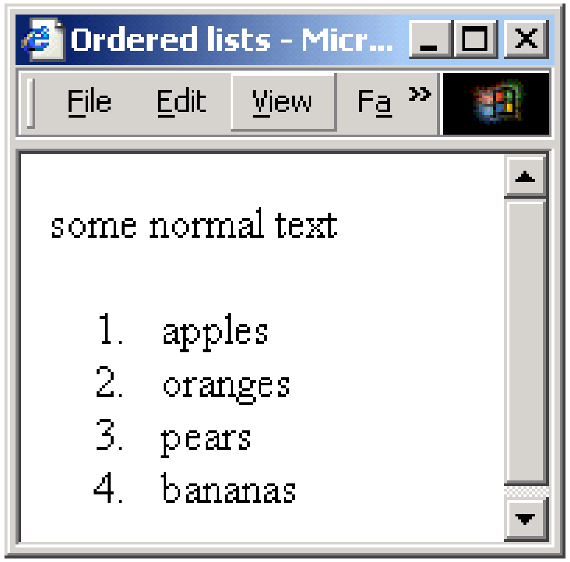
some normal text

- apples
- oranges
- pears
- bananas

Ordered List

- Ordered lists `<ol>...</ol>`
- `<li>...</li>` for the list elements
- each item has a number

```
some normal text
<ol>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
<li>bananas</li>
</ol>
```



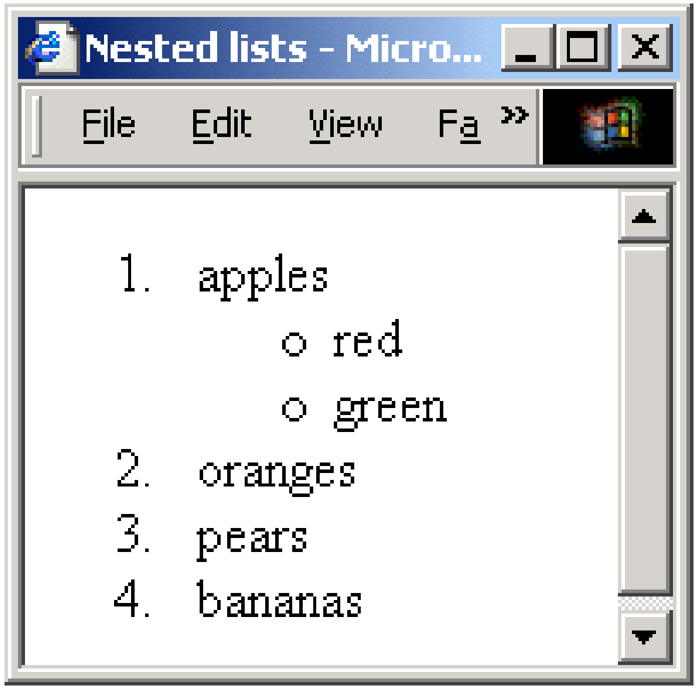
some normal text

1. apples
2. oranges
3. pears
4. bananas

Nested Lists

- A list may contain another list
- The inner list is nested inside the outer list

```
<body>
<ol>
 <li>apples</li>
  <ul>
   <li>red</li>
   <li>green</li>
  </ul>
 <li>oranges</li>
 <li>pears</li>
 <li>bananas</li>
</ol>
</body>
```



1. apples
   - red
   - green
2. oranges
3. pears
4. bananas

Comments

- Comments are delimited by `<!--` and `-->`

```
<!-- this is a comment -->
```

- Comments may span multiple lines

```
<body>
 <!--
  this is
  a comment
 -->
</body>
```

Links

- **The link (anchor) element `<a>...</a>` provides hypertext links between**
  - different documents (using a URL)
  - different parts of an individual document

