

04-02 CSS - Selectores de CSS

Nombre		Curso	
Apellidos		Fecha	

El selector es la parte de la regla de estilo que identifica el elemento concreto al que se aplicarán las instrucciones de presentación. CSS ofrece varios tipos de selectores que permiten mejorar la flexibilidad y la eficiencia en la creación de hojas de estilo.



la sintaxis de las reglas de estilo simple

Regla simple CSS:

```

selector {
    regla1: valor; regla2: valor; ...
}
  
```

Sintaxis de una reglas de estilo completa

Regla avanzada CSS:

```

Etiqueta.Clase#Identificador:Pseudo (espacio>+,) SelectorHijo2
{
    regla1: valor !important ;
    regla2: valor; ...
}
  
```

2.1.- Selectores de elemento.

Los selectores de elemento son los más sencillos. Son aquellos que se aplican a un elemento (etiqueta) del lenguaje (X)HTML.

```
h1 {color: blue;}  
h2 {color: blue;}  
p {color: blue;}
```

Si te fijas en los ejemplos, verás que se está definiendo la misma propiedad (**color**) en todos los elementos e incluso se está asignando el mismo valor (blue). El ejemplo siguiente muestra como se puede escribir una única regla aplicada a varios selectores a la vez.

```
h1, h2, p {color: blue;}
```

Cuando se realiza una declaración sobre varios selectores a la vez, éstos se separan por comas.

Selector Universal *

Existe un selector de elementos "universal" representado por el asterisco (*). El ejemplo siguiente muestra una regla que pondrá en gris todos los elementos del documento que no tengan especificado otro color.

```
* {color: grey;}
```

2.2.- Selectores contextuales. (espacio > + ,)

Como vimos, los **selectores de elemento** se aplican a todos los casos en los que se encuentre el elemento en el documento (X)HTML. En cambio, los selectores contextuales permiten aplicar estilos a los elementos basándose en su contexto o en su relación con otro elemento.

Hay varios tipos de selectores contextuales: **descendente, hijo y hermano**.

Los selectores descendentes hacen referencia a elementos que están contenidos en otro elemento.

Un selector descendente se pone a continuación del selector en el que está contenido separado de él por un espacio en blanco.

Ejemplos

El siguiente ejemplo especifica que los elementos **em** deben tener color azul, pero sólo si son descendientes de un elemento de lista (**li**). El resto de los elementos **em** no se verán afectados.

```
li em {color: blue;}
```

Los selectores descendientes también pueden estar anidados en varias capas de profundidad. El siguiente **ejemplo** pone de color amarillo sólo el texto enfatizado (**em**) de las anclas (**a**) que se encuentren en las listas ordenadas (**ol**).

```
ol a em {color: yellow;}
```

Si se emplea el selector descendente combinado con el selector universal, se puede restringir el alcance de un selector descendente. El siguiente ejemplo muestra dos párrafos que contienen un hipervínculo.

En el primer caso el elemento ancla es descendiente directo del elemento de párrafo y, en el segundo caso, es descendiente directo del elemento **span** que, a su vez, lo es del elemento de párrafo.

```
<p><a href="#">Enlace</a></p>  
<p><span><a href="#">Enlace</a></span></p>
```

Examinemos las dos reglas de estilo siguientes:

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

Con la primera se consigue que se muestren los dos enlaces de color rojo, mientras que con la segunda regla sólo se mostraría en rojo el segundo de los enlaces. La razón es que el selector `p * a` se traduce como "todos los elementos de tipo `<a>` que se encuentran dentro de cualquier elemento que, a su vez, se encuentra dentro de un elemento de tipo `<p>`". Como el primer elemento `<a>` está directamente incluido en un elemento `<p>`, no se cumple la condición del selector `p * a`.

Un selector hijo (`>`) es un caso concreto de un selector descendente en el que el selector está contenido directamente en otro, sin que existan niveles intermedios. Un selector hijo se escribe a continuación de su selector padre separándolo de él por el símbolo "mayor que" (`>`). En el siguiente ejemplo se pone en gris el fondo del texto enfatizado, pero sólo si es hijo directo de un párrafo.

```
p > em {background-color: gray;}
```

El selector adyacente (`+`) tambien llamados hermanos se utiliza para hacer referencia a un elemento que sigue inmediatamente a otro en el código, con el que comparte el mismo elemento padre.

