UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA – ICAI

Departamento de Sistemas Informáticos

Creating web pages
Chapter 6. Cascading Style Sheets CSS

Cristina Puente, Rafael Palacios

2009-2010

Formats vs Styles

- The format applied to an object only affects that particular object.
- Style sheets allow the programmer to unify format definitions in a single file.
- These definitions may affect several objects in one page or in several pages.

HTML formats

- Some HTML formats are defined by means of tags. Examples:
  - `One word displayed in <b>bold</b> and another in <i>italics</i>`
  - `<h1><big>Big heading text</big></h1>`

- HTML also allows for creating format definitions that are applied to one object. Examples:
  - `<body bgcolor="black" text="white">`
  - `<p align="right">This is a right aligned paragraph</p>`

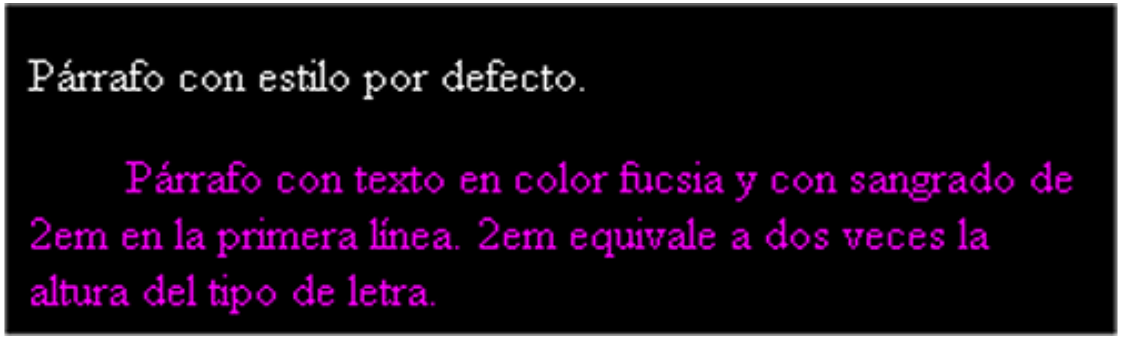
Property="value"

Reference HTML 3.2: http://www.w3.org/TR/REC-html32-19970114

In-line format definitions (CSS kind)

- Since HTML version 4, it is allowed to define styles for one object following the CSS syntax. Examples:
  - `<body style="background: black; color: white">`
  - `<p style="color: fuchsia; text-indent: 2em">Párrafo con texto en color fucsia y con sangrado de 2em en la primera línea.  2em equivale a dos veces la altura del tipo de letra.</p>`

Property: value;



Párrafo con estilo por defecto.

Párrafo con texto en color fucsia y con sangrado de 2em en la primera línea. 2em equivale a dos veces la altura del tipo de letra.

Reference HTML 4.01: http://www.w3.org/TR/html401/
Reference CSS2: http://www.w3.org/TR/REC-CSS2/

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA – ICAI

CSS styles

CSS styles

- Modern browsers support style definitions according to the CSS (Cascading Style Sheet) standard
- Styles are defined in a .css file which is loaded by the browser along with HTML files.
- All styles of the website are centralized in .css files
- If one particular HTML files redefines the style of one object, such definition replaces the default definition of the CSS files.

