

Tema 1.1

Introducción a Internet y al Web

Cientes, servidores y la nube



◆ **Cientes:** dan acceso a

- La información y los servicios de Internet

◆ **Servidores:** alojan la información y los servicios

- Se agrupan en grandes granjas de servidores

◆ **La nube:** conjunto de terminales y servidores

- interconectados con aplicaciones y protocolos de Internet
 - ◆ a través de redes de fibra optica, cable, inalambricas,

Cientes y navegadores



◆ **Cientes** de acceso a Internet mas importantes

- PCs, portatiles, tabletas, telefonos inteligentes

◆ **Navegador (browser)** programa de acceso a servidores

- Siguiendo normas del Web: **URL, HTTP, HTML, CSS y JS**
 - ◆ p.e. Chrome, Firefox, Internet Explorer, Opera, Safari, ...



◆ **Tiendas de aplicaciones**

- Instalan aplicaciones en móviles y tabletas
 - ◆ Las aplicaciones usan las normas del Web (URL, HTTP,)

Servidor



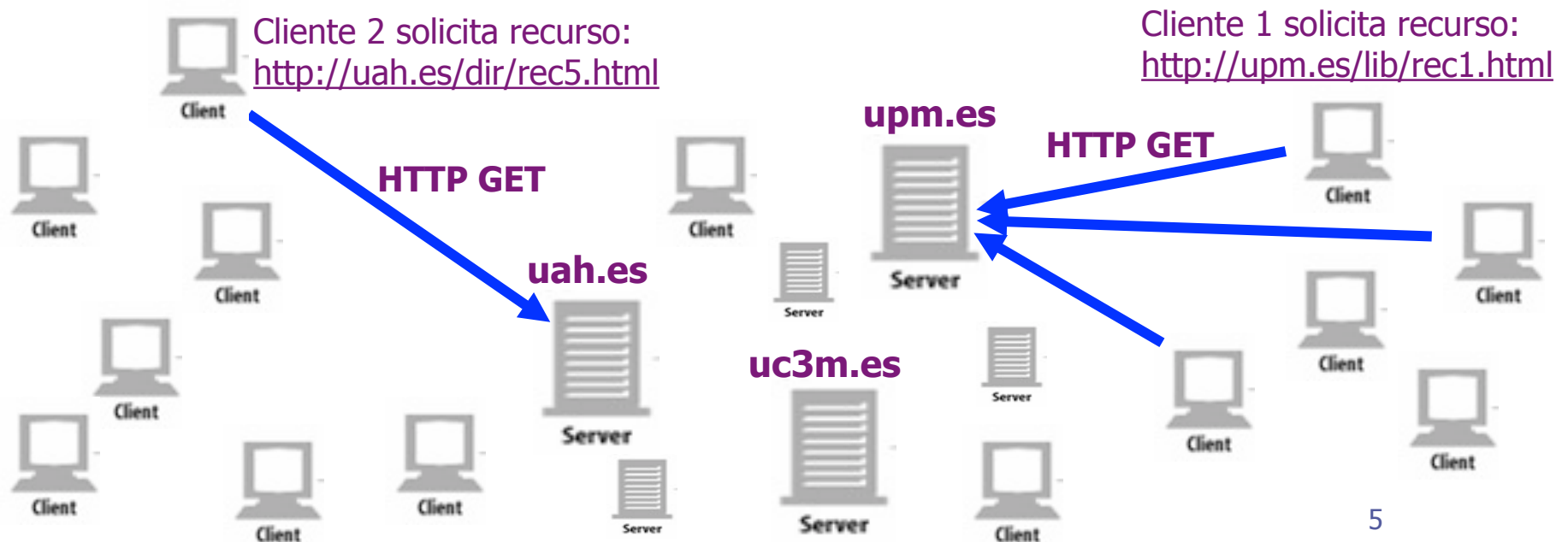
- ◆ Contiene información y servicios
- ◆ Tiene una dirección “**conocida**” en Internet
 - Por ejemplo, **upm.es**, **dit.upm.es**, **google.com**, ...
 - ◆ a cada dirección de dominio le corresponde una dirección IP (binaria)
 - 192.9.0.144, 2001:db8:85a3::8a2e:370:7334,
- ◆ **Puerto**: punto de acceso a una aplicación en un servidor
 - Si el servicio no está en el puerto por defecto este debe incluirse
 - ◆ Por ejemplo: **dit.upm.es:8080**, **192.9.0.144:8080**
- ◆ Los servidores suelen estar virtualizados en granjas de servidores
 - Accesibles a través de la nube

URL y HTTP: dirección y protocolo

◆ URL: dirección de un recurso (<http://upm.es/lib/rec.html>)

◆ Protocolo **HTTP** o **HTTPS** (seguro)

- **GET**: trae al cliente un recurso identificado por un **URL**
- POST, PUT, DELETE, ...: otros comandos de HTTP (HTTPS)



Componentes de un URL



◆ Componentes mas importantes de un URL

- <protocol://><host><path>
 - ◆ **protocol**: protocolo utilizado, en un acceso Web es HTTP o HTTPS
 - ◆ **host**: dirección de dominio de un servidor en Internet
 - ◆ **path**: camino que identifica el recurso dentro del servidor

◆ Ejemplo

- <http://upm.es/lib/rec1.html>
 - ◆ **http**: traer recurso con protocolo **HTTP**
 - ◆ **upm.es**: del servidor con dirección de dominio **upm.es**
 - Puede utilizarse una dirección IP también: **192.0.6.211**
 - También puede incluir el puerto: **upm.es:16** o **192.0.6.211:16**
 - ◆ **/lib/rec1.html**: contenido en el fichero: **/lib/rec1.html**