Un selector adyacente se escribe a continuación de otro selector separándolo de él por el símbolo de suma (`+`). El siguiente ejemplo pondría en color azul el primer párrafo que sigue a una cabecera de primer nivel.

```
h1 + p {color: blue;}
```

Tenemos el siguiente documento XHTML con una hoja de estilo incrustada:

```
<html>  
  <head>  
    <style type="text/css">  
      h2 + p { color: green; }  
    </style>  
    <title>Una prueba de selectores adyacentes</title>  
  </head>  
  <body>  
    <h1>Título 1</h1>  
    <h2>Subtítulo 1</h2>  
    <p>párrafo 1 </p>  
    <h2>Subtítulo 2</h2>  
    <p>párrafo 2</p>  
  </body>  
</html>
```

Al abrir este documento en un navegador ¿qué texto aparecerá en color verde?

Título 1 Subtítulo 1 párrafo 1 Subtítulo 2 párrafo 2 Subtítulo 1 y Subtítulo 2 **párrafo 1 y párrafo 2**

2.3.- Selectores de clase e ID.

Para poder hacer uso de selectores más específicos, se hace necesario introducir los conceptos de identificador (`id`) y clase (`class`).

Identificadores (`id`) (#)

Los elementos HTML disponen de un atributo llamado identificador (`id`), que tiene como finalidad identificar al de manera excluyente. De este modo, CSS u otro lenguaje podrá hacer referencia a él y distinguirlo del resto de los elementos del documento.

Un id debe ser único en cada documento HTML.

```
<p id="textocabecera">
```



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

Para escribir una regla de estilo que se aplique a un determinado identificador hay que escribir el símbolo de la almohadilla (#) seguido del nombre del identificador.

El ejemplo siguiente muestra algunas formas de establecer el tamaño de la fuente en 14 píxeles al elemento p identificado como "textocabecera" del ejemplo anterior pero ambas son lo mismo ya que **Un id debe ser único en cada documento HTML.**

```
p#textocabecera {font-size: 14px;}  
#textocabecera {font-size: 14px;}
```

Con la primera regla indicamos que se aplique el estilo a un párrafo cuyo identificador sea "textocabecera" pero, como el id es único en cada documento, realmente basta con la segunda forma para decir lo mismo, porque no va a haber otro elemento <p> o diferente de <p> que tenga ese mismo identificador.

Si tenemos varios elementos que necesitan un tratamiento similar, emplearemos el atributo **class**.

Clases (class) (.)

Se emplea el atributo **class** para identificar distintos elementos como parte de un grupo conceptual. Así, los elementos de una clase pueden modificarse con una única regla de estilo.

En el siguiente ejemplo se muestra como dos elementos distintos se clasifican de la misma forma mediante la asignación del valor "especial" al atributo **class**.

```
<h1 class="especial">jAtención!</h1>  
<p class="especial">Hoy tenemos grandes rebajas.</p>
```

Un elemento puede pertenecer a más de una clase separando sus nombres de clase con espacios.

El siguiente ejemplo el párrafo pertenece a dos clases: "textocabecera" y "especial".

```
<p class="textocabecera especial">Hoy tenemos grandes rebajas.</p>
```

Para escribir una regla de estilo que se aplique a todos los elementos de una determinada clase hay que escribir un punto seguido del nombre de la clase.

Por ejemplo: **.especial {color: green;}**

El siguiente ejemplo muestra la forma de lograr que todos los elementos de la clase "especial" tengan un color verde a excepción de las cabeceras de primer nivel que tienen que ser rojas.

```
.especial {color: green;}  
h1.especial{color: red;}
```

Los nombres de clases y de identificadores no pueden contener espacios en blanco.

2.4.- Pseudoselectores.

Si queremos aplicar reglas de estilo a elementos especiales como: los vínculos visitados, la primera línea de un párrafo o su primera letra, emplearemos los **pseudoselectores**.

Hay dos tipos de pseudoselectores: **pseudoclases** y **pseudoelementos**.

2.4.1 Pseudoclases link visited hover active.

Clasifican a los elementos basándose en características que van más allá de su nombre, atributos o contenido. La mayoría de las pseudoclases afectan a los elementos ancla (), pudiendo definir un estilo diferente en función del comportamiento del enlace: si todavía no ha sido visitado (`link`), si ya lo ha sido (`visited`), mientras el ratón pasa por encima (`hover`) o justo cuando se pulsa el ratón sobre él (`active`).

