

CSS

Selector universal

Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML (por ahora no es importante fijarse en la parte de la declaración de la regla CSS):

```
* { margin: 0;  
    padding: 0;  
}
```

El selector universal se indica mediante un asterisco (*).

A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

Selector de tipo o etiqueta

- Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:
- ```
p {
 ...
}
```
- Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML

# Selector descendente

- Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.
- El selector del siguiente ejemplo selecciona todos los elementos `<span>` de la página que se encuentren dentro de un elemento `<p>`:
  - `p span { color: red; }`

# Ejemplo

- Con el CSS: `p span { color: red; }`
- Si el código HTML de la página es el siguiente:  

```
<p> ... texto1 ...
 ...texto2 ...
</p>
...
```
- El selector `p span` selecciona tanto `texto1` como `texto2`. El motivo es que en el selector descendente, un elemento no tiene que ser descendiente directo del otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar el nivel de profundidad en el que se encuentre.

# Selector de clase

En el HTML:

```
<body>
```

```
<p class="destacado">Lorem ipsum dolor sit amet...</p>
```

```
<p>Nunc sed lacus et est adipiscing accumsan...</p>
```

```
<p>Class aptent taciti sociosqu ad litora...</p>
```

```
</body>
```

En el CSS:

```
.destacado { color: red; }
```

El selector `.destacado` se interpreta como *"cualquier elemento de la página cuyo atributo class sea igual a destacado"*, por lo que solamente el primer párrafo cumple esa condición.

# Selector de clase II

En ocasiones, es necesario restringir el alcance del selector de clase. Si se considera de nuevo el ejemplo anterior:

```
<body>
```

```
<p class="destacado">Lorem ipsum dolor sit amet...</p>
```

```
<p>Nunc sed lacus et est adipiscing
accumsan...</p>
```

```
<p>Class aptent taciti <em class="destacado">sociosqu ad
litora...</p>
```

```
</body>
```

¿Cómo es posible aplicar estilos solamente al párrafo cuyo atributo class sea igual a destacado? Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:

```
p.destacado { color: red }
```

El selector p.destacado se interpreta como *"aquellos elementos de tipo <p> que dispongan de un atributo class con valor destacado"*.

# Selectores de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id.

La sintaxis de los selectores de ID se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#destacado { color: red; }
```

```
<p>Primer párrafo</p>
```

```
<p id="destacado">Segundo párrafo</p>
```

```
<p>Tercer párrafo</p>
```



# Selector de hijos

Se trata de un selector similar al selector descendente, pero muy diferente en su funcionamiento. Se utiliza para seleccionar un elemento que es *hijo directo* de otro elemento y se indica mediante el "*signo de mayor que*" (>):

```
p > span { color: Blue;}
```

```
<p>Texto1</p>
```

```
<p>Texto2</p>
```

# Selector adyacente

El selector adyacente se emplea para seleccionar elementos que en el código HTML de la página se encuentran justo a continuación de otros elementos. Su sintaxis emplea el signo + para separar los dos elementos:

`elemento1 + elemento2 { ... }`

# Ejemplo

```
<body> <h1>Titulo1</h1>
 <h2>Subtítulo</h2>
 ...
 <h2>Otro subtítulo</h2>
 ...
</body>
```

La página anterior dispone de dos elementos <h2>, pero sólo uno de ellos se encuentra inmediatamente después del elemento <h1>. Si se quiere aplicar diferentes colores en función de esta circunstancia, el selector adyacente es el más adecuado:

```
h2 { color: green; }
h1 + h2 { color: red }
```

Las reglas CSS anteriores hacen que todos los <h2> de la página se vean de color verde, salvo aquellos <h2> que se encuentran inmediatamente después de cualquier elemento <h1> y que se muestran de color rojo.

# Agrupación de reglas

- Cuando se crean archivos CSS complejos con decenas o cientos de reglas, es habitual que los estilos que se aplican a un mismo selector se definan en diferentes reglas:

```
h1 { color: red; }
```

```
...
```

```
h1 { font-size: 2em; }
```

```
...
```

```
h1 { font-family: Verdana; }
```

En uno solo:

```
h1 { color: red;
 font-size: 2em;
 font-family: Verdana; }
```

# Herencia

- Una de las características principales de CSS es la herencia de los estilos definidos para los elementos. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
 body { color: blue; }
</style>
</head>

<body>
 <h1>Titular de la página</h1>
 <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

- En el ejemplo anterior, el selector `body` solamente establece el color de la letra para el elemento `<body>`.
- No obstante, la propiedad `color` es una de las que se heredan de forma automática, por lo que todos los elementos descendientes de `<body>` muestran ese mismo color de letra. Por tanto, establecer el color de la letra en el elemento `<body>` de la página implica cambiar el color de letra de todos los elementos de la página.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
 body { font-family: Arial; color: black; }
 h1 { font-family: Verdana; }
 p { color: red; }
</style>
</head>

<body>
 <h1>Titular de la página</h1>
 <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```



# Colisiones de estilos

- En las hojas de estilos complejas, es habitual que varias reglas CSS se apliquen a un mismo elemento HTML.

```
p { color: red; }
```

```
p { color: blue; }
```

- Hasta que no se expliquen más adelante los conceptos de tipo de hoja de estilo y la prioridad, el mecanismo simplificado que se puede aplicar es el siguiente:
  - Cuanto más específico sea un selector, más importancia tiene su regla asociada.
  - A igual *especificidad*, se considera la última regla indicada.
- Como en el ejemplo anterior los dos selectores son idénticos, las dos reglas tienen la misma prioridad y prevalece la que se indicó en último lugar, por lo que el párrafo se muestra de color azul.

# Unidades

- Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto. Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida).

# Unidades absolutas

- `in`, pulgadas ("*inches*", en inglés). Una pulgada equivale a 2.54 centímetros.
- `cm`, centímetros.
- `mm`, milímetros.
- `pt`, puntos. Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros.
- `pc`, picas. Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros.

# Ejemplo

```
/* El cuerpo de la página debe mostrar un margen de media pulgada */
```

```
body { margin: 0.5in; }
```

```
/* Los elementos <h1> deben mostrar un interlineado de 2 centímetros */
```

```
h1 { line-height: 2cm; }
```

```
/* Las palabras de todos los párrafos deben estar separadas 4 milímetros entre sí */
```

```
p { word-spacing: 4mm; }
```

```
/* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos */
```

```
a { font-size: 12pt }
```

```
/* Los elementos deben tener un tamaño de letra de 1 pica */
```

```
span { font-size: 1pc }
```

# Unidades relativas

- em, (no confundir con la etiqueta <em> de HTML) relativa respecto del tamaño de letra del elemento.
- ex, relativa respecto de la altura de la letra x ("*equis minúscula*") del tipo y tamaño de letra del elemento.
- px, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

# Porcentajes

- El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a em, ex y px. Un porcentaje está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida.

```
div#contenido { width: 600px; }
```

```
div.principal { width: 80%; }
```

# Colores

- CSS define 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:
  - aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow

# RGB decimal

- En CSS, las componentes de los colores definidos mediante RGB decimal pueden tomar valores entre 0 y 255. El siguiente ejemplo establece el color del texto de un párrafo:
  - `p { color: rgb(71, 98, 176); }`



# RGB porcentual

- Por tanto, para transformar un valor RGB decimal en un valor RGB porcentual, es preciso realizar una regla de tres considerando que 0 es igual a 0% y 255 es igual a 100%.
- El mismo color del ejemplo anterior se puede representar de forma porcentual:
  - p { color: rgb(27%, 38%, 69%); }

# RGB hexadecimal

- Siguiendo el mismo ejemplo de las secciones anteriores, el color del párrafo se indica de la siguiente forma utilizando el formato RGB hexadecimal:
- `p { color: #4762B0; }`

El formato RGB hexadecimal es la forma más compacta de indicar un color, ya que incluso es posible comprimir sus valores cuando todas sus componentes son iguales dos a dos:

`#AAA = #AAAAAA`

`#FFF = #FFFFFF`

`#A0F = #AA00FF`

`#369 = #336699`