CSS styles

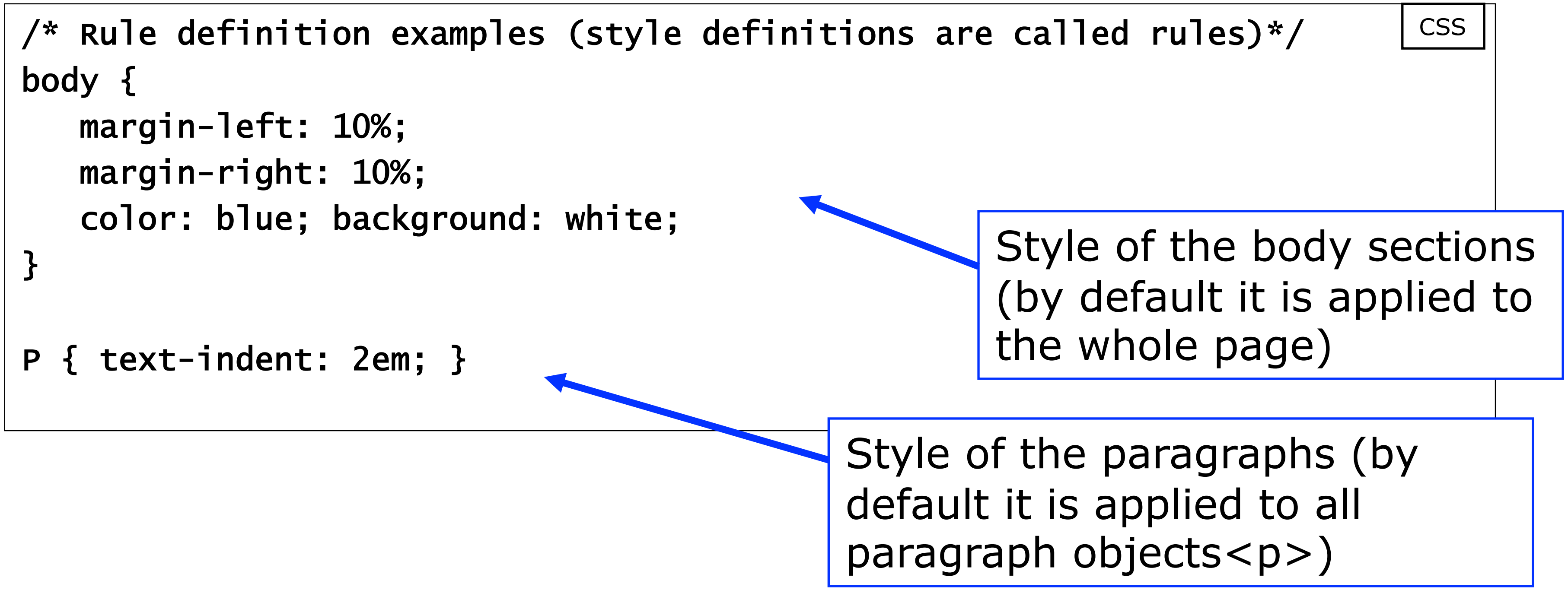
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
   <TITLE>Page with CSS styles</TITLE>
   <LINK href=“my_styles.css" type="text/css" rel="stylesheet">
</HEAD>
<BODY>
   <P>Paragraph with default style, as defined in my_styles.css file which
  is specifyed in the head section of this html files. </P>
</BODY>
```

HTML

In general it is not necessary to define additional format

Reference (link) to the style sheet

CSS styles: Definition of rules



CSS

```
/* Rule definition examples (style definitions are called rules)*/
body {
   margin-left: 10%;
   margin-right: 10%;
   color: blue; background: white;
}

P { text-indent: 2em; }
```

Style of the body sections (by default it is applied to the whole page)

Style of the paragraphs (by default it is applied to all paragraph objects<p>)

Advantages of using style sheets

- All format definitions are centralized in a file (or group of files)
- People developing HTML pages can focus on the content, they don't need to deal with the appearance
- Changing style definitions will automatically change the appearance of all the web pages which share the same CSS sheets.

Defining styles for particular objects

```
/* Rule definition examples */
body {
   margin-left: 10%;
   margin-right: 10%;
   color: blue; background: white;
}

P { text-indent: 2em; }

P.different {
   color: fuchsia;
   border: solid red;
}
```

CSS

Style of the body sections (by default it is applied to the whole page)

Style of the paragraphs (by default it is applied to all paragraph objects<p>)

Style for the paragraphs type "different".

```
<P>Paragraph in the default style</P>
<P class= "different“>Paragraph in the special format</P>
```

HTML

Using several style sheets

- It is possible to define several style sheets and apply them at the same time. It is also possible to define styles in-line (without using a separate file).
- Each new definition has priority over the previous
- It is possible to define different styles sheet to let the browser (user) decide the best appearance:

```
<LINK rel="stylesheet" href="my_styles.css" type="text/css">
<LINK rel="alternate stylesheet" href="larger.css" type="text/css" title="Large fonts">
<LINK rel="alternate stylesheet" href=“smaller.css" type="text/css" title="Small fonts">
```

- It is also possible to define styles specific for different "display types". The contents will be the same, but the appearance will be tuned for the display.