Para emplear una pseudoclase se escribe la misma a continuación del selector separándola de éste por el símbolo de dos puntos (:). El ejemplo siguiente muestra cómo se distinguirían los cuatro estados posibles de un enlace mediante colores diferentes.

```
a:link {color: red;}  
a:visited {color: blue;}  
a:hover {color: fuchsia;}  
a:active {color: maroon;}
```

Con el empleo de estas pseudoclases podemos quitar el subrayado de los hiperenlaces y hacer que aparezca sólo cuando pasamos el puntero por encima. El ejemplo siguiente muestra el empleo de la propiedad `text-decoration` para conseguir dicho objetivo.

```
a:link {color: red; text-decoration: none;}  
a:visited {color: blue; text-decoration: none;}  
a:hover {color: red; text-decoration: underline;}
```

Debes tener en cuenta que las pseudoclases ancla deben aparecer siempre en un determinado orden. Este orden es: `:link`, `:visited`, `:hover` y `:active`. Por si te ayuda, para recordarlo, se emplean las iniciales: **LVHA**.

Existen otras pseudoclases, que se emplean menos:

- ✓ `:focus` hace referencia a los elementos que tienen el foco, como ocurre, por ejemplo, en los elementos de un formulario. Un ejemplo sería: `input:focus {background-color: yellow;}`
- ✓ `:first-child` hace referencia al primer hijo de un elemento padre. En el siguiente ejemplo se aplica el estilo al primer elemento de una lista desordenada: `ul li:first-child {font-weight: bold;}`
- ✓ `:first-child` hace referencia al primer hijo de un elemento padre. En el siguiente ejemplo se aplica el estilo al primer elemento de una lista desordenada: `ul li:first-child {font-weight: bold;}`
- ✓ `:nth-child(4n)` hace referencia al n hijo de un elemento padre. En el siguiente ejemplo se aplica el estilo al 4º elemento de una lista desordenada: `ul li:nth-child(4n) {font-weight: bold;}`
- ✓ `:lang(idioma)` hace referencia al idioma en el que está un determinado elemento. En el siguiente *ejemplo* se aplica el estilo a cualquier párrafo que esté escrito en inglés: `p:lang(en) {color: red}`

2.4.1 Pseudoelementos. `:first-line` `:first-letter` `:after` `:before`

Estos pseudoelementos suelen ser partes de un elemento ya existente, como puede ser su primera línea (`:first-line`) o su primera letra (`:first-letter`), aunque también nos permite hacer referencia a elementos sin concretar en la estructura del documento porque dependen de la estructura del documento (`:before` y `:after`).

```
p:first-line {letter-spacing: 6pt;}  
p.definicion:first-letter {font-size: 300%; color: red;}  
p.incompleto:after { content: " continuará ..."; }
```

En el primer ejemplo añadimos espacio extra a la primera línea del texto de cada párrafo, en el segundo modificamos el estilo (tamaño y color) de la primera letra de los párrafos pertenecientes a la clase "definición" y, en el tercero, añadimos el texto "continuará ..." al final de cada párrafo perteneciente a la clase "incompleto".

2.5. La palabra clave `!important` en CSS.

`!important` se utiliza para dar prioridad a ciertas reglas. Cualquier definición de estilo que vaya acompañada de un `!important` tendrá prioridad sobre cualquier otra.

Como ya sabes, cuando tenemos una propiedad aplicada dos veces, el navegador hará caso a la última.



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

En el ejemplo que se muestra a continuación está claro que se le asignarán 1200 píxeles al ancho del elemento identificado con el `id="principal"`.

```
#principal { width: 800px; width: 1200px; }
```

Sin embargo, en el siguiente ejemplo se muestra como dando prioridad a la primera declaración con la palabra `!important`, podemos tener ese elemento con un ancho de 800 píxeles.

```
#principal { width: 800px !important; width: 1200px; }
```

Las declaraciones acompañadas de la palabra `!important` tienen prioridad sobre otras declaraciones que afecten al mismo elemento.

Con esto de los selectores descendentes, hijos y adyacentes, se puede dar el caso de que, sin querer, estemos aplicando un estilo que no queremos a un elemento porque se ve indirectamente afectado por alguna de las reglas y no sabemos cuál. Por ejemplo: sale un párrafo en color rojo y no nos acordamos dónde lo configuramos porque realmente no tenemos ninguna regla que diga `p {color:red}`.

En ese caso, y para no descerebrarnos buscando dónde está el fallo, si sabemos que los párrafos de texto emplean la letra de color negro, crearíamos la regla `p { color: green !important; }` y solucionado!

