C# .NET PROGRAMMING: A GRAPHICAL APPROACH CLASS 9

ING. HAZAEL FERNANDO MOJICA GARCÍA

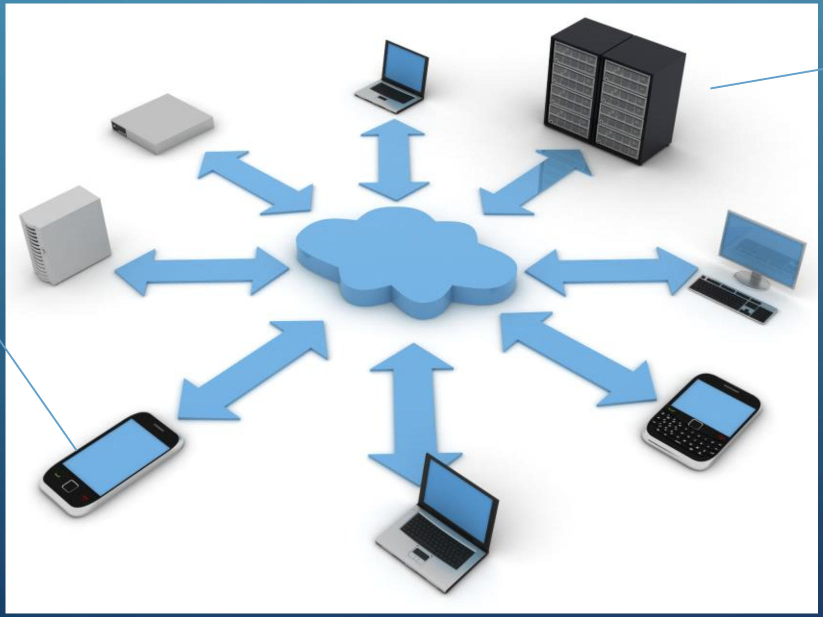
- **WEB PROGRAMMING. AN INTRODUCTION**
- HTML5, CSS
- JAVASCRIPT AND JQUERY
- BOOTSTRAP FOR RESPONSIVE WEB SITES

PROGRAMACIÓN WEB

Se le llama desarrollo web (Web Development) a aquel tipo de programación que será ejecutado en el navegador de una computadora cliente.
La única diferencia entre la programación de aplicaciones de escritorio y la programación web es que en Web uno debe siempre pensar en el aspecto cliente – servidor.

En la programación web uno se preocupa normalmente en cómo se llevará acabo la comunicación entre la computadora cliente (la cual está corriendo HTML / CSS y scripts) y el servidor el cual está corriendo el procesos de Web Server (escuchando el puerto web: 80, 8080…) por medio de algún lenguaje de programación: PHP, Java, .Net, Ruby.

Los **navegadores web** de los **clientes** reciben texto plano (HTML, Js) como respuesta del servidor y lo procesan de tal forma que llegue a significar algo. Ejemplo: Una página WEB.



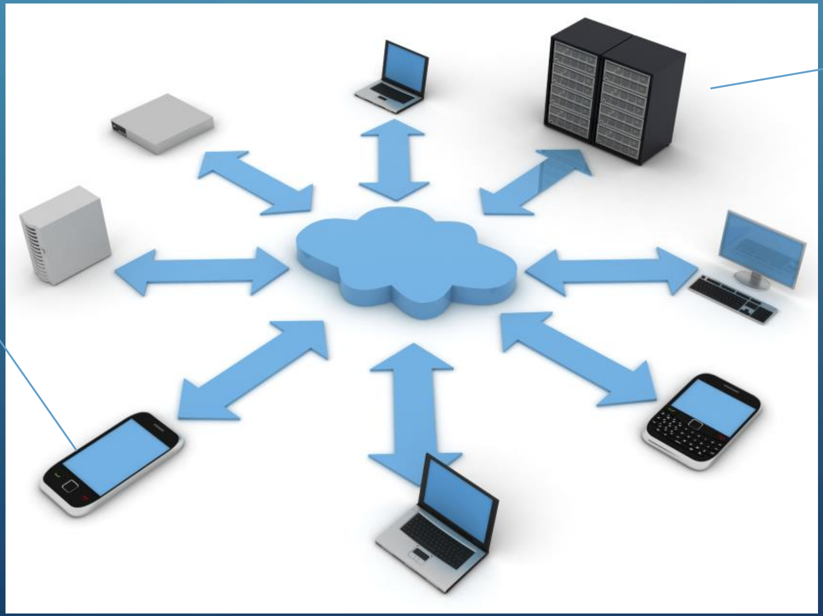
El **servidor** procesa las peticiones del cliente y regresa una respuesta en forma de texto plano: HTML, CSS, Scripts (JavaScript, Jquery)

WEB PROGRAMMING

It is called **Web Development** to that type of programming that is executed in the browser of a client computer. The only difference between the desktop application programming and Web programming Web is that one should always think about the client - server correlation.

Web programming one normally concerned about how it will conduct communication between the client computer (which is running HTML / CSS and scripts) and the server which is running the Web Server processes (to do so it listens to ports: 80, 8080 ...) programmed using a programming language: PHP, Java, .Net, Ruby.

Web browsers in the client computer receive plain text (like HTML, Javascript) as an answer from the server, the browser computes it to build something cool, like a web page.



The server processes client requests and returns a response in the form of plain text: HTML, CSS, scripts (JavaScript, Jquery)

HTML

HTML (HyperText Markup Language) is a standard markup languages (like TEX, SCRIBE, XML ...) and is used to create web pages.
Currently HTML is supported by all web browsers and usability can be extended through styles as CSS languages and scripts like JavaScript.

HTML was created 1990 at **CERN** (European Organization for Nuclear Research - Conseil Européen pour la Recherche Nucléaire), Switzerland, by physicist Tim Berners-Lee and has since undergone many modifications, versions and standards up to the robust version, dynamic multimedia support and we know today.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

