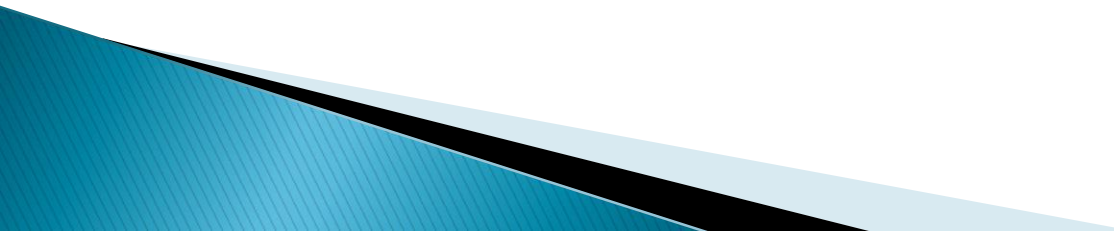


# CSS3

# Selectores

- ▶ Universal
  - ▶ De tipo o etiqueta
  - ▶ Descendentes(anidamiento)
  - ▶ De clase
  - ▶ De id
- 

# Selector Universal

Se utiliza para seleccionar todos los elementos de la página.

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.

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

El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML.



# Selector de tipo o etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector.

**Selector{ atributo : valor }**

El siguiente ejemplo aplica diferentes estilos a los titulares y a los párrafos de una página HTML:

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

# Selector de tipo o etiqueta

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, **se pueden encadenar los selectores**. En el siguiente ejemplo, los títulos de sección h1, h2 y h3 comparten los mismos estilos:

- ▶ 

```
h1 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
h2 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

- ▶ **Sería equivalente a:**

- ▶ 

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

# Selector de tipo o etiqueta

- ▶ Lo más óptimo para que el fichero CSS resultante ocupe lo mínimo posible consiste en agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos. Véase el siguiente ejemplo, donde se establece en primer lugar las propiedades comunes de los títulos de sección (color y tipo de letra) y a continuación el tamaño de letra de cada uno de ellos::
- ▶ 

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
h1 { font-size: 2em; }  
  
h2 { font-size: 1.5em; }  
  
h3 { font-size: 1.2em; }
```

# Selector de tipo o etiqueta

- ▶ ¿Dónde se debe poner esas declaraciones para que el navegador las lea y las interprete?
- ▶ En la cabecera `<head></head>`, entre etiquetas `<style></style>`.
- ▶ `<head>`
- ▶ `<title></title>`
- ▶ `<style>`
  - `h1 { color : blue; background-color : red }`
- ▶ `</style>`
- ▶ `</head>`

# Actividad Selector de etiquetas

- Crea un fichero Html con dos textos, uno entre etiquetas h1 y otro entre etiquetas h2. Las etiquetas h1 con un estilo color azul y las etiquetas h2 con un background rojo.

---

**Esto esta entre etiquetas H1**

**Esto esta entre etiquetas H2**





# Selector de tipo o etiqueta

- ▶ También pueden seleccionarse ciertos elementos, pero sólo cuando estén contenidos dentro de otros. Por ejemplo, si se quisiera aplicar un estilo a todos los items (`<li>`) de una lista desordenada (`<ul>`):
- ▶ 

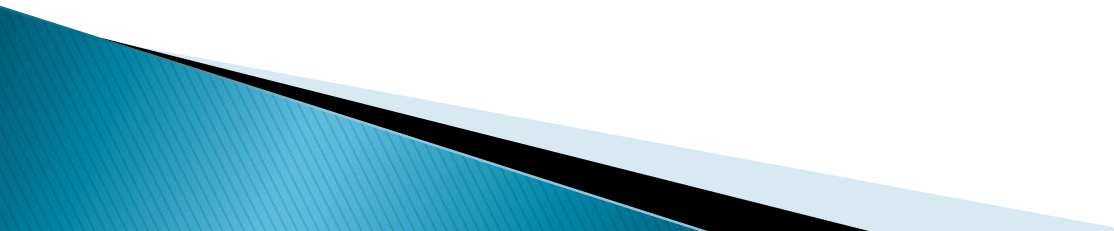
```
ul li {  
  ...  
}
```

# Selector descendente(anidamiento)

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.**

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

El selector del siguiente ejemplo selecciona todos los elementos `<span>` de la página que se encuentren dentro de un elemento `<p>`.