```
media="screen"
media="print"
media="aural"
media="handheld"
```

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA – ICAI

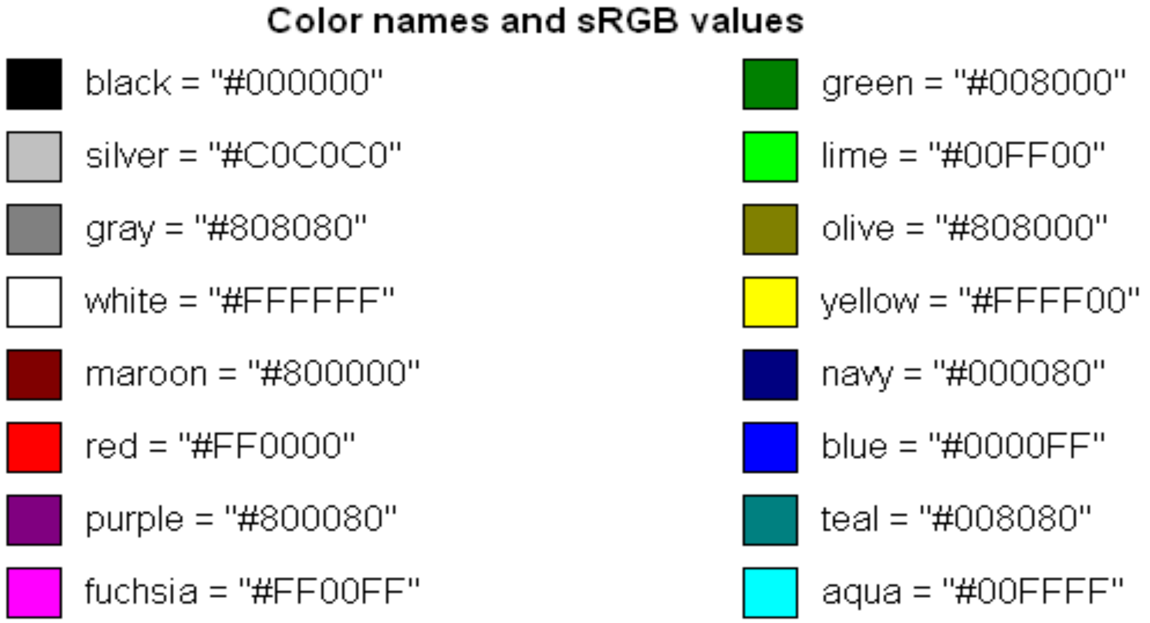
Examples of CSS2 definitions

Size definitions

- Possible size definitions
  - margin-left: 10%; ← Relative to the window width
  - margin-left: 25; ← 25 pixels. Not recommended method
  - margin-left: 2em; ← Relative to the font size
- Font size definitions
  - font-size: 12pt; ← 12 points (1 point = 1/72")
  - font-size: large; ← Relative definition (scaling factor 1.2)
    [xx-small | x-small | small | medium | large | x-large | xx-large]
  - font-size: 120%; ← Relative to the default font size

Color definitions

- Possible color definitions
  - color: purple;
  - color: #800080;
  - color: rgb(128, 0, 128);



Color names and sRGB values

black = "#000000"	green = "#008000"
silver = "#C0C0C0"	lime = "#00FF00"
gray = "#808080"	olive = "#808000"
white = "#FFFFFF"	yellow = "#FFFF00"
maroon = "#800000"	navy = "#000080"
red = "#FF0000"	blue = "#0000FF"
purple = "#800080"	teal = "#008080"
fuchsia = "#FF00FF"	aqua = "#00FFFF"

Font face definitions

font-family

- Generic values: serif | sans-serif | cursive | fantasy | monospace
- Example: font-family: "new century schoolbook", serif;

font-style

- Values: normal | italic | oblique | inherit
- Example: font-style: italic;

font-variant

- Values: normal | small-caps
- Example: font-variant: normal;

font-weight

- Values: normal | bold | bolder | lighter | 100 | 200 | 300 | 400...
- Example: font-weight: bold;

font-stretch

- Values: normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded
- Example: font-stretch: condensed;

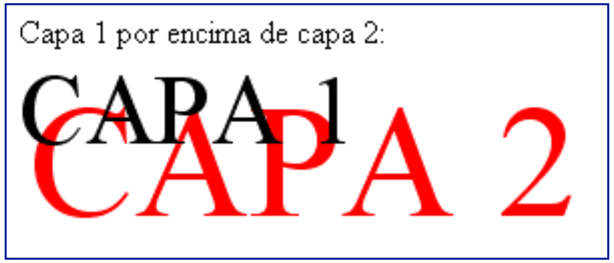
font-size

- Example: font-size: 1.2em;

DIV and SPAN

- DIV defines a section of the web page, and allows the definition of styles applied to that section.
  - For example, a DIV may comprise several paragraphs.
  - This is similar to the "paragraph style" in MS-Word.
  - It serves to group elements sharing the same styles.
  - Allows for the use of events and some special attributes such as z-index or visibility.

