INTRODUCTION TO JAVASCRIPT
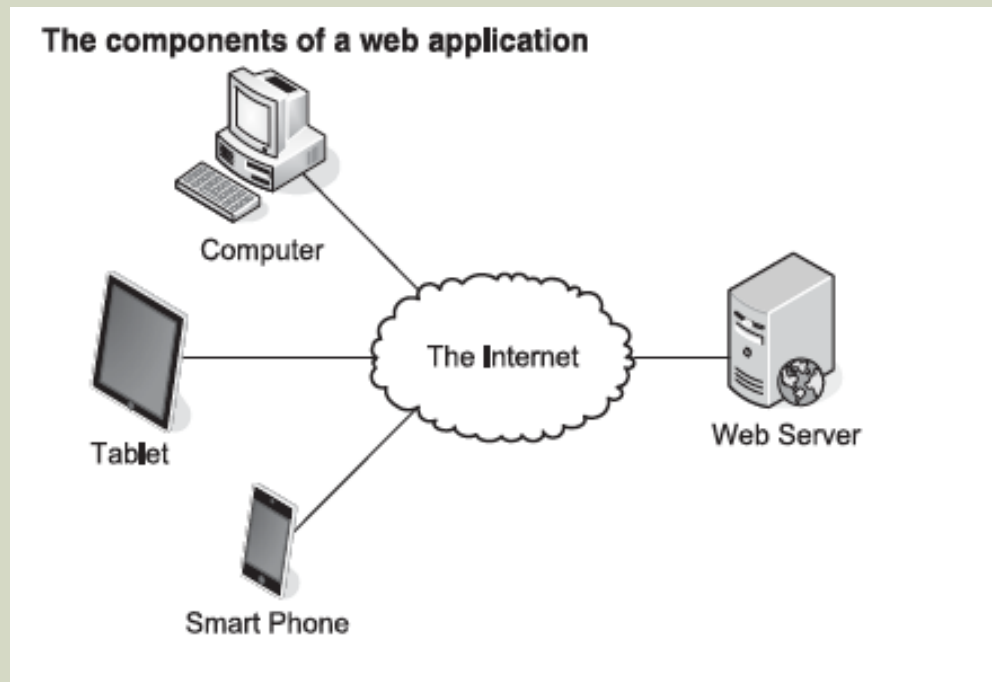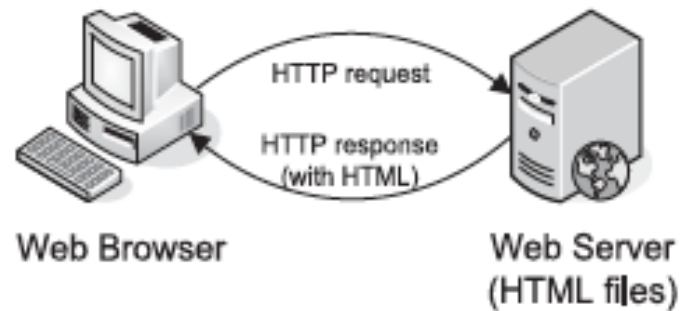
Unit 1

INDEX

1 - HOW A WEB APPLICATION WORKS

- Clients user programs (web browsers) to request web pages from the web server.
- Web Server returns the pages that are requested to the browser.

The components of a web application

2 - HOW STATIC WEB PAGES ARE PROCESSED

- A static web page is a web page that doesn't change each time it is requested (html or htm).
- Example: http://roble.pntic.mec.es/apuente/html/paginas/uno.htm

How a web server processes a static web page

HTTP request

HTTP response
(with HTML)

Web Browser

Web Server
(HTML files)

3 - HOW DYNAMIC WEB PAGES ARE PROCESSED

- A dynamic web page is a page that's created by a program or script on the web server each time it is requested.