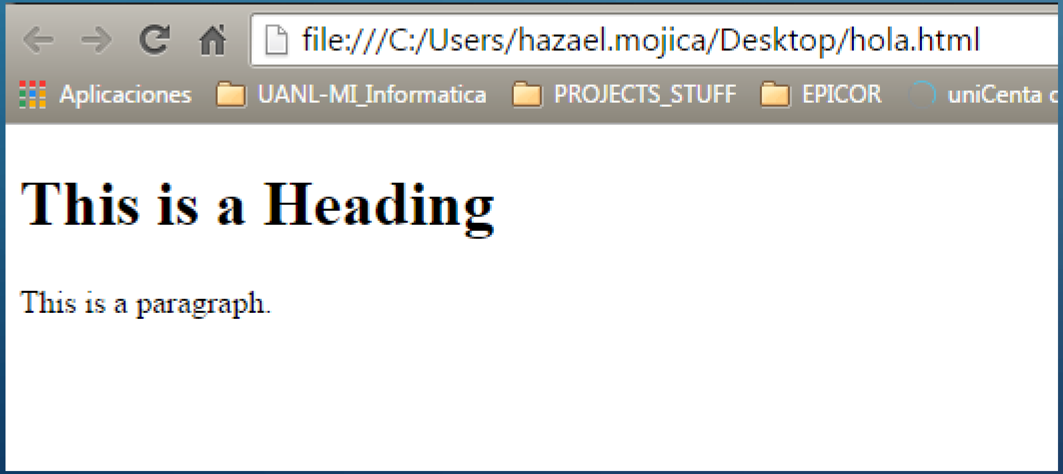
http://www.w3schools.com/html/

Create a text file on your desktop called hola.html, open it with Visual Studio or Notepad (HTML is plain text and therefore can be edited with any text program) and add the following code, save. Open the same file with the browser.

```html
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
</head>
<body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

Header

Body



This is a Heading

This is a paragraph.

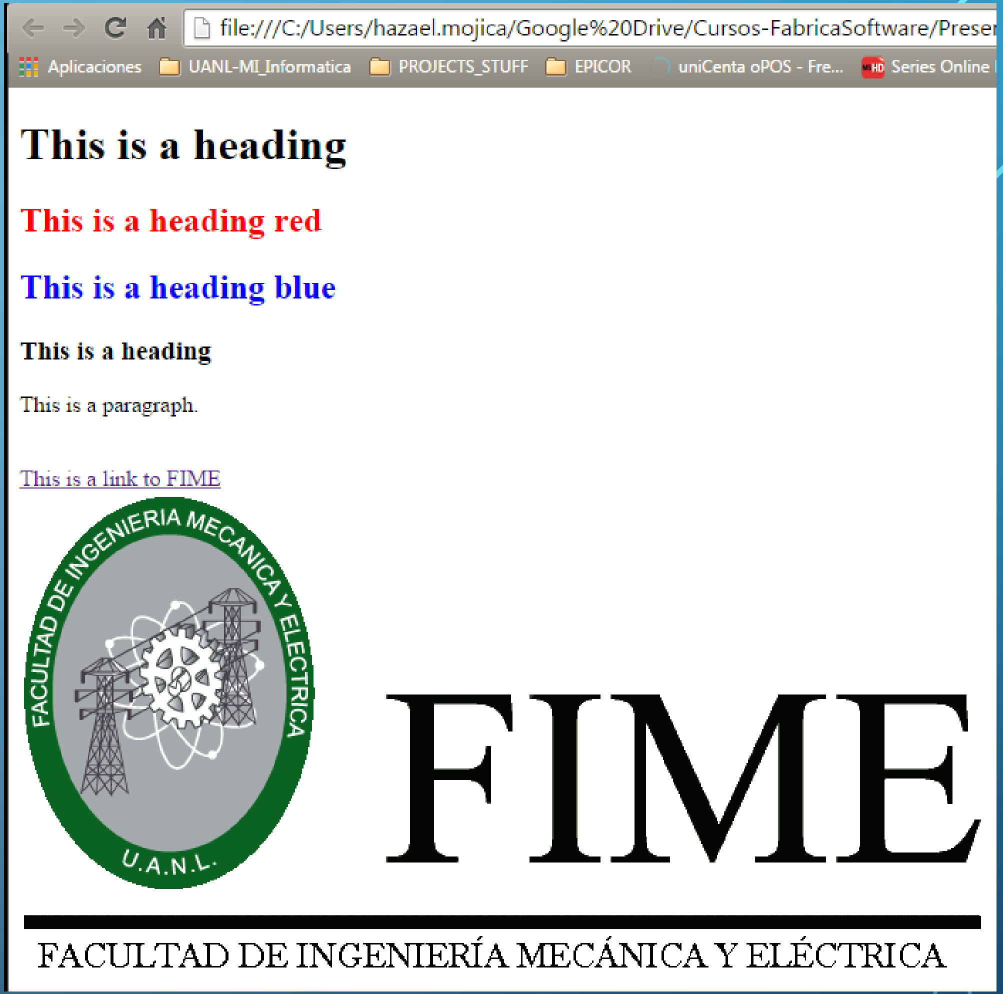
Example Explained

- The **DOCTYPE** declaration defines the document type to be HTML
- The text between **<html>** and **</html>** describes an HTML document
- The text between **<head>** and **</head>** provides information about the document
- The text between **<title>** and **</title>** provides a title for the document
- The text between **<body>** and **</body>** describes the visible page content
- The text between **<h1>** and **</h1>** describes a heading
- The text between **<p>** and **</p>** describes a paragraph

Using this description, a web browser can display a document with a heading and a paragraph.

http://www.w3schools.com/html/html_intro.asp