URLs relativos



◆ URL relativos

- son relativos al URL del recurso que está cargado
 - ◆ Están formados solo por un **path**, porque se refieren al mismo servidor

◆ Path relativo: **<path>**

- `quiz/page.html`
 - ◆ camino relativo desde el directorio del recurso actual

◆ Path absoluto: **</path>**

- `/lib/quiz/page.html`
 - ◆ camino absoluto desde el directorio raíz del servidor



URL con pregunta o query



◆ Se puede añadir una pregunta o query con parámetros

- <protocol://><host><path><**query**>
 - ◆ **pregunta o query**: parámetros con la siguiente sintaxis
 - **?param1=valor1¶m2=valor2**

◆ Ejemplo

- **http://upm.es/lib/rec1.html?usuario=1527**
 - ◆ **?usuario=1527**: envía el parámetro “usuario=1527”



Ancla o Anchor



◆ Se puede añadir un ancla (anchor)

- `<protocol://><host><path><anchor>`
 - ◆ **anchor**: identifica un elemento dentro una página Web
 - **#capitulo3** identifica el elemento
 - `<h1 id="capitulo3"> </h1>`

◆ Ejemplo

- `http://upm.es/lib/rec1.html#capitulo3`
 - ◆ **#capitulo3**: identifica el capitulo 3 de la página HTML



Aplicación Web: HTML, CSS y JavaScript

◆ HTML

- Lenguaje de marcado de páginas Web
 - ◆ define la estructura del contenido
- En WebApps define el interfaz

◆ CSS

- Define la visualización

◆ JavaScript

- Lenguaje de programación de aplicaciones de cliente



```
<!DOCTYPE html><html>
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8">

  <style type="text/css">
    body {color: blue;}
  </style>
</head>

<body>
  <h3>Fecha y hora</h3>

  <script type="text/javascript">
    document.write(new Date());
  </script>
</body>
</html>
```



WebApps o aplicaciones de cliente

◆ Aplicaciones que **residen en un servidor**

- pero se **ejecutan en un cliente**

- ◆ El cliente trae la aplicación del servidor con el **protocolo HTTP (GET)**

- El **URL** es su dirección: <http://upm.es/apps/webapp.html>

◆ Las apps. se construyen con las tecnologías del Web

- URLs, HTTP, HTML, CSS y JavaScript

1) Cliente solicita WebApp identificada con URL

2) Script se ejecuta al cargar la página Web en el navegador:



Solicitud HTTP asociada a un URL

Respuesta HTTP: página Web con script



Servidor sirve fichero identificado por URL



Ejercicio

- ◆ Si el servidor **upm.es** contiene el recurso **/lib/upm.html**

```
<!DOCTYPE html>
<html> ... <body>
  <h1 id="id1" >Servidor upm</h1>
</body>
</html>
```

upm.es



- ◆ y el servidor **uah.es** contiene el recurso **/lib/uah.html**

```
<!DOCTYPE html>
<html> ... <body>
  <h1 id="id2" >Servidor uah</h1>
</body>
</html>
```

uah.es



- ◆ que recurso identificarán los siguientes URLs

<http://upm.es//lib/upm.html#id1> => el fichero completo upm.html,
el fichero completo uah.html,
el elemento <h1 id="id1" >Servidor UPM</h1> de upm.html,
el elemento <h1 id="id2" >Servidor UAH</h1> de uah.html
elemento o recurso inexistente

<http://upm.es//lib/uah.html#id1> => (los mismos recursos)

<http://upm.es//lib/upm.html> => (los mismos recursos)

<http://upm.es//lib/uah.html> => (los mismos recursos)

Tema 1.4

HTML

Estructura de un documento HTML

Qué es HTML

- Es un **lenguaje de marcas** para formatear y estructurar un documento, que puede leerse en cualquier plataforma.
- Estandarizado en la norma ISO de **SGML** (Standard Generalized Markup Language).
- El **W3C** desarrolla especificaciones técnicas y directrices, de forma que se pueda asegurar una alta calidad técnica y editorial.

Lenguaje HTML

Un documento HTML inserta en el texto **etiquetas** que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos.

Las marcas están encerradas entre los símbolos "<" y ">" que la activa y terminan su función con el mismo carácter precedido por la barra inclinada es decir "</" y ">".

<etiqueta>texto afectado **</etiqueta>**

Las marcas funcionan por parejas, algunas marcas no requieren su pareja de cierre (**empty elements**) como el salto de línea **
** o La línea horizontal **<hr>** ...

Editores HTML

Antes de comenzar al trabajar con un editor específico, es recomendable conocer el código.

Podemos usar programas que trabajen con **texto plano**, sin añadir sus propias marcas de edición, en Windows el **Bloc de Notas** y en Macintosh el **TextEdit**. TextEdit está por defecto en Rich Text es necesario configurarlo para que guarde tu trabajo como archivos de texto plano.

Los documentos HTML deben tener la **extensión** html o htm.

Usar un editor **wysiwyg** como el Adobe *Dreamweaver* o el *KompoZer* (<http://www.kompozer.net/>) o un editor con ayudas visuales como el *sublime* (<http://www.sublimetext.com/>) nos facilitará las cosas.

Estructura HTML

Un documento HTML tiene tres etiquetas que describen su estructura general de un documento y dan una información sencilla sobre él. Estas etiquetas no afectan a la apariencia del documento y solo interpretan y filtran los archivos HTML.