¿Cuál de los siguientes es el selector CSS que selecciona a todos los objetos con class="nuevo" que están contenidos en un tabla?

table .nuevo

table.nuevo

table#nuevo

Practica

Para cada ejercicio debes aplicarlo en un elemento que se vea con formato

<h2> Ejercicio i</h2>

<p> ejercicio ... </p>

Ejercicio 1. Rellena esta tabla y creala en el documento html

Regla avanzada CSS:

```
_____ # _____ . _____ : _____ ( _____ ) _____  
 {  
     _____ : _____ ! _____ ;  
     regla2: valor; ...  
 }
```

Ejercicio 2. Crea una regla CSS para que todos los elementos sean de color gris (selector *)

Ejercicio 3. Crea una regla para que los elementos que estén dentro de un p sean de color verde y fuente Georgia. (con descendientes)

Ejercicio 4. Crea una regla para que todos los elementos que estén dentro de una tabla tengan bordes rojos y amarillos (con descendientes)

Ejercicio 5. Crea una regla para que todos los elementos que estén inmediatamente dentro de un p sean de color azul y fuente Arial. (con hijo >)

Ejercicio 6. Crea una regla para que todos elementos **strong** deben tener color azul, pero sólo si son descendientes de un elemento de lista (ii). El resto de los elementos **strong** no se verán afectados.

Ejercicio 7. Crea una regla para que todos enlaces dentro de un div y de una etiqueta debe tener color magenta (con descendiente + selector *)

<div> hola </div> --> no magenta

<div> <p> hola </p> </div> --> SI magenta

Ejercicio 8. Crea una regla para que todos enlaces que se hallan empezando un div no tengan subrayado (con descendiente hijo + selector *) **div>a:first-child { text-decoration:none}**

<div> NO </div> --> no subrayado

<div> <p> hola</p> SI </div> --> SI subrayado

Ejercicio 9. Crea una regla para que todos strong que se *hallan justo dentro* un h1 tengan subrayado (con descendiente hijo) **h1>strong { text-decoration:underline}**

<h1> <div> no es nada </div> </h1> --> NO subrayado

<h1> no es nada </h1> --> SI subrayado



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

Ejercicio 10. Crea una regla para que todos los p que van junto con un h3 sean rojos (con selectores adyacentes-hermanos)

Ejercicio 11. Crea una regla para que el elemento con el identificador ejercicio10 sean rojos (#ejercicio10)

Ejercicio 12. Crea una regla para que los elementos con la clase ejercicio11 sean Azul (.ejercicio11)

Ejercicio 13. Crea dos reglas para que los elementos con la clase ejercicio12 sean Azules excepto los h2 que deben ser verdes y los h3 que deben ser amarillos

Ejercicio 14. Crea una regla de pseudoclase para que los enlaces con clase ejercicio13 tengan estos colores

- Verde si todavía no ha sido visitado
- Verde Oliva si ha sido visitado
- Rojo mientras el ratón pasa por encima
- Azul justo cuando se pulsa el ratón sobre
- Escribe dentro del párrafo un memotécnico para el concepto Orden Pseudoclase = LVHA

Ejercicio 15. Crea dos regla para que los elementos con la clase ejercicio14

- Regla 1 : que todos los p sean Azules con la clave !important
- Regla 2 : que los elementos con la clase ejercicio14 de p sean verdes

Indica en un párrafo porque teniendo más prioridad la regla 2 se ven azules los que cumplen la 2.

Ejercicio 16. Crea dos regla para que los elementos con la clase ejercicio15

- Regla 1 : que todos los p sean Azules con la clave !important en un archivo externo
- Regla 2 : que todos los p sean Negros en css en linea

Indica en un párrafo porque teniendo más prioridad la regla 2 se ven azules los que cumplen la 2.

Ejercicio 17. Crea cuatro reglas para que los elementos p con la clase ejercicio16

- Regla 1 : que la su primera línea de todos los p sean Azules
- Regla 2 : que la su primera letra de todos los p sean rojas y tamaño 15
- Regla 3 : que los elementos que siguen al texto hola dentro de un p sean verde
- Regla 4 : que los elementos que siguen al texto adios dentro de un p sean Amarillo

Debes usar (:first-line) (:first-letter) (:before) (:after).

Ejercicio 18. Crea una regla con :focus hace referencia a los elementos que tienen el foco de un formulario se iluminen . Un ejemplo sería: input:focus {background-color: yellow;}. Crea un formulario con 5 input text para observarlo.

Ejercicio 19. Crea una regla con ✓ :first-child hace referencia el primer hijo de un elemento padre. En el siguiente ejemplo se aplica el estilo al primer elemento de una lista desordenada: ul li:first-child {font-weight: bold;}

Aplica el anterior estilo a una lista con 5 elementos

Ejercicio 20. ✓ :lang(idioma) hace referencia al idioma en el que está un determinado elemento. En el siguiente **ejemplo** se aplica el estilo a cualquier párrafo que esté escrito en inglés: p:lang(en) {color: red}