```html
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
</head>
<body>
    <h1>This is a heading</h1>
    <h2 style="color:red">This is a heading red</h2>
    <h2 style="color:blue">This is a heading blue</h2>
    <h3>This is a heading</h3>
    <p>This is a paragraph.</p>
    <br />
    <a href="http://www.fime.uanl.mx/">This is a link
to FIME</a>
    <br />
    <img src="logoFime.gif" alt="FIME" width="680"
height="340">
</body>
</html>
```


file:///C:/Users/hazael.mojica/Google%20Drive/Cursos-FabricaSoftware/Preser

This is a heading

This is a heading red

This is a heading blue

This is a heading

This is a paragraph.

This is a link to FIME

FACULTAD DE INGENIERIA MECANICA Y ELECTRICA
U.A.N.L.
FIME
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Tags h1, h2, h3 are used to create titles and demarcate parts of our site.
The p tag indicates a paragraph.
The a tag creates a link.
The img tag creates an image.

9.1-Hola

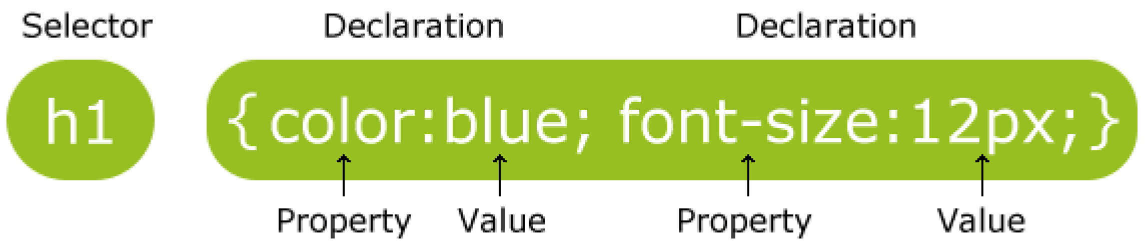
CSS

What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS defines **how HTML elements are to be displayed**
- Styles were added to HTML 4.0 **to solve a problem (HTML was NEVER intended to contain tags for formatting a document.)**
- CSS saves a lot of work
- External Style Sheets are stored in **CSS files**

CSS Syntax

A CSS rule set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

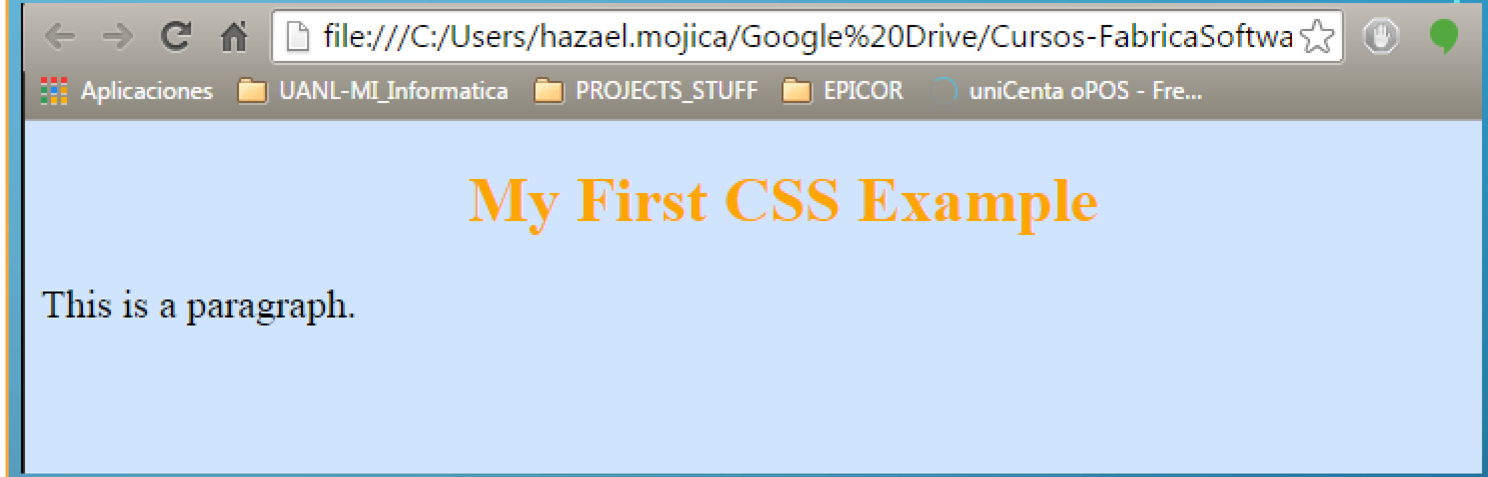
The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a property name and a value, separated by a colon.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body {
            background-color: #d0e4fe;
        }

        h1 {
            color: orange;
            text-align: center;
        }

        p {
            font-family: "Times New Roman";
            font-size: 20px;
        }
    </style>
</head>
<body>
    <h1>My First CSS Example</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```


file:///C:/Users/hazael.mojica/Google%20Drive/Cursos-FabricaSoftwa

Aplicaciones UANL-MI_Informatica PROJECTS_STUFF EPICOR uniCenta oPOS - Fre...

My First CSS Example

This is a paragraph.

To create a file of styles, the **link** tag is used.
And the file will have a **.css** extension

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

9.2-CSS

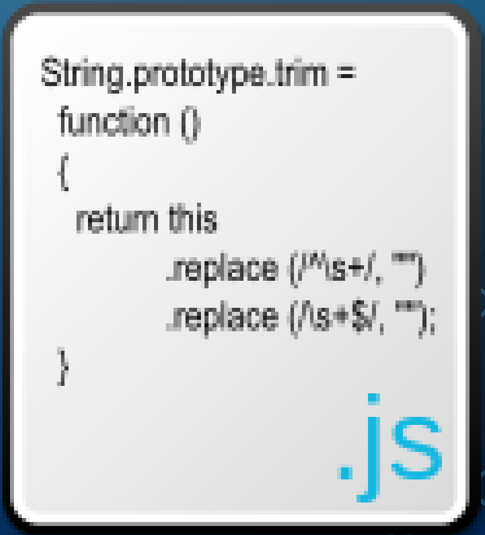
9.3-CSS

INTRODUCCIÓN A JAVASCRIPT

JavaScript is a high level programming language, fully focused on the web since it is executed in the browser. Along with HTML and CSS, javascript is one of the three key technologies of the WWW. JavaScript is prototype-based, object-oriented and extended use with first-class functions. This makes it a multi-paradigm language.

It was developed by Brendan Eich while working for Netscape Communications Corporation in 1995.

JavaScript has nothing to do with **Java,** the syntax changes a lot and can not be seen as a copy or anything related. In development stage JavaScript was called **Mocha,** in its first release was called **LiveScript** and finally was renamed **JavaScript** when he left along with 2D version of Netscape.



```
String.prototype.trim =
 function ()
 {
  return this
          .replace (/^\s+/, "")
          .replace (/\s+$/, "");
 }
```

Ejemplo 1

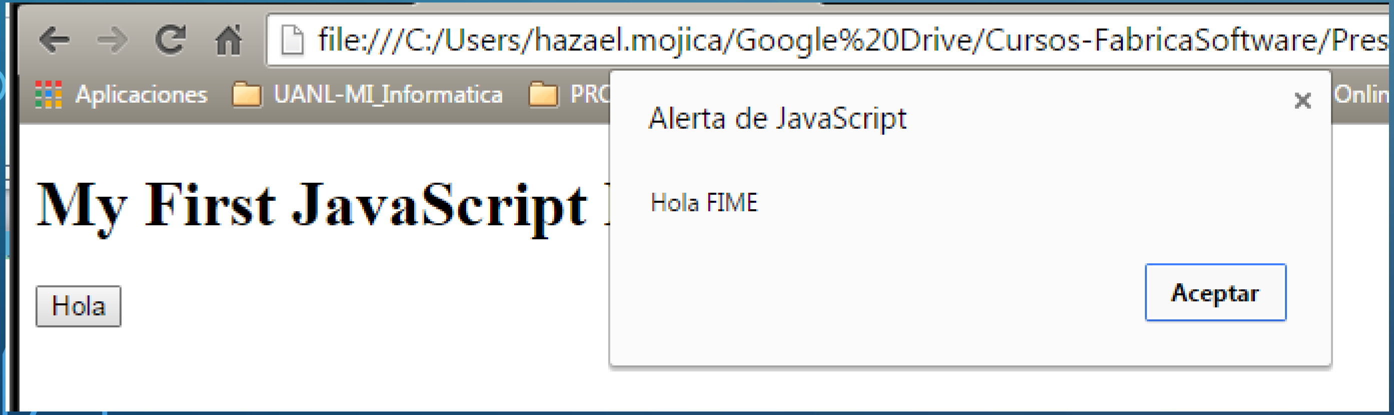
```
<!DOCTYPE html>
<html>
<head>
    <title>The title </title>
    <script type="text/javascript">
        function hola()
        {
            alert("Hola FIME");
        }
    </script>