```
layer 1 above layer 2:
<div style="position:relative; font-size:
50px; z-index:2;">CAPA 1</div>


<div style="position:relative; top:-50; left:
5; color:red; font-size:80px; z-index:
1">CAPA 2</div>
```



Capa 1 por encima de capa 2:

CAPA 1

CAPA 2

DIV and SPAN

- <DIV> attributes:
  - Visibility, defines whether the layer is visible or not
    - Visible;
    - Hidden;
  - Name, id="capa1"
  - Z-index, position of a layer compared to others. Depends on the assigned number.
  - Overflow: defines the behavior of the browser when a layer's content is larger than the layer.
    - Visible
    - Hidden
    - Scroll

DIV and SPAN

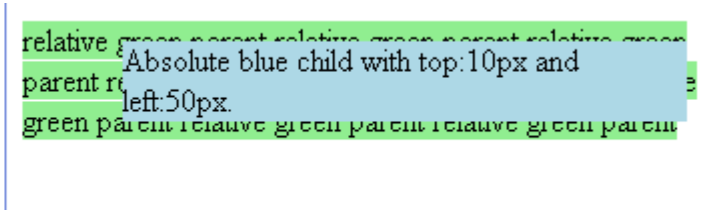
- <DIV> attributes:
  - Positioning:

Attribute	Values	Example
position	absolute, relative, static	position:relative;
top	%, pixels	top:-50px;
left	%, pixels	left:20px;
right	%, pixels	right:-20px;
bottom	%, pixels	bottom:0px;
height	%, pixels	height:50%
width	%, pixels	width:50%

DIV and SPAN

- SPAN delimits a group of characters
  - This is similar to formats like <b> </b>
  - In MS-word it corresponds to the "character style".
  - Allows for superposition of layers

```
<SPAN STYLE="position:relative; background-
color:#90EE90"> relative green parent relative
elative green parent relative green parent
relative green parent

<SPAN STYLE="position:absolute; top:10px;
left:50px;background-color:#ADD8E6">
Absolute blue child con top:10px y left:50px. </
SPAN></SPAN>
```



relative green parent relative green parent relative green
parent re e
green parent relative green parent relative green parent

Absolute blue child with top:10px and
left:50px.

Example of DIV and SPAN

```
/* Definición de reglas */
DIV.Abstract { text-align: justify }
SPAN.programa {
   font-family: Lucida Console, monospace;
   font-size: 0.8em;
}
```

CSS

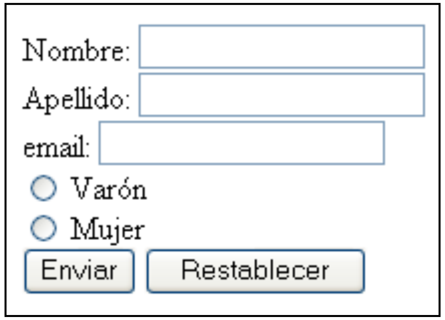
```
<DIV class="Abstract">
<p>Párrafoprinero…</P>
<p>Segundo párrafo…</p>
</DIV>

<h1>La variable <SPAN class="programa“>int</SPAN> en lenguaje C</h1>
```

HTML

Forms

- An HTML form is a special section of a document which gathers the usual content plus codes, special elements called controls (checkboxes, radiobuttons, menus), etc.

```
<FORM action="http://algunsitio.com/prog/usuarionuevo.php"
method="post">
Nombre:<INPUT type="text" id="nombre"><BR>
   Apellido:<INPUT type="text" id="apellido"><BR>
   email:<INPUT type="text" id="email"><BR>
<INPUT type="radio" name="sexo" value="Varón"> Varón<BR>
<INPUT type="radio" name="sexo" value="Mujer"> Mujer<BR>
<INPUT type="submit" value="Enviar"><INPUT type="reset">
</FORM>
```



Forms

- The *form* element is something like a container for all the controls inside a web page.
- Attributes:
    - Action: refers to the program which will process the content of the form once sent.
    - Name: name of the form to be referred to. It’s optative, but it is highly recommended to include.
    - Method: methods of sending data to the server. (Get by URL, Post by standard input stdio)
    - Accept-charset: set of accepted characters.

Forms

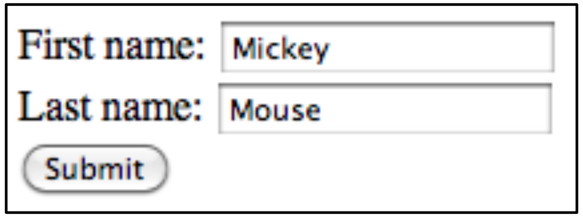
- Data sending methods:
  - get: with this method, the set of data is linked to the URL specified by the action attribute (using a question mark as a separator ("?") all this information is sent to the program in charge of processing it.
  - post: with this method, the set of data is included and sent to the program in charge of processing it by the standard input/output via.
  - The *post* method is more secure than *get*.

Forms

```
<html>
 <head>
   <title>PHP example</title>
 </head>
 <body>
 <H1>Ejemplo de procesado de formularios</H1>