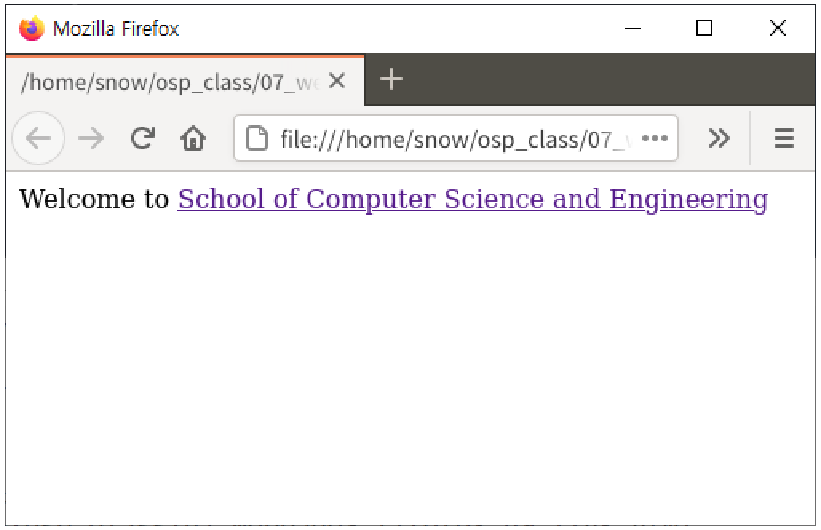
- **User selection of the link (hot spot) results in**
  - retrieval and display of the designated document
  - movement to relevant part of same document

```
<body>
The Department of
<a href="http://www.doc.gold.ac.uk/index.html">
Computer Science</a> is a very ....
</body>
```

Link with URL

- The **`href`** attribute gives the URL of the target page
- The text between the tags is highlighted – selecting it activates the link

```
<body>
Welcome to
<a href="http://cse.knu.ac.kr/main/index.html">
School of Computer Science and Engineering </a>
</body>
```



Welcome to School of Computer Science and Engineering

Relative Addressing

- The previous example gave the full path name, known as the absolute address

```
<a href="research.html">Research</a>
<a href="./pub.html">Publications</a>
<a href="../../index.html">Computer Science home</a>
```

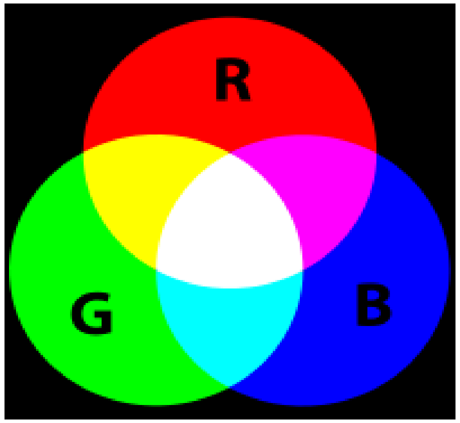
- The ‘root’ directory for the link is assumed to be the directory containing the parent page of the link

Colour

- Colours are specified with hexadecimal numbers for the red, green and blue primary colours, preceded by a "#".
- To set the background colour of a web page

```
<body bgcolor="#994422">
```

- RGB Model
  - #ff0000 (red)
  - #00ff00 (green)
  - #0000ff (blue)
  - #ffff00 (yellow)
  - #3395ab (a pastel blue)



Forms

- **Server-based programs may return data to the client as a web page**

- **Client-side scripts can read input data**
  - To validate the data, prior to sending to server
  - To use in local processing which may output web page content that is displayed on the client