</head>
<body>
    <h1>My First JavaScript Example</h1>
    <input type="button" onclick="hola();" value="Hola" />
</body>
</html>
```

To embed JavaScript code within the HTML, the **script** tag is used.

The **alert (...)** function is used to send a message to the user.

You can run JavaScript code in a given event **DOM** (Document Object Model).


file:///C:/Users/hazael.mojica/Google%20Drive/Cursos-FabricaSoftware/Pres
Aplicaciones
UANL-MI_Informatica
PRC
Onlin
Hola
Aceptar

My First JavaScript

Alerta de JavaScript

Hola FIME

Example 2

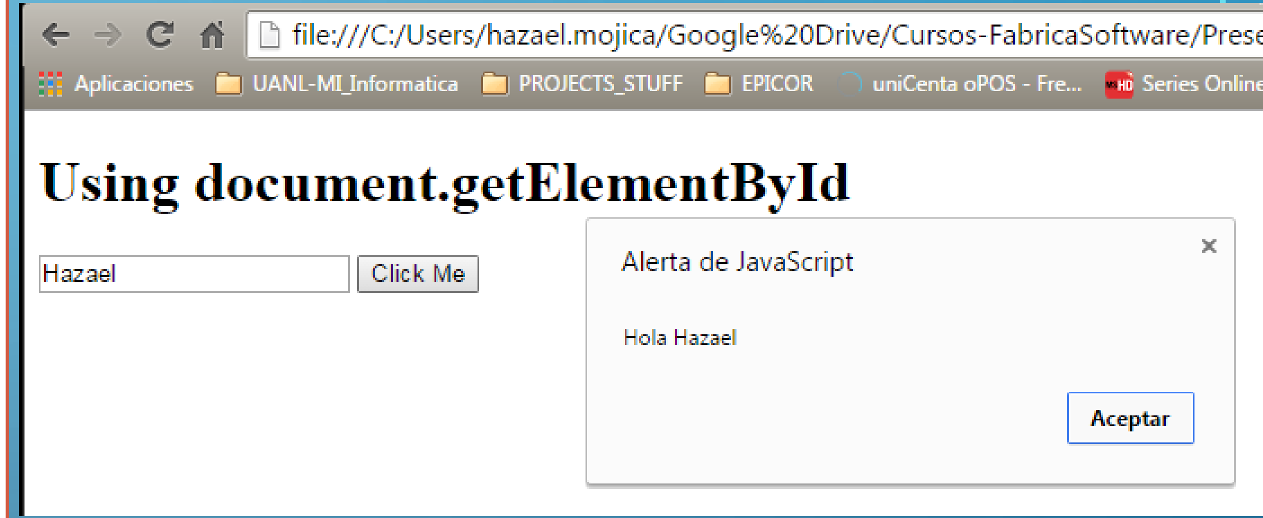
hola.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>The title </title>
    <script type="text/javascript" src="scripts.js">
    </script>

</head>
<body>
    <h1>Using document.getElementById</h1>
    <input id="text_Name" type="text" />
    <input type="button" onclick="tellMyName();" value="Click Me" />
    <h1 id="h1_name"></h1>
</body>
</html>
```

scripts.js

```javascript
function tellMyName()
{
    var text = "Hola ";
    var name = document.getElementById("text_Name").value;
    text += name;
    alert(text);
    document.getElementById("h1_name").innerHTML = text;
}
```



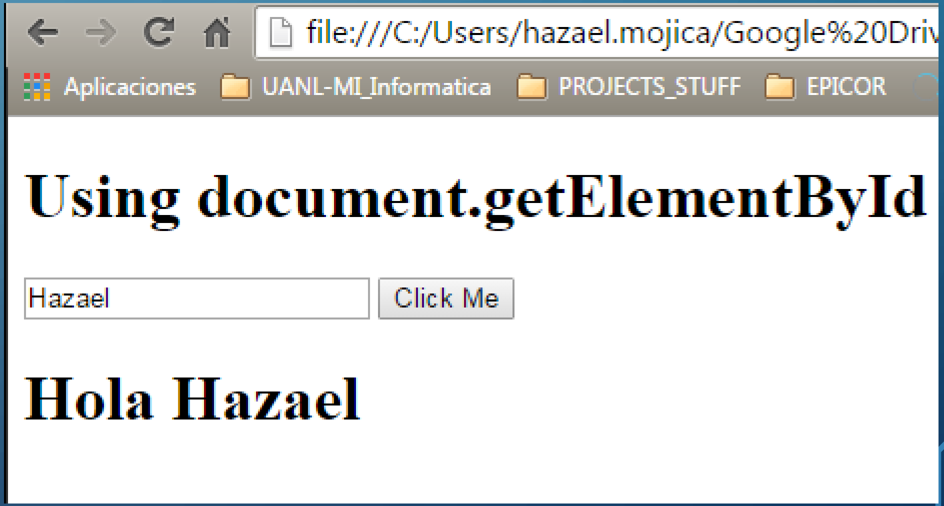
Using document.getElementById

Hazael | Click Me

Alerta de JavaScript

Hola Hazael

Aceptar



Using document.getElementById

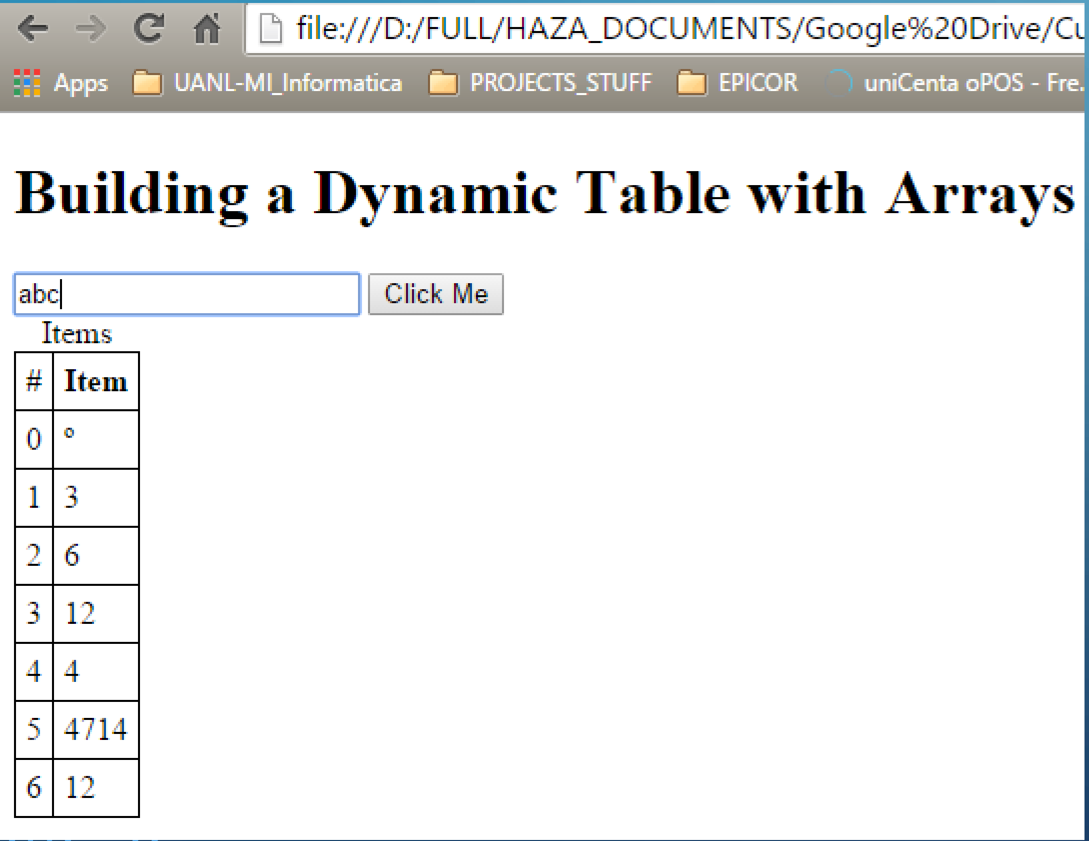
Hazael | Click Me

Hola Hazael

Example 3

hola.html

```
<!DOCTYPE html>
<html>
<head>
    <title>The title </title>
    <script type="text/javascript" src="scripts.js">
    </script>
</head>
<body>
    <link rel="stylesheet" type="text/css" href="style.css">
    <h1>Building a Dynamic Table with Arrays</h1>
    <input id="text_Item" type="text" />
    <input type="button" onclick="addItem();" value="Click Me" />
    <div id="tableContainer"></div>
</body>
</html>
```