- Important!!!! Web browser doesn't know if the html is created dynamically or not. Web browser just display the html response from the server

- Example: https://www.amazon.es/
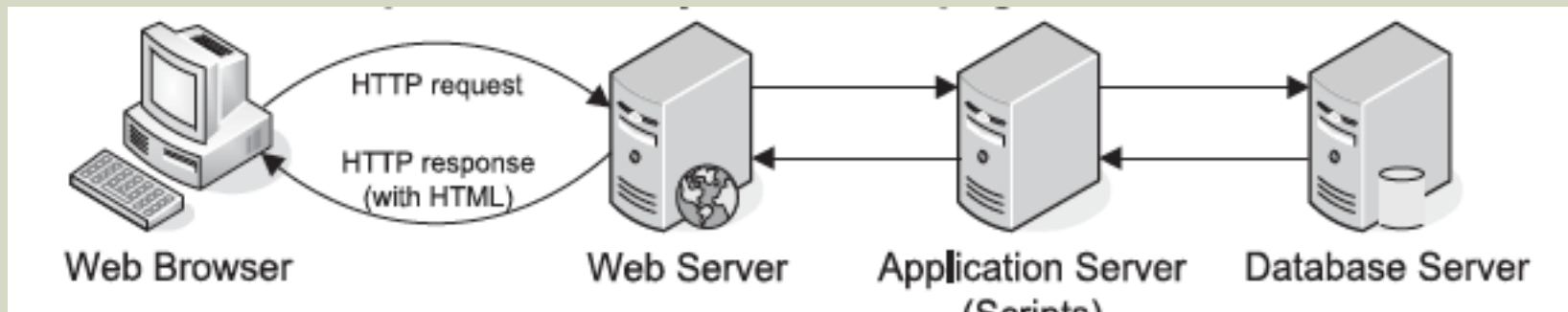
4 - HOW JAVASCRIPT AND JQUERY ARE USED FOR CLIENT-SIDE PROCESSING

- Javascript is a scripting language that provides client-side processing.
- HERE, javascript is used to change the images that are shown without using server-side processing.
- To make this work, all the images are loaded into the browser when the page is requested.
- The javascript code is executed in the web browser by the browser's JavaScript engine.
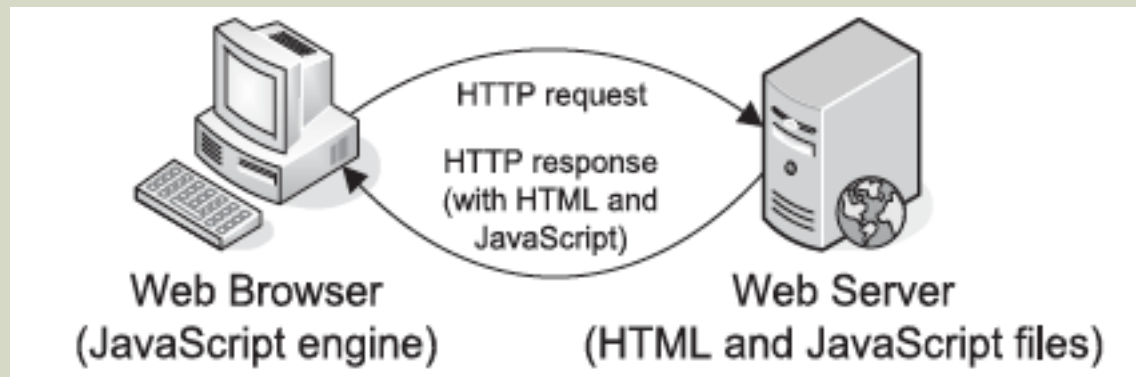
4 - HOW JAVASCRIPT AND JQUERY ARE USED FOR CLIENT-SIDE PROCESSING

- There are many other uses for Javascript:
  - Validate form before it is sent to the server for processing (save unnecessary trips to the server).
  - Slide shows (carousels)
  - Image swaps and rollovers.