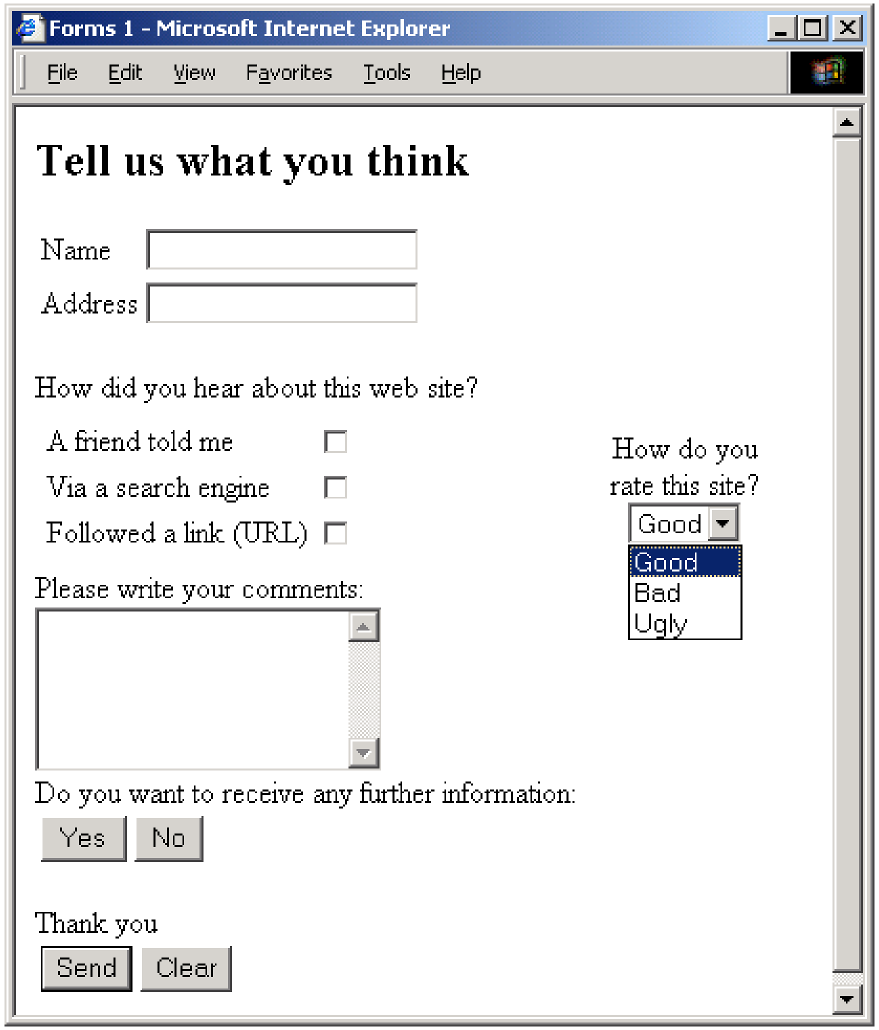
Web Applications

- **E.g.**
  - **Questionnaires** to provide feedback on a web site
  - **e-commerce**, to enter name, address, details of purchase and credit-card number
  - request **brochures** from a company
  - make a **booking** for holiday, cinema etc.
  - **buy** a book, CD, etc.
  - obtain a **map** giving directions to a shop

- **Run a database query and receive results (an important part of e-commerce)**

Input Types

- `text`
- `checkbox`
- `radio` (buttons)
- `select` (options)
- `textarea`
- `password`
- `button`
- `submit`
- `reset`
- `hidden`
- `file`
- `image`


Forms 1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Tell us what you think

Name ☐

Address ☐

How did you hear about this web site?

A friend told me ☐
Via a search engine ☐
Followed a link (URL) ☐

How do you rate this site?
Good
Good
Bad
Ugly

Please write your comments:

Do you want to receive any further information:
Yes No

Thank you
Send Clear

The `method` and `action` attributes

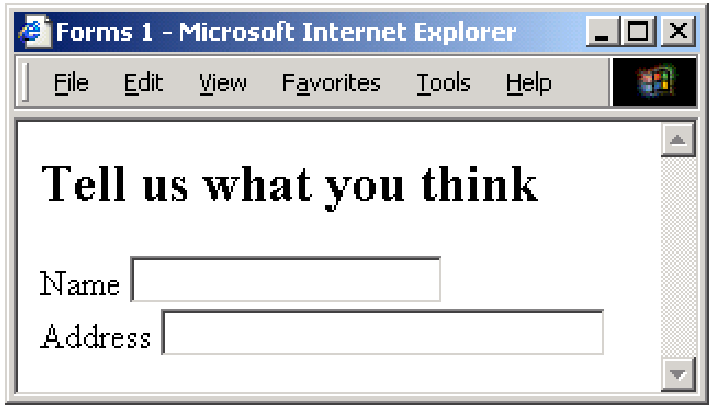
- **The `method` attribute specifies the way that form data is sent to the server program**
  - `GET` appends the data to the URL
  - `POST` sends the data separately
- **The `action` attribute specifies a server program that processes the form data (often as a URL)**

```
<body>
 <form method="POST" action="comments.php">
  <h2>Tell us what you think</h2>
  <!- etc. -->
 </form>
</body>
```

The input element: type="text"

- The type attribute specifies the type of user input
- The name attribute gives an identifier to the input data

```
<form method="POST" action="comments.php">
  <h2>Tell us what you think</h2>
  Name <input name="name" type="text" size="20"><br>
  Address <input name="address" type="text" size="30">
</form>
```