file:///D:/FULL/HAZA_DOCUMENTS/Google%20Drive/Cu

Building a Dynamic Table with Arrays

abc

Click Me

Items

#	Item
0	°
1	3
2	6
3	12
4	4
5	4714
6	12

scripts.js

```
var tableObject = "";
var items = [];

//Esta funcion agrega un elemento al array
//y llama a la funcion displayTable()
function addItem()
{
    var item = document.getElementById("text_Item").value;
    if (item != "")
    {//Añadimos el elemento al array y desplegamos la tabla
        items.push(item);
        displayTable();
        document.getElementById("text_Item").value = "";
    }
}

//Esta funcion convierte el arreglo items
//en una tabla HTML y la coloca en el div
//con id "tableContainer"
function displayTable()
{
    var textTable = "<table><caption>Items</caption>";
    textTable += "<thead><tr><th>#</th><th>Item</th>";
    textTable += "</tr></thead><tbody>";
    for(var i = 0; i < items.length; i++)
    {
        textTable += "<tr><td>" + i + "</td>";
        textTable += "<td>" + items[i] + "</td>";
        textTable += "</tr>";
    }
    textTable += "</tbody></table>"
    document.getElementById("tableContainer").innerHTML = textTable;
}
```

style.css

```
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
th, td {
    padding: 5px;
    text-align: left;
}
```

http://www.w3schools.com/js/js_arrays.asp

Example 4

scripts.js

```
var tableObject = "";
var items = [];

//Esta funcion agrega un elemento al array
//y llama a la funcion displayTable()
function addItem()
{
    var item = $("#text_Item").val();
    if (item != "")
    {//Añadimos el elemento al array y desplegamos la tabla
        items.push(item);
        displayTable();
        $("#text_Item").val("");
    }
}

//Esta funcion convierte el arreglo items
//en una tabla HTML y la coloca en el div
//con id "tableContainer"
function displayTable()
{
    var textTable = "<table class=\"table\"><caption>Items</caption>";
    textTable += "<thead><tr><th>#</th><th>Item</th>";
    textTable += "</tr></thead><tbody>";
    for(var i = 0; i < items.length; i++)
    {
        textTable += "<tr><td>" + i + "</td>";
        textTable += "<td>" + items[i] + "</td>";
        textTable += "</tr>";
    }
    textTable += "</tbody></table>"
    $("#tableContainer").html(textTable);
}
```

Using jQuery many instructions as **document.getElementById** are simplified to **$ ("# id")**

JQuery is a library, written 100% in Javascript that facilitates control the DOM of a website and has a more natural syntax . AJAX calls are possible to do easily in addition to build-in JSON which facilitates the storage of information.

Bootstrap is a set of JavaScript / jQuery and CSS code that allows us to create responsive web pages (which identify the size of the client screen and adapt to it) in a simple manner in addition to providing a range of modern styles made entirely with pure code, which increases web sites performance.

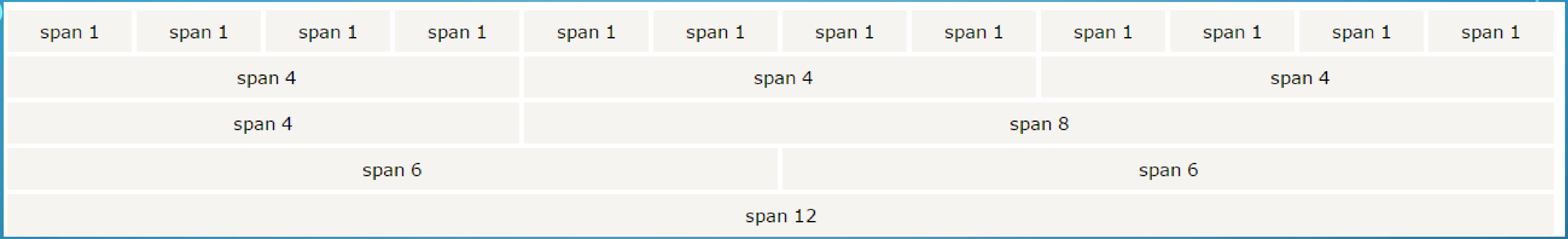
Check the HTML index.html in the sample code **9.7-Bootstrap.**

http://getbootstrap.com/
http://getbootstrap.com/examples/theme/
https://bootswatch.com/cerulean/

Bootstrap Grid

Bootstrap Grid System

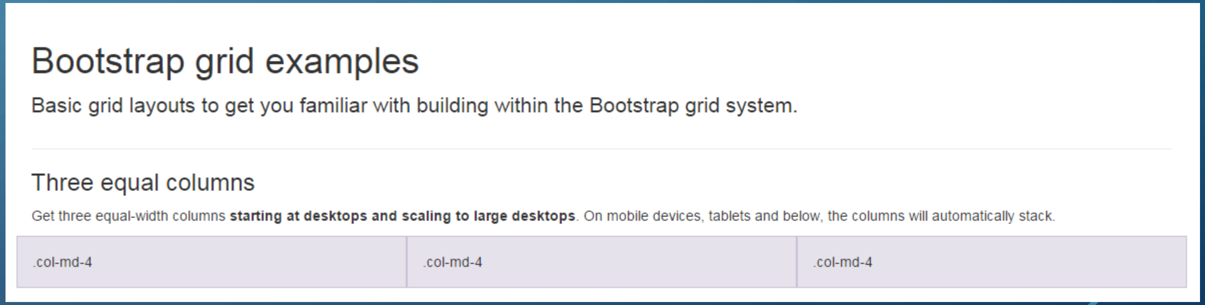
Bootstrap's grid system allows up to 12 columns across the page.
If you do not want to use all 12 column individually, you can group the columns together to create wider columns:



Bootstrap' grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items where stacked on top of each other.

Basic Structure of a Bootstrap Grid

```
<div class="container">
  <div class="row">
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    ...
  </div>
</div>
```


Bootstrap grid examples

Basic grid layouts to get you familiar with building within the Bootstrap grid system.

Three equal columns

Get three equal-width columns **starting at desktops and scaling to large desktops**. On mobile devices, tablets and below, the columns will automatically stack.

.col-md-4	.col-md-4	.col-md-4

http://getbootstrap.com/examples/grid/
http://www.w3schools.com/bootstrap/bootstrap_grid_system.asp

Example 5

Exercise Example

Using Bootstrap Basic template to create a website that works entirely with JavaScript / JQuery and HTML components.