- jQuery is a JavaScript library that makes easier to do many of the functions that JavaScript can be used for.

5 - HOW TO INCLUDE JAVASCRIPT / JQUERY IN AN HTML DOCUMENT

- **BEST OPTION**
  - Include it as a external reference in the head section.

```
1   <!DOCTYPE html>
2   <html lang="es">
3       <head>
4           <meta charset="utf-8" />
5           <title>Including javascript</title>
6           <script src="JavaScriptCode.js"></script>
7           <!--  <script src="./javascript/yourJavaScriptCode.js"></script>!-->
8       </head>
9       <body>
10          HELLO WORLD!!!
11      </body>
12  </html>
```

```
1   <link rel="stylesheet" href="jquery-ui.min.css">
2   <script src="external/jquery/jquery.js"></script>
3   <script src="jquery-ui.min.js"></script>
```

5 - HOW TO INCLUDE JAVASCRIPT / JQUERY IN AN HTML DOCUMENT

- **Other option (not recommended)**
 - Include all the JS code in the head.

```
1   <!DOCTYPE html>
2   <html lang="es">
3       <head>
4           <meta charset="utf-8" />
5           <title>Including javascript</title>
6           <script>
7               alert("This is hell. Not use it or your teacher will kill you. ;-)");
8           </script>
9
10      </head>
11      <body>
12          HELLO WORLD!!!
13      </body>
14  </html>
```

6 – JAVASCRIPT SINTAX

- Very similar to Java. Creating identifiers rules, comments, etc.
- JavaScript is case-sensitive. Simpsons != simpsons
- JavaScript ignores extra whitespace in statements.
- JavaScript statements must end with a semicolon (;)

MISS ONE SEMICOLON
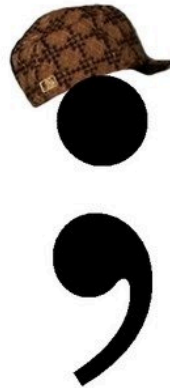
300 LINES OF SYNTAX ERRORS

7 – DATA TYPES

- **Numbers**
  - Can represent both integer and decimal values (also with sign)
  - var subtotal = 74;   //subtotal = 74
  - subtotal += 20;   //subtotal = 94

- **String**
  - Concatenation and some functions work as Java.
  - var firstName = "Ray";
  - var fullName = firstName + "Harris";

- **Boolean**
  - var isValid = false;

- **Array**
  - var cars = ["ford","seat","audi"];

- **Undefined**
  - var miZip;

8 – HOW TO CODE CONDITIONAL EXPRESSIONS

- isNan(string) -> determines wether a string value is a valid numeric value
- Relational and logical operators are like java. (==,>=,&&, etc)
- If statement is like java too.

```javascript
if(name == 'ramonet'){
    alert("It's Ramonet");
}else if(name == 'pepet' || name == 'marieta'){
    alert("It's pepet or marieta");
}else {
    alert("It's dead");
}
```

- while, do-while and for loops are like java.
- do-while

```
var sumOfNumbers = 0;
var numberOfLoops = 5;
var counter = 1;
do {
    sumOfNumbers += counter;      // adds counter to sumOfNumbers
    counter++;                     // adds 1 to counter
}
while (counter <= numberOfLoops);
alert(sumOfNumbers);               // displays 15
```

- while

```
var total = 0, count = 0, number;
number = parseFloat( prompt("Enter a number:") );
while ( !isNaN(number) ) {
    total += number;
    count++;
    number = parseFloat(
        prompt("Enter another number or click Cancel to stop:") );
}
var average = total / count;
alert("The average is: " + average);
```

- for

```
var sumOfNumbers = 0;
var numberOfLoops = 5;
for ( counter=1; counter <= numberOfLoops; counter++ ) {
    sumOfNumbers += counter;      // adds counter to sumOfNumbers
}
alert(sumOfNumbers);              // displays 15
```