Forms 1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Tell us what you think

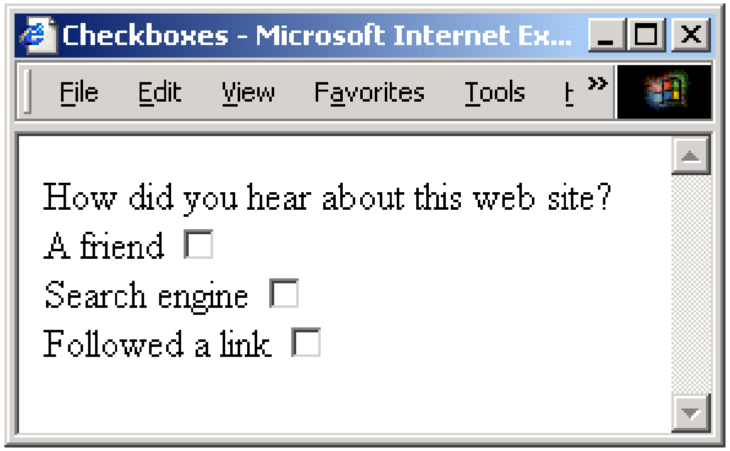
Name

Address

The input element: type=“checkbox”

- The name attribute is used to define a set of checkboxes
- The value attribute identifies the individual checkbox
- If the checked attribute is set the box is initially checked

```
How did you hear about this web site?<br>
A friend
<input type="checkbox" name=“web" value="friend"><br>
Search engine
<input type="checkbox" name=“web" value="engine"><br>
...
```

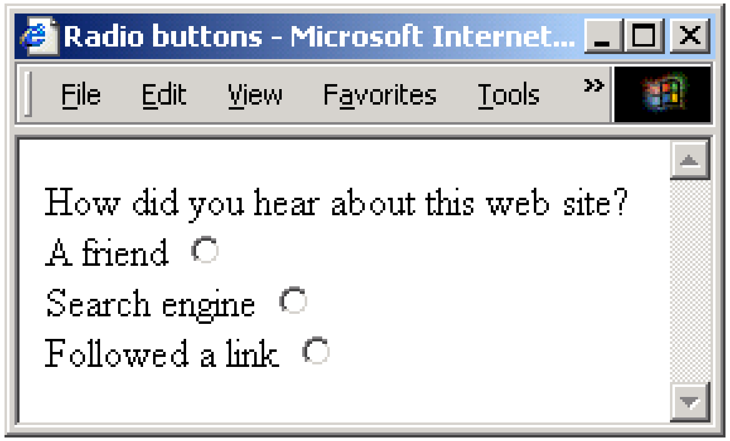

Checkboxes - Microsoft Internet Ex...
File Edit View Favorites Tools H

How did you hear about this web site?
A friend ☐
Search engine ☐
Followed a link ☐

The input element: type="radio"

- Radio buttons are similar to checkboxes, but only one can be selected
- To select a button by default, use the `checked` attribute (for one button only)

```
How did you hear about this web site?<br>
A friend
<input type="radio" name="web" value="friend"><br>
Search engine
<input type="radio" name="web" value="engine"><br>
...
```

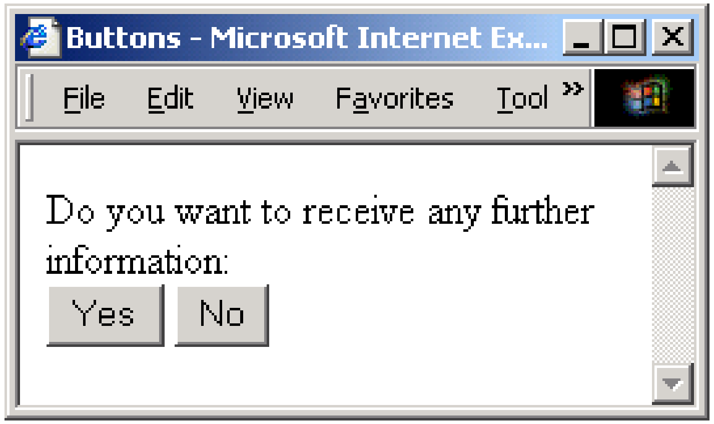


How did you hear about this web site?
A friend ○
Search engine ○
Followed a link ○