# Selector descendente(anidamiento)

Si el código HTML de la página es el siguiente:

```
<p> ...  
    <span>texto1</span> ...  
    <a href="">...<span>texto2</span></a> ... </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.**

Al resto de elementos `<span>` de la página que no están dentro de un elemento `<p>`, no se les aplica la regla CSS anterior.

# Selector descendente(anidamiento)

La sintaxis formal del selector descendente se muestra a continuación:

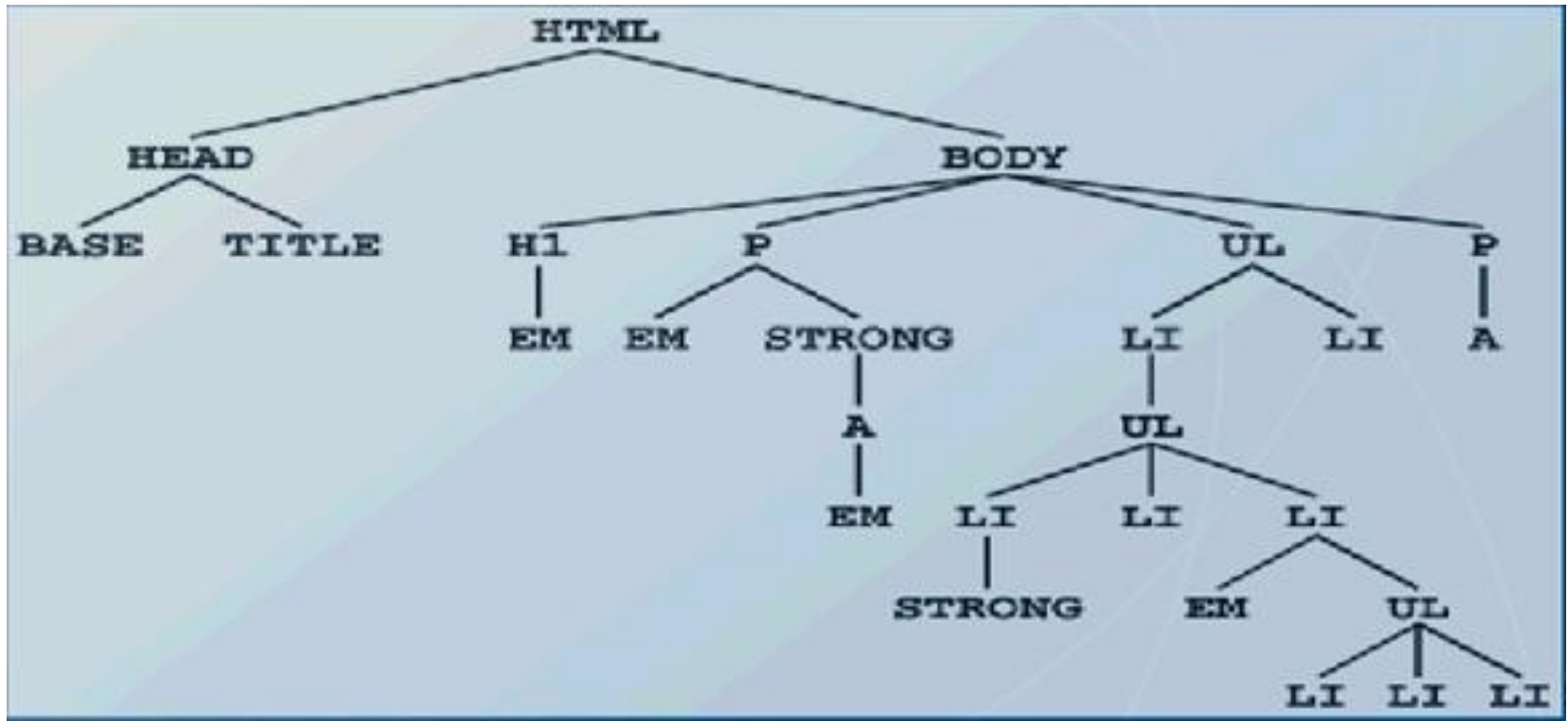
**selector1 selector2 selector3 ...  
selectorN{atributo:valor;atributo:valor...}**

En el siguiente ejemplo, el selector descendente se compone de cuatro selectores:

*p a span em { text-decoration: underline; }*

Los estilos de la regla anterior se aplican a los elementos de tipo `<em>` que se encuentren dentro de elementos de tipo `<span>`, que a su vez se encuentren dentro de elementos de tipo `<a>` que se encuentren dentro de elementos de tipo `<p>`.

# Selector descendente(anidamiento)



# Actividad selectores decendientes

Crea un fichero Html donde aparezca:

**Un texto h1 en negrita,  *cursiva y rojo*. Otro texto en negrita y rojo.**

# Selector descendente(anidamiento)

No hay límite en el número de agrupamiento que se pueden hacer, lo recomendable es no hacerlos de más de 4 ó 5.

Se pueden hacer con cualquier tipo de selector: etiqueta, class, id o combinaciones.

`h1 i b, h1 b { color: red }`

Equivale a:

`h1 i b { color: red }`

`h1 b {color : red }`

# Selector descendente(anidamiento)

Si lo que se desea es restringir que las etiquetas, además de estar en el mismo contexto, estén seguidas unas de otras, entonces la sintaxis es la siguiente (**selectores de hijos**):

**SelectorX>SelectorY{atributo1:valor1;  
atributo2:valor2...}**

En este ejemplo:

**h1 > b {color:red}**

Mostrará en rojo sólo el texto que tiene una etiqueta <b> dentro de <h1> **sin ninguna entre medias.**



# Selector descendente(anidamiento)

Cuando se quiere aplicar un estilo a un elemento que tiene adyacente ( **selectores hijos** ) a otro elemento en el mismo nivel de anidamiento en Html.

**SelectorX+SelectorY{propiedad1:valor1;propiedad2:valor2....}**

En este ejemplo:

```
<head>  
<style>  
  i+b {color:red}  
</style>
```

Muestra el anidamiento sobre dos etiquetas. Sólo se aplicará el estilo sobre <b> si hay una etiqueta adyacente <i>.

# Selector descendente(anidamiento)

Realiza un documento html donde aparezca el texto:

*Nota*, esto es una advertencia.

**Leer detenidamente.**



# Selector de clases

Los selectores basados en etiquetas tienen un uso muy claro, sin embargo no son flexibles ni reutilizables. Por ejemplo si se desea establecer diferentes estilos según el párrafo que se ponga, de manera que se distinga el encabezado del resto.

- ▶ Si se hace una declaración del tipo `p{color:blue}`, siempre que aparezca un texto entre etiquetas `<p></p>` se mostrará azul.
- ▶ Si se desea es que unos párrafos respondan a una regla de estilo y otros a otra, esta alternativa no es válida. Será necesario utilizar selectoras basados en clases.

# Selector de clases

La solución consistiría en utilizar el atributo **class** de HTML sobre ese elemento para indicar directamente la regla que se ha de aplicar:

- ▶ **NombreEtiqueta.nombreClase{atributo:valor}**
- ▶ Hay diferentes tipos de clases:
- ▶ Unas asociadas a una etiqueta Html (**h1.verde{color:green}.**)
- ▶ Otras más genéricas que se pueden aplicar a cualquier etiqueta (**.citas{color:green}**)

# Selector de clase

- ▶ Esta alternativa es mucho más potente ya que permite aplicar a las mismas etiquetas diferentes estilos:
- ▶ `<head>`
- ▶ `<style>`
- ▶ `h1.roja {color:red}`
- ▶ `h1.verde{color:green}`
- ▶ `h1.azul{color:blue}`
- ▶ `</style>`
- ▶ `</head>`
- ▶ Para indicar en cada etiqueta h1 que estilo se quiere aplicar se usa el atributo class:
- ▶ `<body>`
- ▶ `<h1 class="roja"> </h1>`
- ▶ `<h1 class="azul"> </h1>`
- ▶ `<h1 class="verde"> </h1>`
- ▶ `</body>`

# Selector de clase

- ▶ Las clases más genéricas no se aplican a ninguna etiqueta Html, por lo que en su descripción se emite en el selector el nombre de ninguna etiqueta.

- `.nombreClase{atributo:valor}`
- `.verde{color:green}`

Una clase así definida puede aplicarse a cualquier elemento de la página.

`<p class="verde">....` Se aplica a todo el párrafo.

`<h1 class="verde">....` A todo el encabezamiento h1.

`<div class="verde">....` A todo el bloque div.

Con la siguiente declaración:

```
<head>
```

```
<style>
```

```
  .roja {color:red}
```

```
  .verde{color:green}
```

```
  .azul{color:blue}
```

```
</style>
```

```
</head>
```

Y aplicando las clases a las diferentes etiquetas h1:

```
<body>
```

```
  <h1 class="roja">Un encabezamiento rojo</h1>
```

```
  <h1 class="azul">ahora azul</h1></h1>
```

```
  <h1 class="verde">y ahora verde</h1>
```

```
</body>
```

Pero no sólo a etiquetas h1 sino que a cualquier otra etiqueta, por ejemplo a `<p>`