The purpose of this website is to capture personal data, store them in a two-dimensional array and display them in table form.

TIP: Giving right click -> View Source to this web page: http://getbootstrap.com/examples/theme/# we can copy and paste the component we need, Bootswatch has an intuitive web page where you can copy the components you need, check out their theme Cerulean.

See source code in 9.9-Bootstrap

In class exercise

Taking as an example the previous exercise to create a responsive website (using Bootstrap) operating entirely with Jquery / Javascript and do any of the following purposes:

1. **Function as a calculator.**
    1. That is, there is an input in which the user can write numbers
    2. You must have numbered buttons (should be large, remember that there are users using tablet and mobile) and operations +, -, *, /, =
2. **Function as a ISR calculator** as seen for example in .Net Windows Forms
    1. That is, you must calculate the income tax and net salary from a gross salary entered by the user.

WEB SERVERS

A server is a computer with hardware and software designed to work 24 hours every day and its main function is to "listen" client computers requests and send response, your software and hardware must be robust enough to operate without need to reset.

The **Debian** operating system boasts not need to be restarted for updates software packages (unlike Windows Server) and that has kept servers running for up to a year without a single "downtime" and the only time it had to shut down was for hardware failure.

A "Web Server Software" is a piece of software programmed to listen for HTTP requests, process and send the requested information via the network.

The most widely used web server in the world is Apache, which can be run on most OS and supports running multiple programming languages (PHP, Perl, Python, Java).
Microsoft has its own web server that supports running .Net, this is called IIS (Internet Information Services). His most recent to date version is IIS8.

LAMP
Linux Apache MySQL PHP
XAMPP

Apache, MySQL, PHP, Perl

IIS8
Windows Server
Internet Information Services 8.0
microsoft.com/web

The inside and front of a Dell PowerEdge web server, a computer designed for rack mounting

HTTP METHODS: GET VS. POST

Two commonly used methods for a request-response between a client and server are: GET and POST.

GET - Requests data from a specified resource

POST - Submits data to be processed to a specified resource

The GET Method

Note that the query string (name/value pairs) is sent in the URL of a GET request:

/test/demo_form.asp?name1=value1&name2=value2

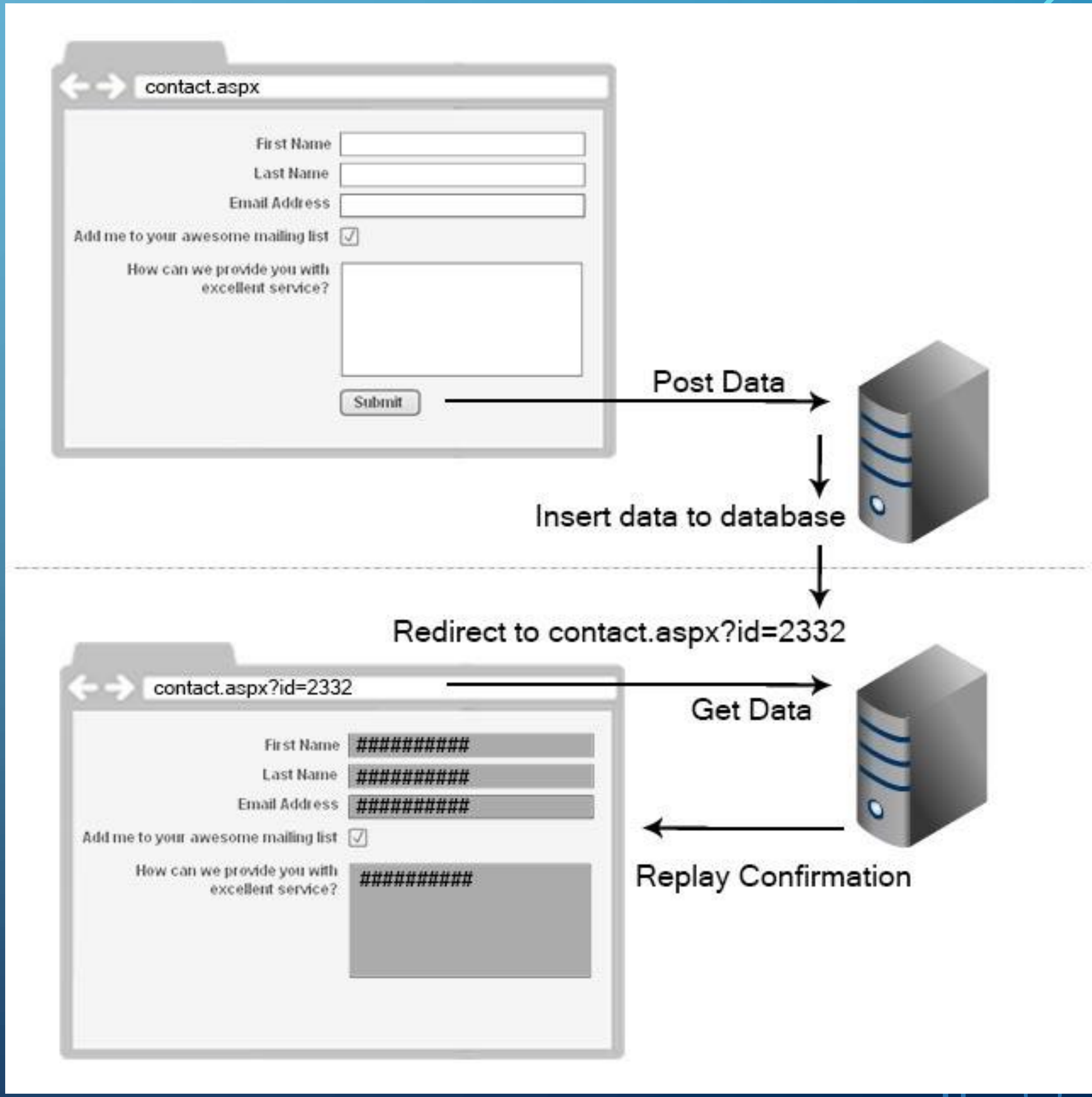
- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests should be used only to retrieve data

Usado principalmente para obtener datos de una consulta a base de datos en el servidor.

The POST Method

POST requests are never cached
POST requests do not remain in the browser history
POST requests cannot be bookmarked
POST requests have no restrictions on data length

Usado principalmente para mandar datos sensibles (contraseñas) y esperar un dato de retorno sencillo (Un "OK").


contact.aspx
First Name
Last Name
Email Address
Add me to your awesome mailing list
How can we provide you with excellent service?
Submit
Post Data
Insert data to database
Redirect to contact.aspx?id=2332
contact.aspx?id=2332
Get Data
First Name ##########
Last Name ##########
Email Address ##########
Add me to your awesome mailing list
How can we provide you with excellent service? ##########
Replay Confirmation

http://www.w3schools.com/tags/ref_httpmethods.asp

AJAX

AJAX is Asynchronous JavaScript And XML.
AJAX technology allows calls to web server without reloading the entire page, ie, the programmer using JavaScript code can call the server (send and receive information) so that a complete page reload is avoided giving a more intuitive experience for the end user.