The input element: type="button"

- The name attribute uniquely identifies a button
- The value attribute gives a label to the button
- Actions can be associated with buttons using JavaScript

```
Do you want to receive any further information:<br>
<input type="button" name="yes" value=" Yes ">
<input type="button" name="no" value=" No "><br>
```

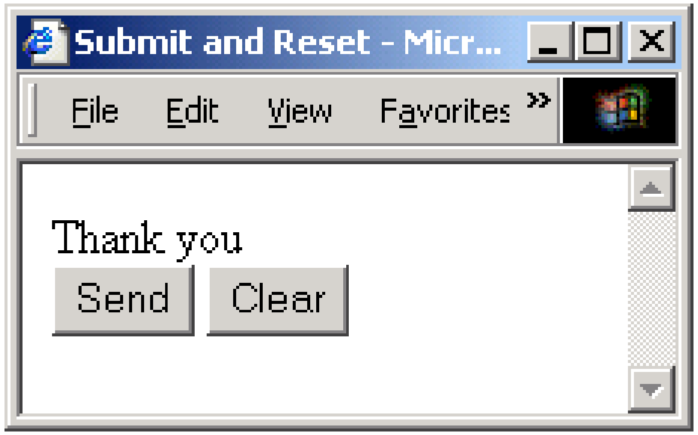

Buttons - Microsoft Internet Ex...
File Edit View Favorites Tool
Yes
No

Do you want to receive any further information:

The input element: `type="submit/reset"`

- `type="submit"`
  - clicking this button sends the form data to the program (URL) specified in the **`action`** attribute of the form
- `type="reset"`
  - clicking this button clears all data entered so far

```
Thank you<br>
<input type="submit" name="send" value="Send">
<input type="reset" name="clear" value="Clear"><br>
```

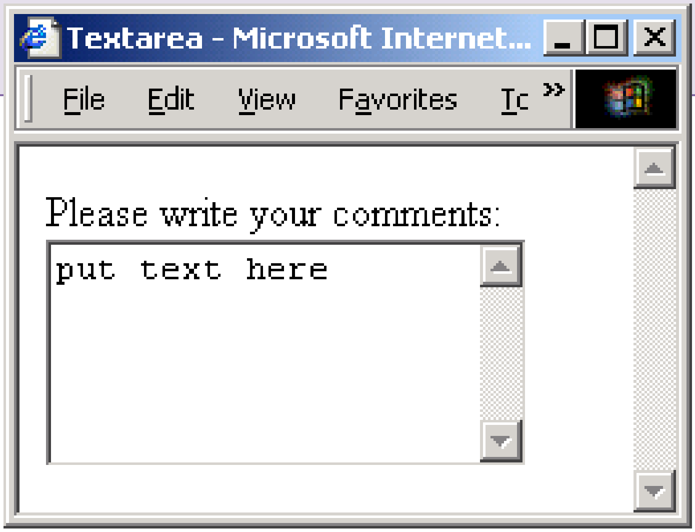


Thank you

The textarea element

- Used for multi-line text input
- The size of the input area is specified with the cols and rows attributes
- Any text placed inside the element appears in the input area (this can be deleted).

```
Please write your comments:<br>
<textarea name="comments" rows="5" cols="20">
  put text here
</textarea>
```



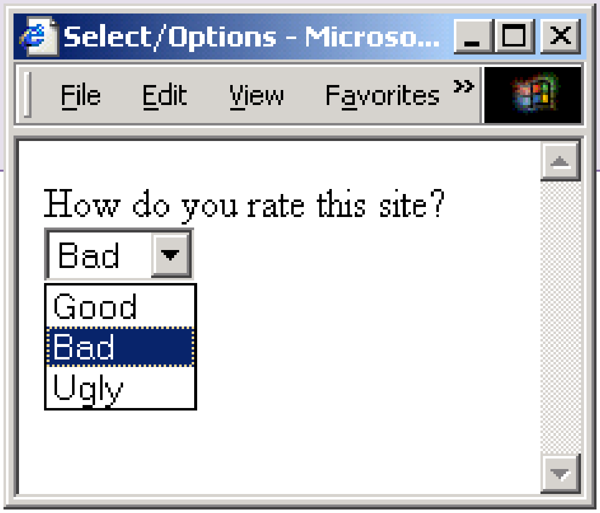
Please write your comments:

```
put text here
```

The select element

- The select element provides a menu of options
- An option can be selected by default using the selected attribute (otherwise the first in the list is initially selected)

```
How do you rate this site?<br>
<select name="rating">
  <option>Good
  <option selected>Bad
  <option>Ugly
</select>
```



How do you rate this site?