`<html>`, `<head>` y `<body>`

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Título del documento</title>
6  </head>
7
8  <body>
9  Mi primera página
10 </body>
11 </html>
12
```

Estructura HTML

Declaración <!DOCTYPE> de versión documento HTML para que se visualice correctamente.

Para el HTML5 será: <!DOCTYPE HTML>

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Título del documento</title>
6 </head>
7
8 <body>
9 Mi primera página
10 </body>
11 </html>
12
```

Editores HTML

Ejemplos

	DOCTYPE
HTML 5	<code><!DOCTYPE html></code>
HTML 4.01 Strict	<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>
HTML 4.01 Transitional	<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>
XHTML 1.0 Strict	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"></code>
XHTML 1.0 Transitional	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"></code>

Estructura HTML

<HTML> para identificar que esta codificado en este lenguaje y
Limitar, principio y fin del documento.

`<html> </html>`

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Título del documento</title>
6  </head>
7
8  <body>
9  Mi primera página
10 </body>
11 </html>
12
```

Estructura HTML

<HEAD> la cabecera, que contiene información y recursos sobre el propio documento y que no aparece en el documento, destacando el título **<TITLE>** será el nombre que aparece en la cabecera del visualizador y en los buscadores de ahí la importancia de que sea significativo.

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Título del documento</title>
6  </head>
7
8  <body>
9  Mi primera página
10 </body>
11 </html>
12
```

<head> </head>

<title> </title>

Estructura HTML

La etiqueta **meta** representa varios tipos de metadatos.

El atributo **charset** se utiliza para especificar la codificación usada en nuestra página.

`<meta charset="character_set">`

Únicamente debe existir una etiqueta *meta* con el atributo *charset* en la página y estar dentro de los primeros 512 bytes de la página.

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Título del documento</title>
6  </head>
7
8  <body>
9  Mi primera página
10 </body>
11 </html>
12
```

`<meta charset="utf-8">`

Estructura HTML

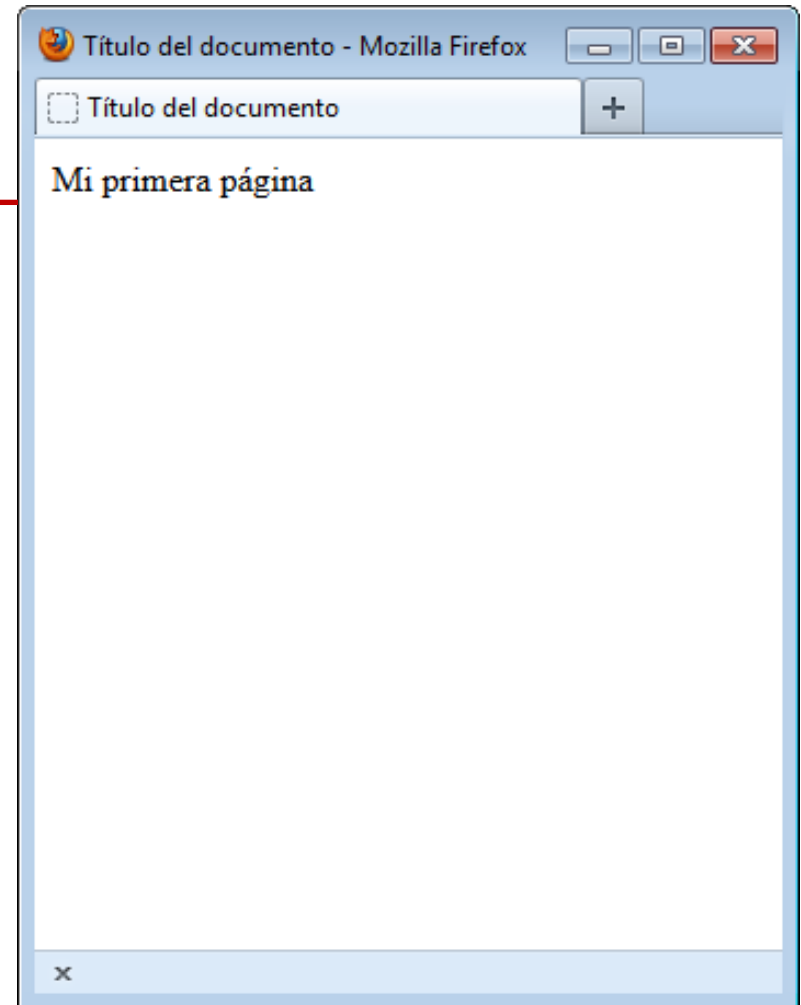
<BODY> El cuerpo contiene el documento. Puede contener atributos como son: BACKGROUND, BGCOLOR, TEXT, LINK, VLINK and ALINK. Para utilizar una imagen o color de fondo, las características por defecto del texto de la página y el modo de indicar los vínculos de hipertexto, enlazable y visitado.

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Título del documento</title>
6  </head>
7
8  <body>
9  Mi primera página
10 </body>
11 </html>
12
```