AJAX works thanks to XMLHttpRequest protocol created by Microsoft for IE 5.0, it became standard in 2006 and is currently supported by all browsers.

```
<!DOCTYPE html>
<html>
<head>
    <script>
        function loadXMLDoc() {
            var xmlhttp;
            if (window.XMLHttpRequest) {// code for IE7+, Firefox, Chrome, Opera, Safari
                xmlhttp = new XMLHttpRequest();
            }
            else {// code for IE6, IE5
                xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
            }
            xmlhttp.onreadystatechange = function () {
                if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
                    document.getElementById("myDiv").innerHTML = xmlhttp.responseText;
                }
            }
            xmlhttp.open("GET", "ajax_info.txt", true);
            xmlhttp.send();
        }
    </script>
</head>
<body>

    <div id="myDiv"><h2>Let AJAX change this text</h2></div>
    <button type="button" onclick="loadXMLDoc()">Change Content</button>

</body>
</html>
```

In the "antiquity" it had to create a XMLHttpRequest () object and handle the somewhat complicated logic to make a call AJAX

This example loads a document called ajax_info.txt the server via GET and place its contents into a DIV tag.

http://www.w3schools.com/ajax/ajax_xmlhttprequest_send.asp

AJAX WITH JQUERY

```
$.ajax({
        type: "POST",
        url: “responsePage.php",
        dataType: "html",
        data:
        {
            var1: "var1val",
            var2: "var3val",
            var3: "var2val",
        },
        success:
            function (data)
            {

            },
        error:
            function ()
            {

            }
   });
```

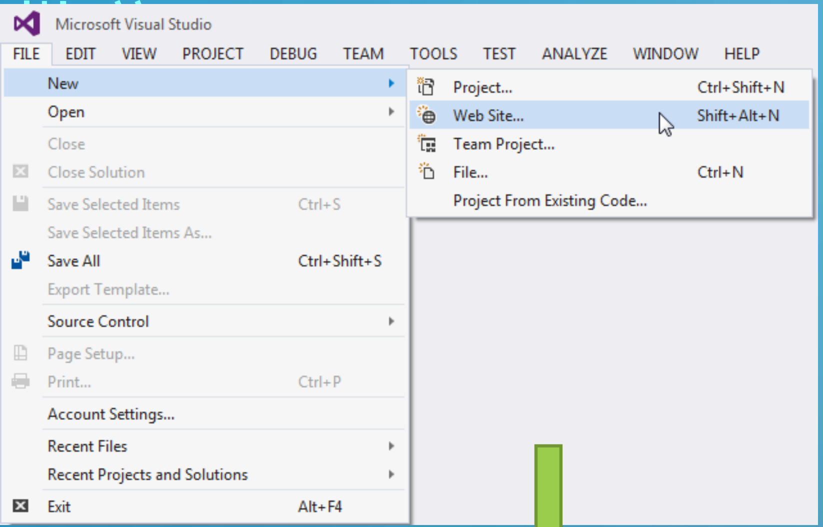
- type: is the type of call “POST”or “GET”
- url: the url of the page or file to call
- dataType: the type of data expected to be received, usually “HTML” or “JSON”.
- data: variables are to be transmitted and their respective values in JSON format. On the server side it receives the variable using the name set, so for example in PHP would be:
  $ PhpVar1 = $ _ POST ["var1"];

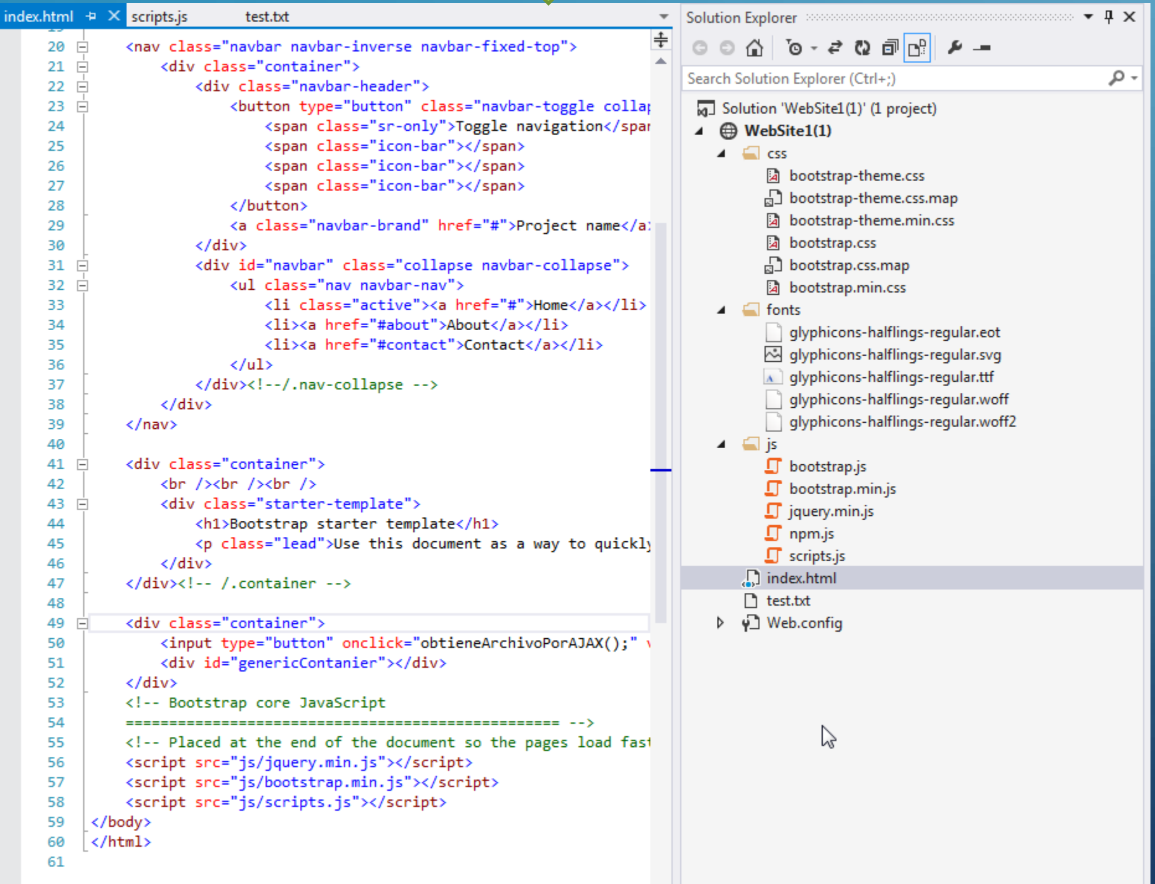
success and *error* is two callback functions, these are used to continue the actions to take once the call has been completed with success or failure respectively.

In turn, the success function receives a variable parameter which contains information received call.

Example 6

Get a text file from the server via AJAX using a GET call.