Bad

Good

Bad

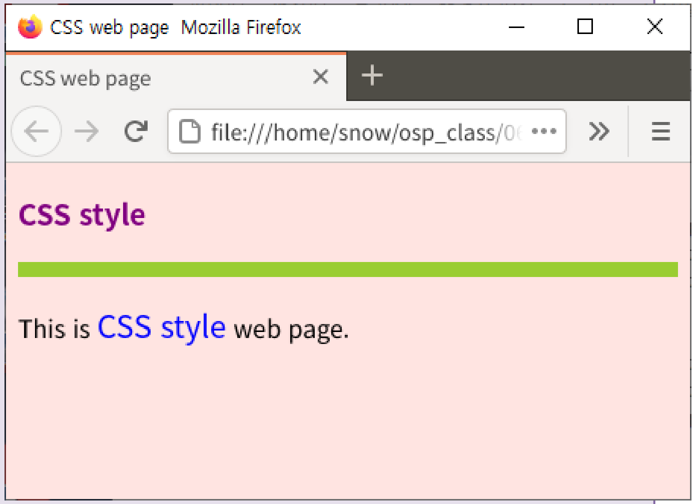
Ugly

CSS (Cascading Style Sheet)

- A style sheet language
- Designed to enable the separation of presentation and content, including layout, colors, and fonts

```
<!DOCTYPE html>
<html>
<head><title>CSS web page</title>
<style>
  body { background-color : mistyrose; }
  h3 { color : purple; }
  hr { border : 5px solid yellowgreen; }
  span { color : blue; font-size : 20px; }
</style>
</head>

<body>
<h3>CSS style</h3>
<hr>
<p>This is <span>CSS style</span> web page.</p>
</body>
</html>
```



CSS style

This is CSS style web page.

Introduction to XML

What is XML?

- **eXtensible Markup Language**

- **Tags are added to the document to provide the extra information**

- **HTML tags tell a browser how to display the document**
- **XML tags give a reader some idea what some of the data means**

Advantages of XML

- **XML is text (Unicode) based**
  - takes up less space
  - can be transmitted efficiently

- **One XML document can be displayed differently in different media**
  - HTML, Video, CD, DVD,
  - You only have to change the XML document in order to change all the rest

- **XML documents can be modularized. Parts can be reused.**

Example of an HTML & XML Documents

```
<html>
 <head><title>Example</title></head.
 <body>
  <h1>This is an example of a page.</h1>
  <h2>Some information goes here.</h2>
 </body>
</html>
```

```
<?xml version=“1.0”/>
<address>
 <name>Alice Lee</name>
 <email>alee@aol.com</email>
 <phone>212-346-1234</phone>
 <birthday>1985-03-22</birthday>
</address>
```

HTML vs. XML

- HTML tags have a fixed meaning and browsers know what it is
- XML tags are different for different applications, and users know *what they mean*

- HTML tags are used for display
- XML tags are used to describe documents and data

XML Rules

- **Tags come in pairs with *start-tags* and *end-tags***
- **Tags must be properly nested**
  - **<name> <email>…</name> </email> is not allowed**
  - **<name> <email>…</email> <name> is OK**

- **Tags are *case sensitive*.**
  - <address> is not the same as </Address>

- **Tags may not contain ‘<‘ or ‘&’.**

- **etc.**

XML Example

- Markup for the data aids understanding of its purpose
- A flat text file is not nearly so clear

```
<?xml version="1.0"/>
<address>
 <name>Alice Lee</name>
 <email>alee@aol.com</email>
 <phone>212-346-1234</phone>
 <birthday>1985-03-22</birthday>
</address>
```