 <FORM ACTION="procesa2.php" METHOD="GET">
 Introduzca su nombre:<INPUT TYPE="text" NAME="nombre"><BR>
 Introduzca sus apellidos:<INPUT TYPE="text" NAME="apellidos"><BR>
 <INPUT TYPE="submit" VALUE="Enviar">
 </FORM>
 </body>
 </html>
```

Forms

- Controls: text input



First name: Mickey

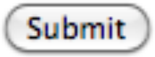
Last name: Mouse

- The <input> tag defines the input fields in the form.
- Attributes:
  - type="‶ type of field.
  - maxlenght="" indicates the maximum number of characters.
  - size="" maximum number of characters displayed on screen.
  - value="" initial value on this field.
  - name="" name to be referenced.

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

Forms

- Controls, buttons



  - Defined by the <input> tag plus some attributes:
    - type, followed by *submit* to send the data, or followed by *reset* to erase the data and leave the fields blank, or *button* to perform some action.
    - value, to indicate the text inside a button (usually send or reset).
    - name, to identify a button.

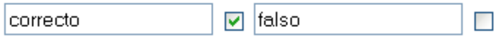
```
<input type="button" value="Cerrar esta ventana"
onclick="window.close();">
```

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

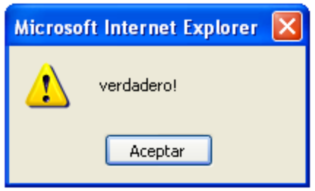
Forms

- Controls, checkbox
  - To select one or some options from several choices. (main difference with radiobutton).
  - One option may be preselected (checked) when the page loads.

```
<INPUT TYPE="label" value="correcto">
<INPUT TYPE="checkbox" name="c1" value= "1"

onClick="if(this.checked == true){alert('verdadero!'); this.checked=false;}">
<INPUT TYPE="label" value="falso">
<INPUT TYPE="checkbox" name="c1" value= "0"

onClick="if(this.checked == true){alert('falso!'); this.checked=false;}">
```



correcto ☑ falso □



Microsoft Internet Explorer

verdadero!

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

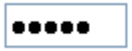
Forms

- Controls, special inputs

– There are some special inputs like:

- Password: this field won’t display the characters while typing. It will display asterisks instead.
- Hidden: the value of this field can’t be modifyed because the user can’t see this field. It usually has a fixed value defined by the value attribute.
- Their attributes are the same as for text.

```
<input type="password" name="clave" maxlength="5" size="6">
```

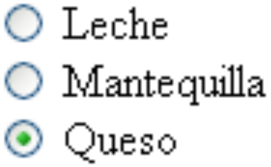


http://www.w3schools.com/tags/tag_input.asp

Forms

- Controls, radio
  - Similar to checkbox, to select an option from some choices.
  - Only one active field allowed.