`<body> </body>`

Estructura HTML

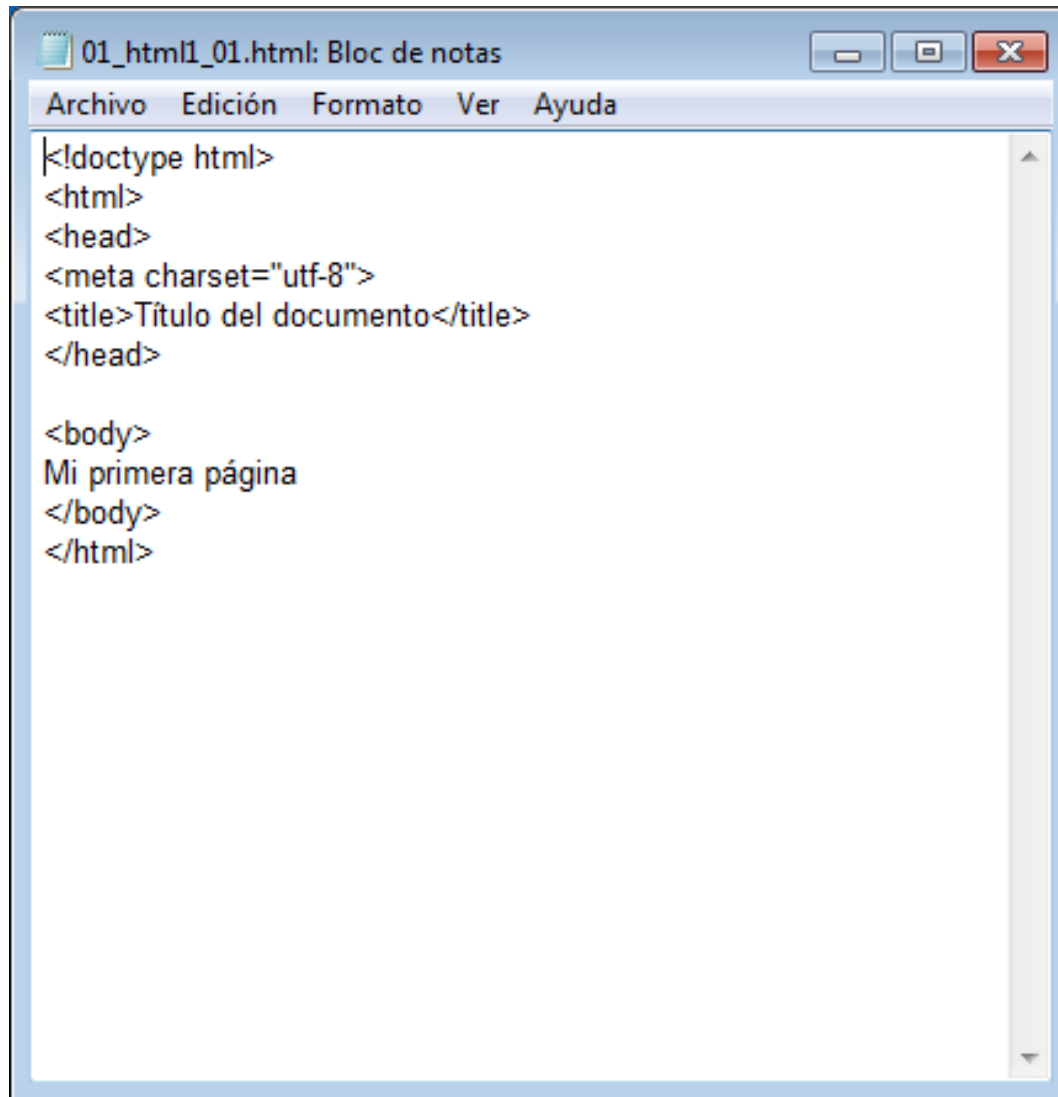
Visualización en el navegador



```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Título del documento</title>
6  </head>
7
8  <body>
9  Mi primera página
10 </body>
11 </html>
12
```