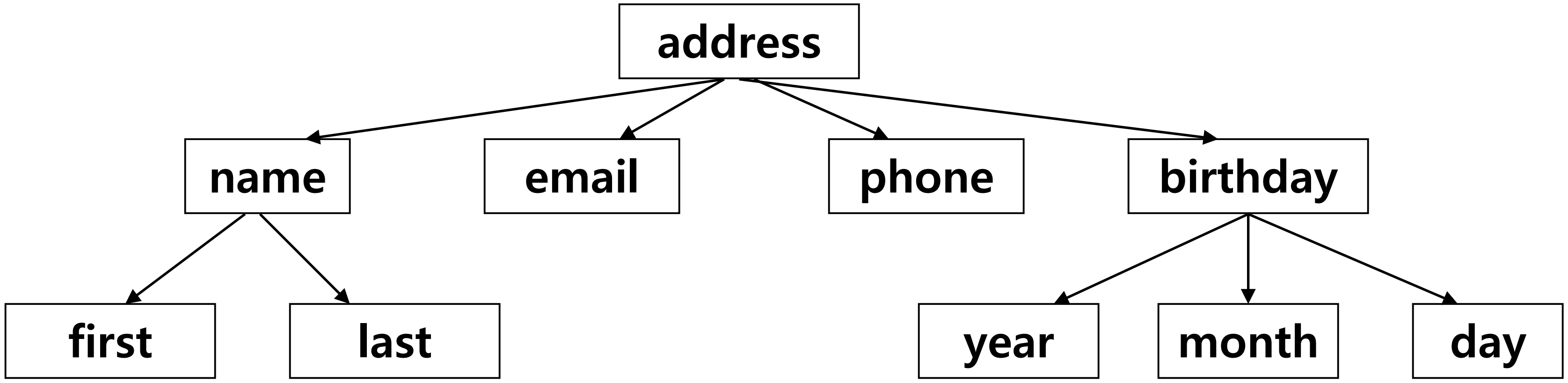
- The last line looks like a date, but what is it for?

XML Advanced Example

```
<?xml version = “1.0” ?>
<address>
   <name>
     <first>Alice</first>
     <last>Lee</last>
   </name>
   <email>alee@aol.com</email>
   <phone>123-45-6789</phone>
   <birthday>
     <year>1983</year>
     <month>07</month>
     <day>15</day>
   </birthday>
</address>
```

XML Files are Trees

- **An XML document has a single root node**
- **The tree is a general ordered tree.**
  - A parent node may have any number of children.
  - Child nodes are ordered, and may have siblings.
- **Preorder traversals are usually used for getting information out of the tree.**



Introduction to JSON (JavaScript Object Notation)

JavaScript (JS)

- Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web
- Interactive web pages
- Event-driven, functional, and imperative programming styles

- JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js
- Although there are similarities between JavaScript and Java, including language name, syntax, etc., the two languages are distinct and differ greatly in design

JavaScript Example

```
<!DOCTYPE html>
<html>
 <head><title>JavaScript</title>
  <script>
  function over(obj) {
    obj.src="media/banana.png";
  }
  function out(obj) {
    obj.src="media/apple.png";
  }
  </script>
 </head>
 <body>
  <img src="media/apple.png" alt=“apple_image"
            onmouseover="over(this)"
            onmouseout="out(this)">
 </body>
</html>
```

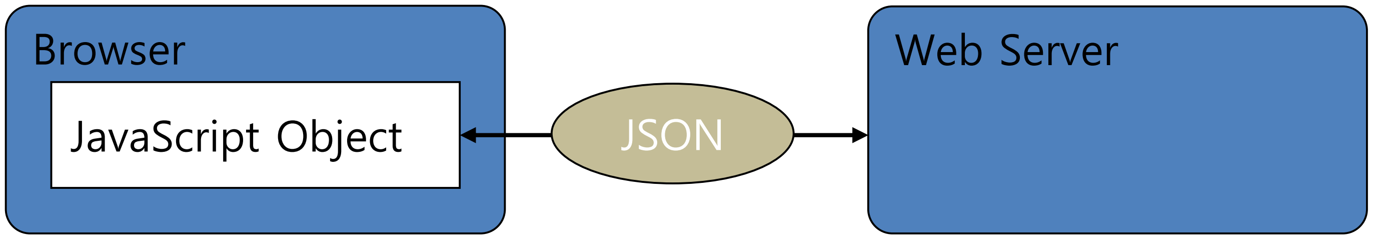
JavaScript Example

```
/* JavaScript “lib.js” file */
  function over(obj) {
    obj.src="media/banana.png";
  }
  function out(obj) {
    obj.src="media/apple.png";
  }
```

```
<!DOCTYPE html>
<html>
 <head><title>JavaScript</title>
  <script src=“lib.js”></script>
 </head>
 <body>
  <img src="media/apple.png" alt=“apple_image"
            onmouseover="over(this)"
            onmouseout="out(this)">
 </body>
</html>
```