```
<input type="radio" name="g1" value="Leche"> Leche<br>
<input type="radio" name="g1" value="Mant" checked> Mantequilla<br>
<input type="radio" name="g1" value="Queso"> Queso
```

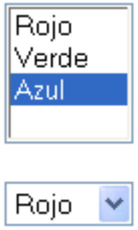


○ Leche
○ Mantequilla
● Queso

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

Forms

- Controls, select lists
  - The tags <select>.....</select> hold the values to be selected from a list of choices. The attributes of the opening tag are:
  - name="" identifies the selection tag.
  - Size="" indicates the number of visible options. If it is 1, the selection will be a menu. If the value is greater than 1, the list will have a scroll bar.
  - Multiple: indicates multiple selection (the user can select more than one option).

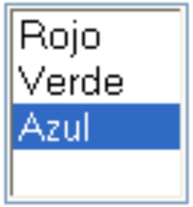


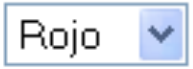
http://www.w3schools.com/tags/tag_select.asp

Forms

- Controls, select lists
  - The <option> tag indicates the different options.
  - The *selected* attribute, indicates the default option.
  - If it is not specified, it will be the first element on the list.

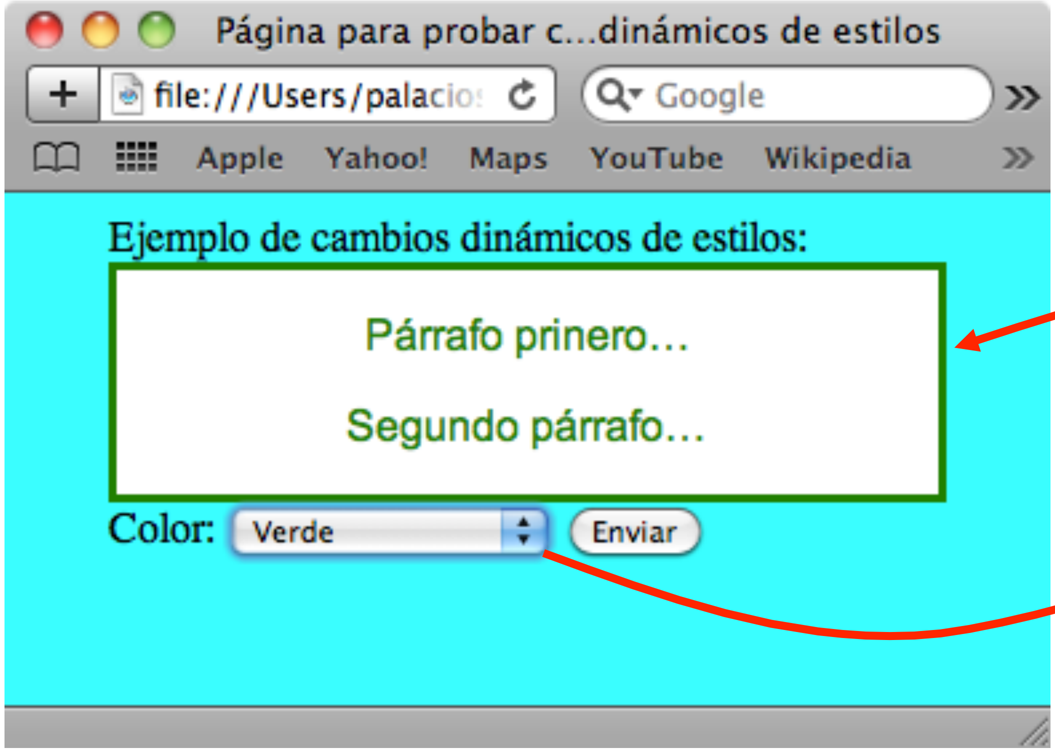
```
<SELECT NAME="Colores" MULTIPLE>
<OPTION VALUE="r">Rojo</OPTION>
<OPTION VALUE="g">Verde</OPTION>
<OPTION VALUE="b">Azul</OPTION>
</SELECT>
<BR><BR>
<SELECT NAME="Colores" SIZE="1">
<OPTION VALUE="r">Rojo</OPTION>
<OPTION VALUE="g">Verde</OPTION>
<OPTION VALUE="b">Azul</OPTION>
</SELECT>
```





Dynamic Styles

- A web site can behave as an application if CSS styles are linked to forms and JavaScript functions.



Ejemplo de cambios dinámicos de estilos:

Párrafo prinero…

Segundo párrafo…

Color: Verde

Enviar

Changes the style

OnChange

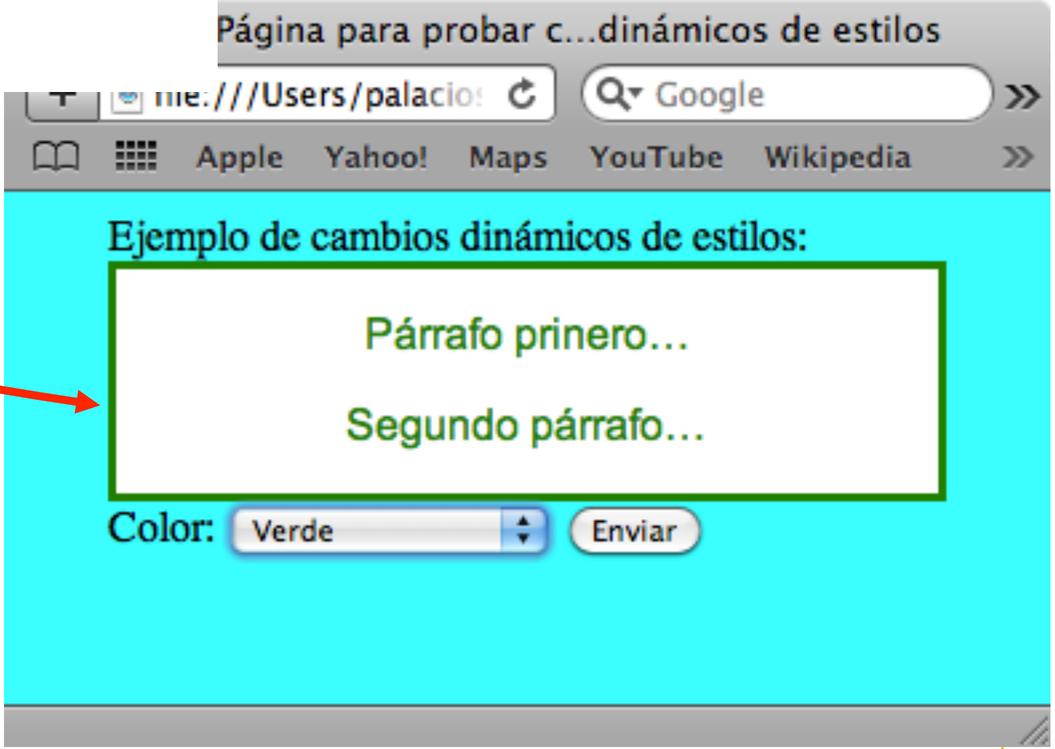
Dynamic Styles

- div object with style type “cuadro” named “id_cuadro1”