Estructura HTML

Mi primera página vista en el programa **Bloc de Notas**



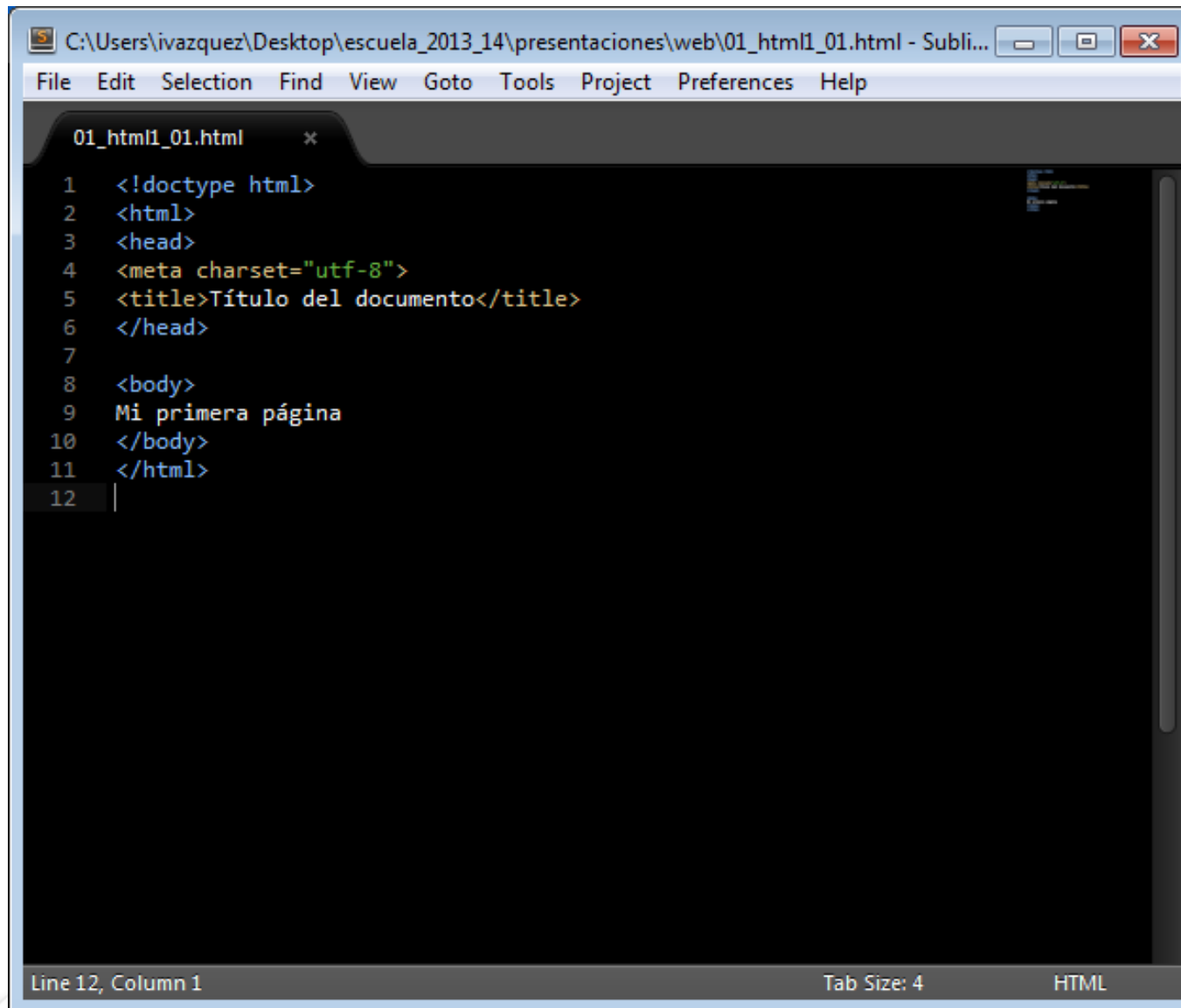
The screenshot shows a Windows Notepad window titled "01_html1_01.html: Bloc de notas". The menu bar includes "Archivo", "Edición", "Formato", "Ver", and "Ayuda". The text content is as follows:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Título del documento</title>
</head>

<body>
Mi primera página
</body>
</html>
```

Estructura HTML

Mi primera página vista en el programa **Sublime**.



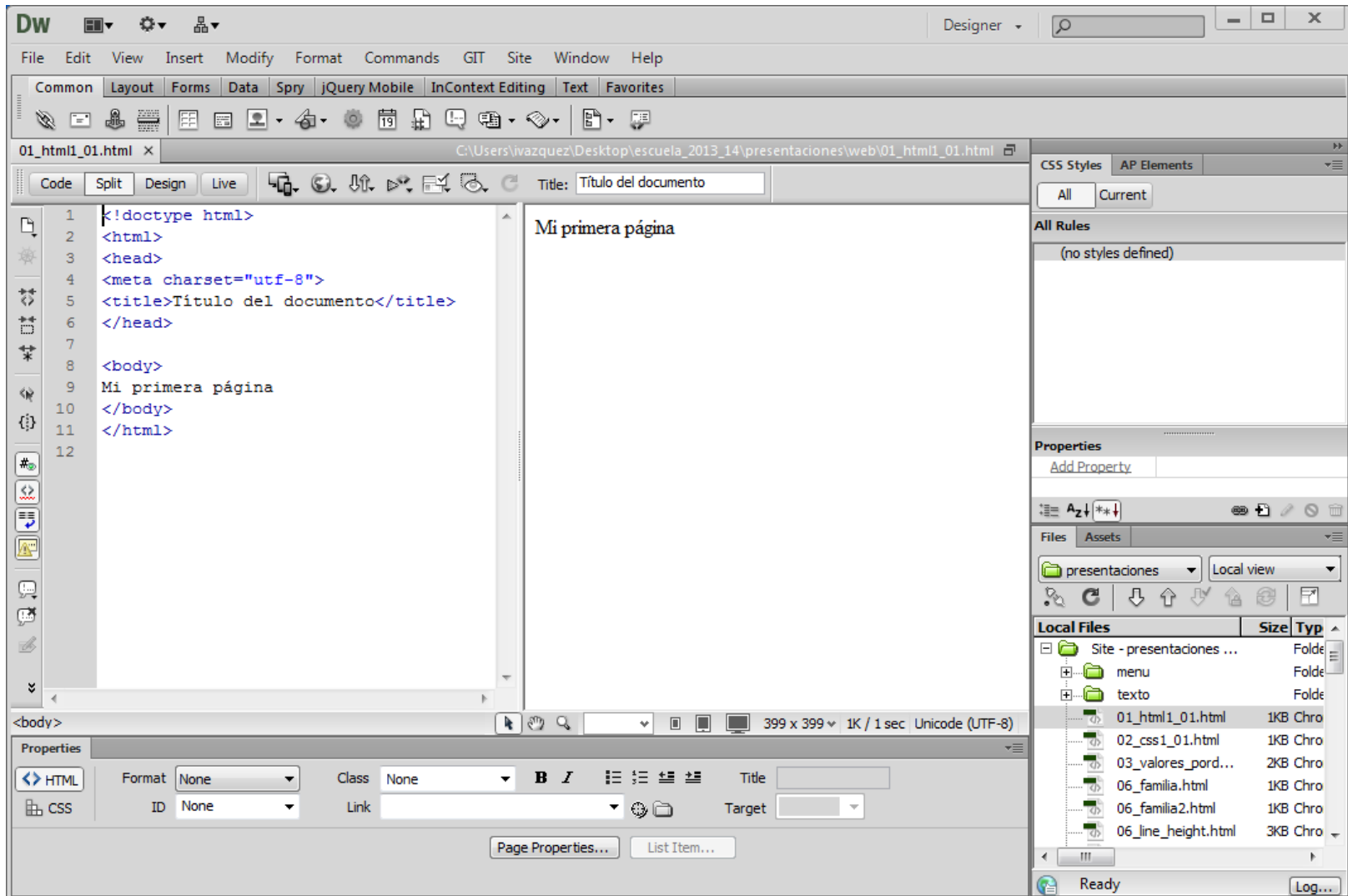
```
C:\Users\ivazquez\Desktop\escuela_2013_14\presentaciones\web\01_html1_01.html - Subli...
File Edit Selection Find View Goto Tools Project Preferences Help