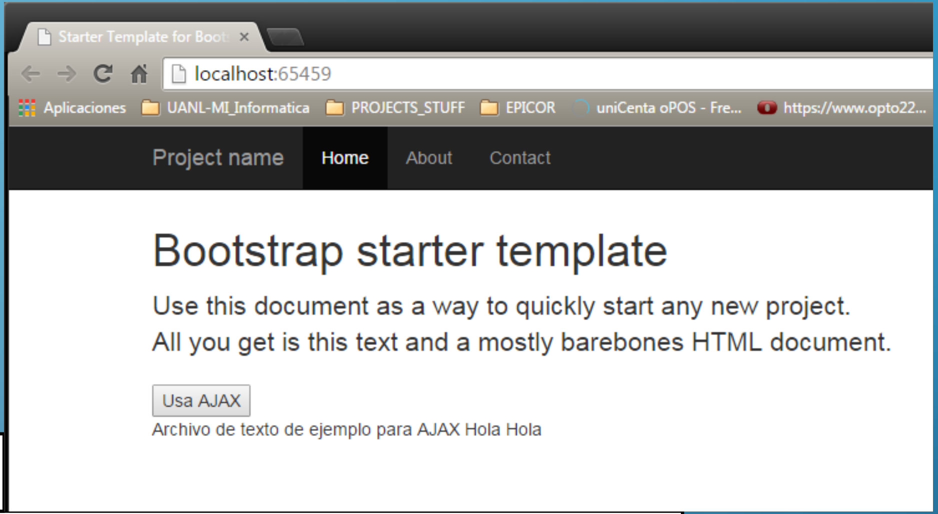




```
    <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle colla
                    <span class="sr-only">Toggle navigation</spa
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Project name</a
            </div>
            <div id="navbar" class="collapse navbar-collapse">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="#">Home</a></li>
                    <li><a href="#about">About</a></li>
                    <li><a href="#contact">Contact</a></li>
                </ul>
            </div><!--/.nav-collapse -->
        </div>
    </nav>

    <div class="container">
        <br /><br /><br />
        <div class="starter-template">
            <h1>Bootstrap starter template</h1>
            <p class="lead">Use this document as a way to quickl
        </div>
    </div><!-- /.container -->

    <div class="container">
        <input type="button" onclick="obtieneArchivoPorAJAX();"
        <div id="genericContanier"></div>
    </div>
    <!-- Bootstrap core JavaScript
    ================================================== -->
    <!-- Placed at the end of the document so the pages load fas
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/scripts.js"></script>
</body>
</html>
```



Bootstrap starter template

Use this document as a way to quickly start any new project.
All you get is this text and a mostly barebones HTML document.

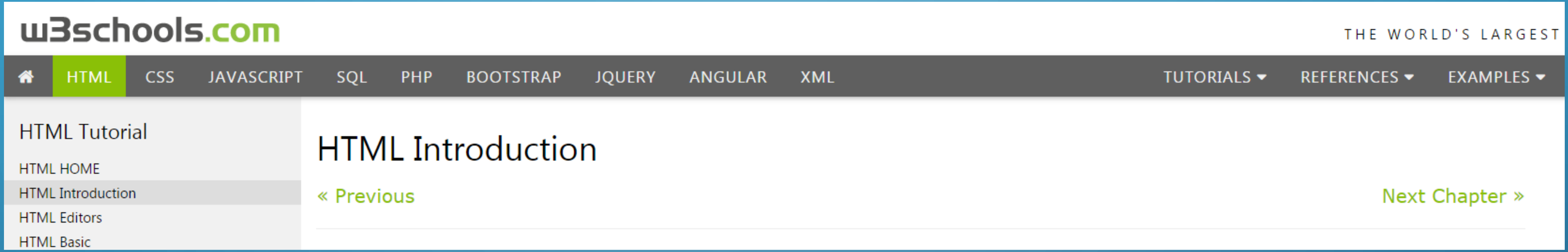
Archivo de texto de ejemplo para AJAX Hola Hola

scripts.js

```
function obtieneArchivoPorAJAX()
{//Usando la librería de AJAX de jquerys
    $.ajax({
        type: "GET",
        url: "test.txt",
        dataType: "html",
        data:
        {
        },
        success:
            function (data)
            {
                $("#genericContanier").html(data);
            },
        error:
            function ()
            {
                alert("ERROR");
            }
    });
}
```

HOMEWORK

1. Take the complete HTML tutorial http://www.w3schools.com/html/default.asp
    1. Read from HTML Introduction.
    2. To HTML HEAD



w3schools.com THE WORLD'S LARGEST

HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP JQUERY ANGULAR XML TUTORIALS REFERENCES EXAMPLES

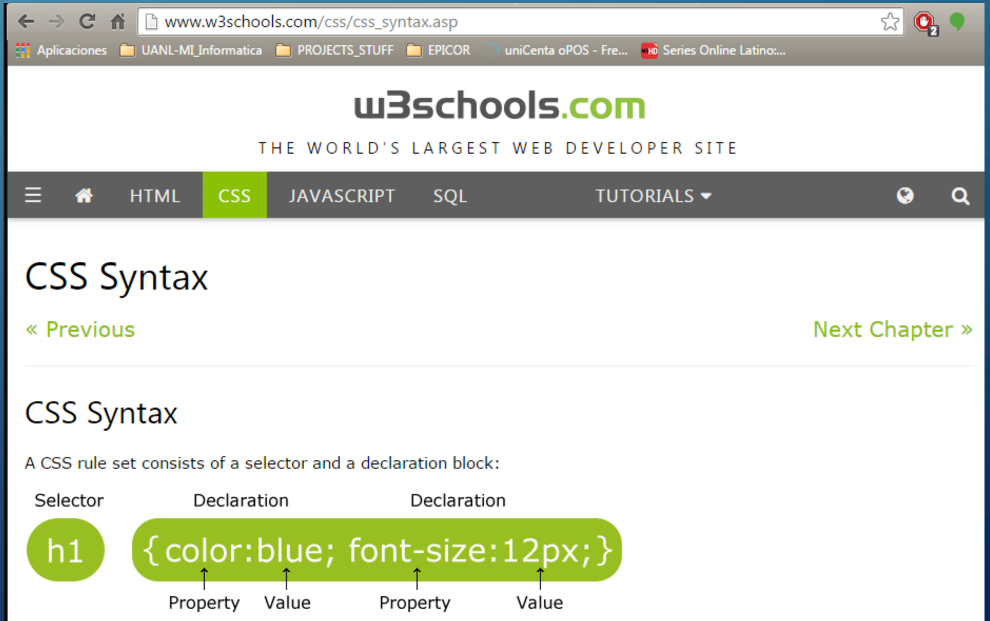
HTML Tutorial

HTML HOME
HTML Introduction
HTML Editors
HTML Basic

HTML Introduction

« Previous Next Chapter »

2. Take the tutorial for CSS http://www.w3schools.com/css/default.asp
    1. From CSS Intro
    2. To CSS Attr Selectors



www.w3schools.com/css/css_syntax.asp

Aplicaciones UANL-MI_Informatica PROJECTS_STUFF EPICOR uniCenta oPOS - Fre... Series Online Latino:...

w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE

HTML CSS JAVASCRIPT SQL TUTORIALS

CSS Syntax

« Previous Next Chapter »

CSS Syntax

A CSS rule set consists of a selector and a declaration block:

Selector Declaration Declaration

h1 {color:blue; font-size:12px;}

Property Value Property Value