```
<DIV class="cuadro" id="id_cuadro1">
<p>Párrafoprinero…</P>
<p>Segundo párrafo…</p>
</DIV>
```

mis_estilos.css

```
div.cuadro {
   font-family:arial,helvetica,sans-serif;
color:black;
   background: white;
   text-align:center;
   border-style:solid;
}
```



Página para probar c…dinámicos de estilos

Ejemplo de cambios dinámicos de estilos:

Párrafo prinero…

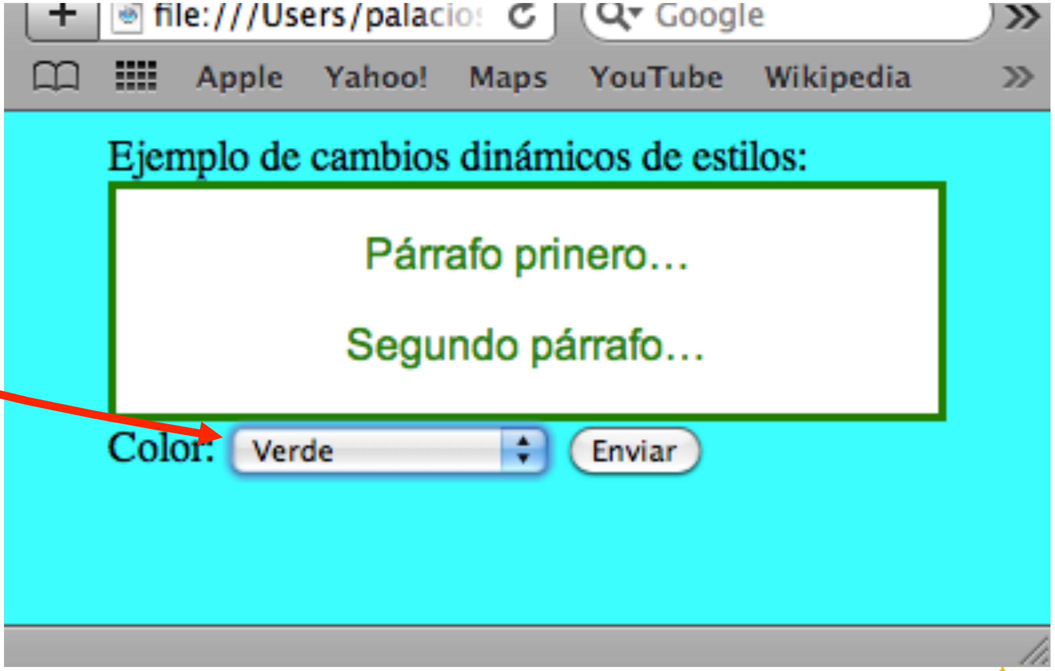
Segundo párrafo…

Color: Verde

Dynamic Styles

- <select> object to call a JavaScript for events type “OnChange”