JSON (JavaScript Object Notation)

- **A Format (or Syntax) to store and exchange object data**
- **Text data**



From JavaScript to JSON

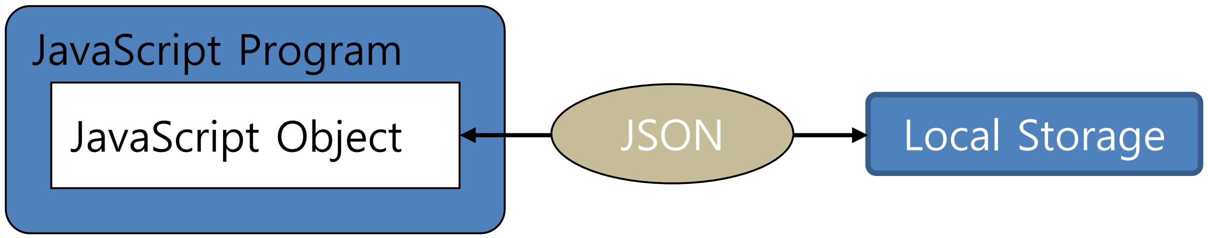
```
var myObj = { "name":"John", "age":31, "city":"New York" };
var myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;
```

From JSON to JavaScript

```
var myJSON = '{ "name":"John", "age":31, "city":"New York" }';
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
```

JSON (JavaScript Object Notation)

- **Storing data in local storage as text data**



Storing data:

```
myObj = { "name":"John", "age":31, "city":"New York" };
myJSON = JSON.stringify(myObj);
localStorage.setItem("testJSON", myJSON);
```

Reading data

```
text = localStorage.getItem("testJSON");
obj = JSON.parse(text);
document.getElementById("demo").innerHTML = obj.name;
```

JSON – Data Types and Values

- **JSON values must be one of the following types**
  - a string (should be always doubl-quoted): {"name":"John" }
  - a number: {“age":50 }
  - an object (JSON object):
    { "employee":{"name":"John", "age":30, "city":"New York" } }
  - an array: {"employees":["John", "Anna", "Peter"] }
  - a boolean: {“married”:true}
  - null: { “middlename”: null}

- **JSON file: .json**

JSON – Object

- JSON objects are surrounded by curly braces { }.
- JSON objects are written in key/value pairs.
- Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null).
- Keys and values are separated by a colon.
- Each key/value pair is separated by a comma.

- Accessing JSON Object within JavaScript

```
myObj = { "name":"John", "age":30, "car":null };
x = myObj.name;
x = myObj["name"];
```

```
myObj = { "name":"John", "age":30, "car":null };
for (x in myObj) {
    document.getElementById("demo").innerHTML += myObj[x] + "<br>";
}
```

Nested JSON Objects

- **Values in a JSON object can be another JSON object.**

```
myObj = {
    "name":"John",
    "age":30,
    "cars": {
        "car1":"Ford",
        "car2":"BMW",
        "car3":"Fiat"
    }
 }
```

```
x = myObj.cars.car2;
//or:
x = myObj.cars["car2"];
```

```
myObj.cars["car2"] = "Mercedes";
```

```
delete myObj.cars.car2;
```

JSON Array Objects

- **Values in a JSON object can be another JSON object.**

```
<!DOCTYPE html>
<html> <body>
<p>Loopin through an array using a for loop:</p>
<p id="demo"></p>
<script>
 var myObj, i, x = "";
 myObj = {
    "name":"John",
    "age":30,
    "cars":[ "Ford", "BMW", "Fiat" ]
 };
 for (i in myObj.cars) {
    x += myObj.cars[i] + "<br>";
 }
 document.getElementById("demo").innerHTML = x;
</script>
</body> </html>
```

JSON vs. XML

- **Both JSON and XML can be used to receive data from a web server.**

```
{"employees":[
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
]}
```

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

Any Questions...
Just Ask!