01_html1_01.html x
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Título del documento</title>
6 </head>
7
8 <body>
9 Mi primera página
10 </body>
11 </html>
12 |

Line 12, Column 1 Tab Size: 4 HTML
```

Estructura HTML

Mi primera página vista en el programa **Dreamweaver**.



Recursos HTML

W3schools: educate yourself

http://www.w3schools.com/html/html5_intro.asp

W3C: HTML5 elements

<http://dev.w3.org/html5/markup/elements.html>

Woork Up: HTML5 tag reference

<http://woorkup.com/wp-content/uploads/2009/12/HTML5-Visual-Cheat-Sheet1.pdf>

Tema 1.5

CSS

Estilos de presentación CSS

HTML y CSS

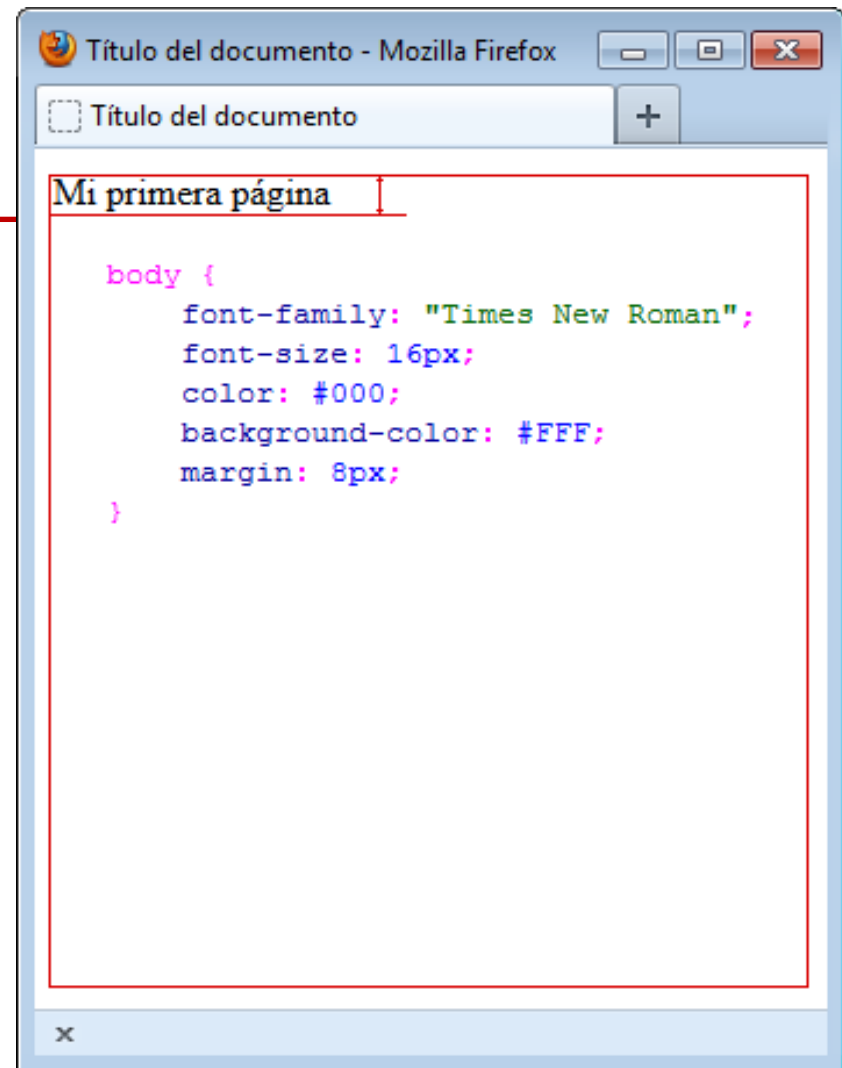
El lenguaje de marcas **HTML**, está orientado a la definición de la estructura y la semántica del documento y no a su representación gráfica.

CSS controlan el aspecto gráfico del documento mediante hojas de estilo (*style sheets*), indicando al navegador como se deben renderizar los elementos de un documento HTML. Consiguiendo así separar el aspecto del contenido.

Reglas CSS

Visualización valores por defecto CSS en el navegador

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Título del documento</title>
6 </head>
7
8 <body>
9 Mi primera página
10 </body>
11 </html>
12
```



Reglas CSS

Más información:

Internet Explorer: <http://www.iecss.com/>

Firefox:

<http://mxr.mozilla.org/mozilla-central/source/layout/style/html.css>

Chrome, Safari:

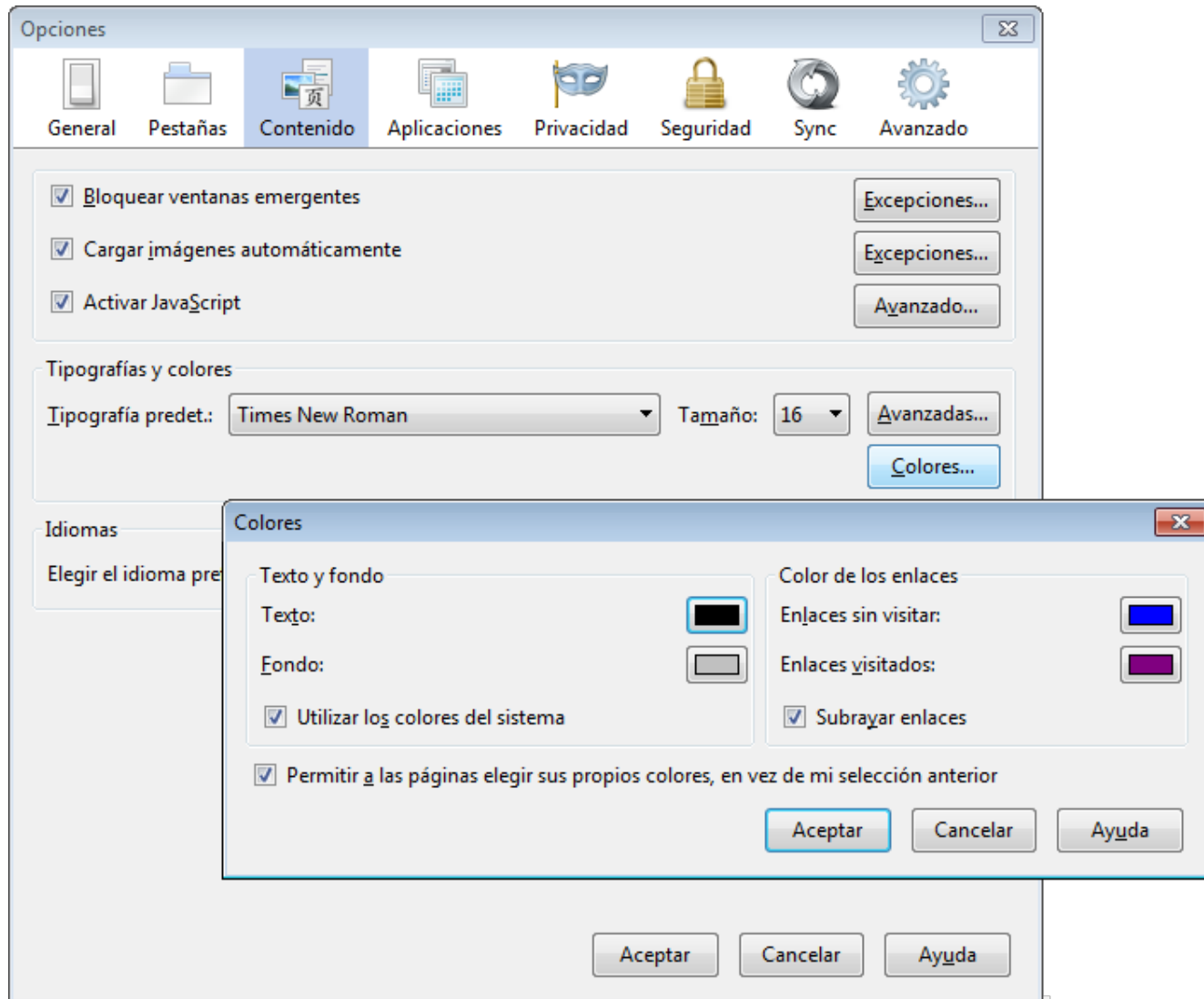
[http://trac.webkit.org/browser/trunk/Source/WebCore/css/html.cs](http://trac.webkit.org/browser/trunk/Source/WebCore/css/html.css)
[s](#)