```
<p class="roja">Aquí está el párrafo en rojo</p>
```

# Selector de clase

- ▶ Una característica muy interesante del uso de clases es que se puede asignar más de una clase a una misma etiqueta, siempre y cuando no haya conflicto entre ellas.
- ▶ `<head>`
- ▶ `<style>`
- ▶ `.textorojo {color:red}`
- ▶ `.fondoazul { background-color:blue }`
- ▶ `</style>`
- ▶ `</head>`
- ▶ Se podría hacer el siguiente contenido en el `<body>`
- ▶ `<h3 class="textorojo fondoazul">título en rojo, fondo azul</h3>`
- ▶ `<p class="textorojo">texto en rojo; fondo el que herede de la página </p>`

# Actividad selector de clases

- ▶ Crea un fichero Html con dos textos, uno entre etiquetas `<h1></h1>` y otro entre `<p></p>`. Luego incluye en el `<head>` estilos con clases abstractas que muestren:
- ▶ El texto `<h1>` en verde con fondo azul.
- ▶ El texto `<p>` en azul con fondo verde.
- ▶ Resuelve la actividad de dos maneras:
  - ▶ 1.– Usando una declaración de estilos con 4 etiquetas (2 para los colores y 2 para los fondos).
  - ▶ 2.– Usando dos etiquetas: una se puede llamar `.textoverdefondoazul` y la otra `.textoazulfondoverde`.



# Selectores de ID

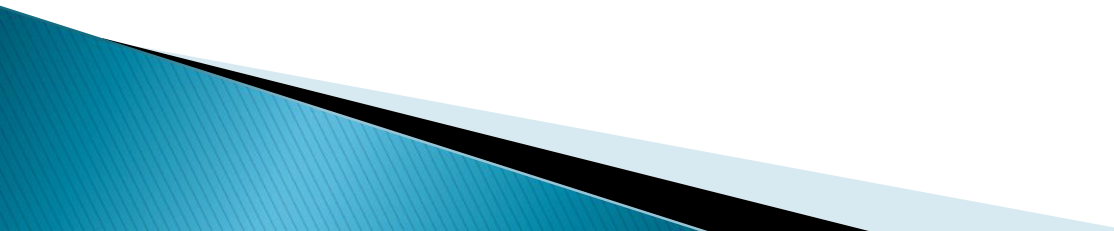
- ▶ En ocasiones, es necesario aplicar estilos CSS a **un único elemento de la página**. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.
- ▶ **El selector de ID** permite seleccionar un elemento de la página a través del valor de su atributo **id**. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo **id** no se puede repetir en dos elementos diferentes de una misma página.

# Selectores de ID

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

- ▶ **nombreEtiqueta.#nombreClase { atributo: valor;... }**
- ▶ `<head>`
- ▶ `<style>`  
`#rojo {color:red;}`
- ▶ `</style>`
- ▶ `</head>`
- ▶ `<body>`
- ▶ `<p id="rojo">el párrafo va en rojo</p>`
- ▶ `</body>`

# Selectores de ID

- ▶ La principal diferencia entre este tipo de selector y el selector de clase tiene que ver con HTML y no con CSS. Como se sabe, en una misma página, **el valor del atributo id debe ser único**, de forma que dos elementos diferentes no pueden tener el mismo valor de id. **Las clases** definen un patrón que deben de cumplir todos los objetos de la clase. Cada objeto se identifica con un identificador único que lo diferencia de los demás objetos de esa clase.
- 

# Selectores de ID

- ▶ Sin embargo, el atributo **class** no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo **class**.

# Selectores de ID

- ▶ De esta forma, la **recomendación general** es la de utilizar el **selector de ID** cuando se quiere aplicar un estilo a **un solo elemento** específico de la página y utilizar el **selector de clase** cuando se quiere aplicar un estilo a **varios elementos** diferentes de la página HTML.
- ▶ `<p id="nombreid" >      </p>`
- ▶ [Lista de elementos en HTML 5](#)

# Buenas prácticas al escribir CSS

Los selectores se nombran en minúsculas.

El nombre de los selectores debe ser específico y claro.

El nombre de las clases e identificadores no debe describir una característica visual, como color, tamaño o posición.

Los comentarios se ponen entre `/* Comentario */`

Separa las palabras mediante guiones o mayúsculas.

No hacer uso excesivo de las clases.

Agrupar las reglas según su selector.

Define al principio los selectores de etiquetas.

Estructura visualmente los atributos.