```
<SELECT NAME="colores" OnChange="CambiarColor(this.value,'id_cuadro1');">
<OPTION VALUE="">--Elegir color--</OPTION>
<OPTION VALUE="r">Rojo</OPTION>
<OPTION VALUE="g">Verde</OPTION>
<OPTION VALUE="b">Azul</OPTION>
</SELECT>
```



Ejemplo de cambios dinámicos de estilos:

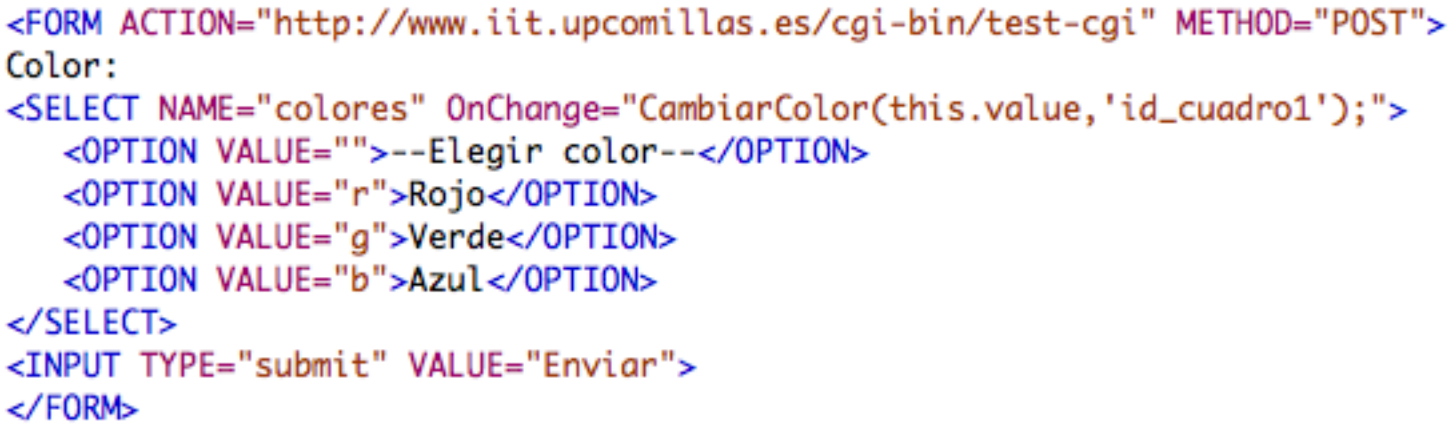
Párrafo prinero…

Segundo párrafo…

Color: Verde

Dynamic Styles

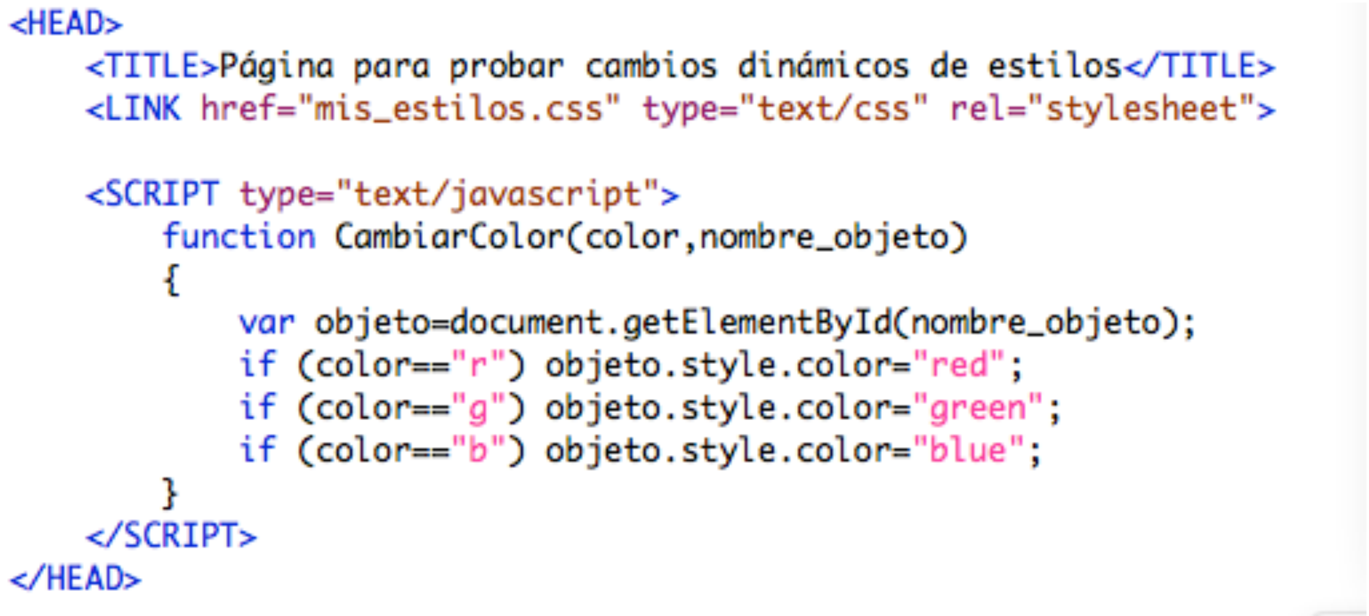
- Form code



```
<FORM ACTION="http://www.iit.upcomillas.es/cgi-bin/test-cgi" METHOD="POST">
Color:
<SELECT NAME="colores" OnChange="CambiarColor(this.value,'id_cuadro1');">
   <OPTION VALUE="">--Elegir color--</OPTION>
   <OPTION VALUE="r">Rojo</OPTION>
   <OPTION VALUE="g">Verde</OPTION>
   <OPTION VALUE="b">Azul</OPTION>
</SELECT>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
```

Dynamic Styles

- JavaScript function defined on <HEAD>



```
<HEAD>
    <TITLE>Página para probar cambios dinámicos de estilos</TITLE>
    <LINK href="mis_estilos.css" type="text/css" rel="stylesheet">

    <SCRIPT type="text/javascript">
        function CambiarColor(color,nombre_objeto)
        {
            var objeto=document.getElementById(nombre_objeto);
            if (color=="r") objeto.style.color="red";
            if (color=="g") objeto.style.color="green";
            if (color=="b") objeto.style.color="blue";
        }
    </SCRIPT>
</HEAD>
```