Opera

<http://www.iecss.com/opera-10.51.css>

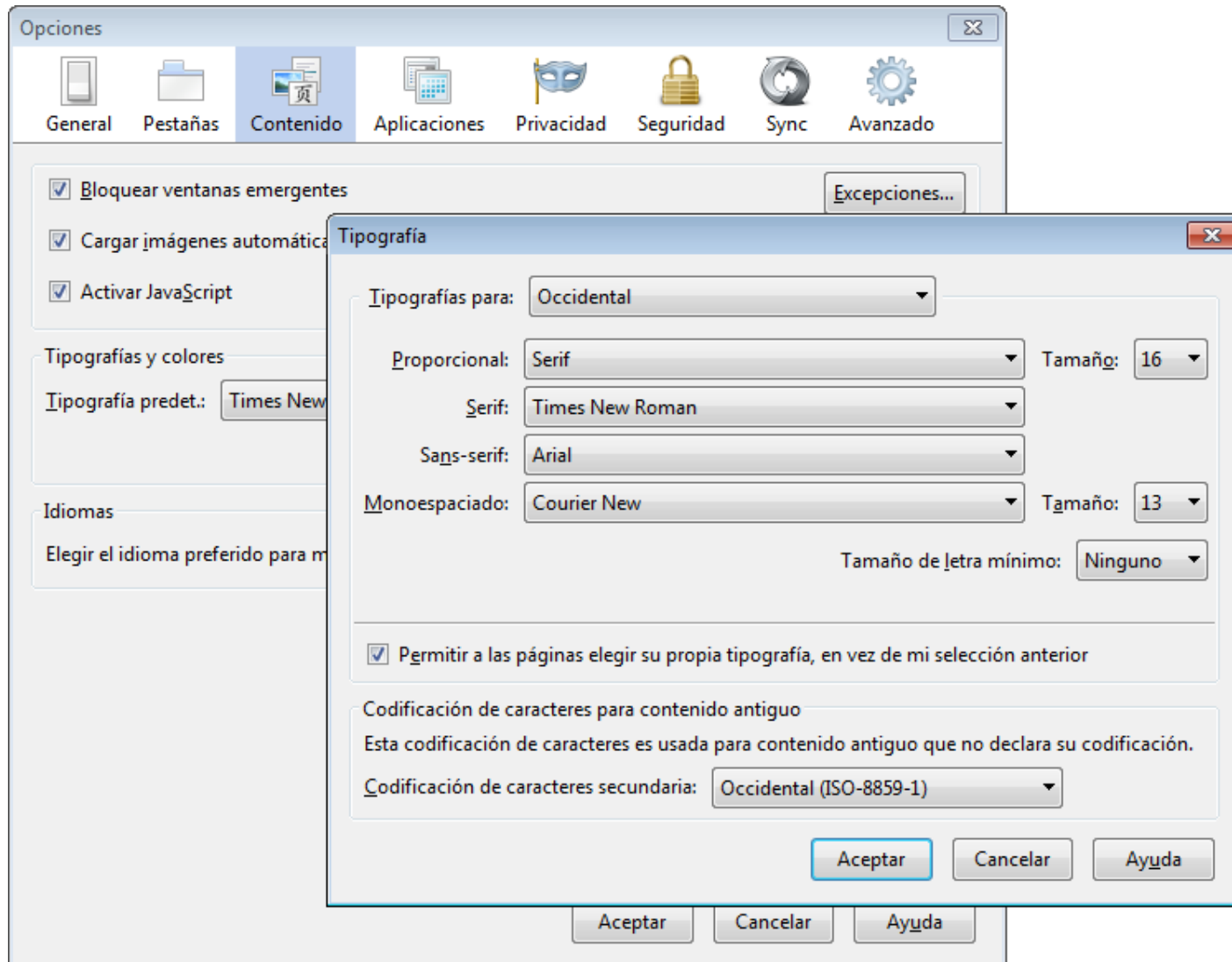
Reglas CSS

Modificación de estilos, por el usuario en el navegador: colores



Reglas CSS

Modificación de estilos, por el usuario en el navegador: tipografías



Reglas CSS

HTML tiene tres maneras de referenciar la información del estilo:

-En línea: es el método más sencillo, situando STYLE en el elemento concreto dentro de la página usando el atributo **style**, no se pueden reutilizar para varios elementos que comparten las mismas propiedades. Esta **desaprobado** por el W3C.

Se escriben las propiedades del estilo en las marcas del html como "*nombre: valor*" separadas por punto y coma.

```
<body style="color: red; ">
```

```
<p style="font-size: 16px; color: blue; font-family: Arial,  
Helvetica, sans-serif;">
```

```
Mi primera página</p>
```

Reglas CSS

-**En la cabecera** de la página: Tiene muchas de las ventajas de las hojas de estilo. Ya que un mismo estilo puede ser aplicado a varios elementos, pero sólo en esa página debe expresarse en un elemento **<STYLE>** dentro de la cabecera **<HEAD>** del documento.

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Título del documento</title>

<style type="text/css">
body {
    background-color: #FFF;
}
p {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 16px;
    color: #000;
}
</style>
</head>

<body>
<p> Mi primera página</p>
</body>
</html>
```

Reglas CSS

-**Como un archivo aparte**, que es lo más aconsejable para mantener la separación entre contenido y presentación, la solución es crear una hoja de estilo separada del documento HTML y referenciarla con:

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

La principal ventaja es que las hojas de estilo son aplicables a una o a muchas páginas. Pueden contener indicaciones de estilo para cada uno de los distintos dispositivos de presentación y según tamaño y resolución de la pantalla.

```
<link rel="stylesheet" type="text/css" media="screen"
href="sans-serif.css">
```

<http://www.w3.org/TR/css3-mediaqueries/>

Reglas CSS

Cómo se declara una norma de estilo

H1	{	color	:	green	;	text-align	:	center	}
Selector	propiedad	valor	propiedad	valor					
	declaración		declaración						
	bloque de declaración								
norma de estilo									

Comienza con una llave inicial "{" y termina con la correspondiente llave final "}"; cada *bloque de declaración*: **{*declaración*; *declaración*}** separadas por punto y coma ";".

Cada *declaración*: ***propiedad* : *valor***, (fija un *valor* para una *propiedad*).

El valor de cada *propiedad* puede ser: palabra clave, literal, número, longitud, porcentaje, URL, color, ángulo, tiempo, frecuencia o contador.

```
H1 {color : green; text-align : center}
```

Prioridad CSS

La regla CSS más importante ordenadas de menor a mayor son:

- **Navegador** (CSS por defecto del navegador)
- **Autor** (CSS definida por el diseñador)
- **Usuario** (CSS definida según las preferencias del usuario)
- **!important autor** (valores CSS importantes imprescindibles para el diseñador)
- **!important usuario** (valores CSS importantes imprescindibles para el usuario)

Recursos CSS

w3schools: CSS3 tutorial

<http://www.w3schools.com/css3/default.asp>