

Tema 2: CSS3

PROGRAMACIÓN EN INTERNET

Pedro Delgado Pérez

Juan Boubeta Puig



Grado en Ingeniería Informática
Departamento de Ingeniería Informática

2023-2024

Contenido

- **Introducción**
- Aplicando estilos
- Buenas prácticas con CSS3
- Elementos fundamentales
- *Media queries* y responsividad
- Elementos extra
- Preprocesadores de CSS



Introducción

Definición

- **CSS3** (*Cascading Style Sheets*) es la versión 3 de las *hojas de estilo en cascada*. Una tecnología W3C que permite:
 - Especificar la **presentación** de los elementos de una página web (fuentes, tamaños, colores, posicionamiento...).
 - **Separar la presentación de la estructura y el contenido** del documento HTML (simplifica el mantenimiento de la página web).

“If you must return to your HTML to change the presentation (or styling) of the page, you’re doing it wrong”.

(Glen Stansberry en [code.tuts+](https://code.tutsplus.com/))

- Los estilos se aplican a las *etiquetas* (*div*, *ul*, *span*, *section*,...) o a etiquetas identificadas por *id* o *class*, entre otras opciones.

Introducción

¿Cómo añadir estilos?

Tres formas para añadir estilos:

1. *En línea*: `<div style="color: red;">Esto es un div</div>`
2. *Incrustado en el HTML*:
 - Se recomienda incluir justo antes de cerrar la etiqueta **head**.

```
<style type="text/css">
  div { color: red; }
</style>
```
3. *Archivo externo: (A USAR EN ESTA ASIGNATURA)*
 - Se añade antes de cerrar la etiqueta **head**.

```
<link href="estilos.css" rel="stylesheet" type="text/css">
```
 - En `estilos.css`:

```
/* Archivo externo de CSS3 con un comentario */
div { color: red; }
```

Nota: el atributo style tiene la mayor prioridad y el link la que menos.

Introducción

Estilos

Definición de estilos:

- Los estilos se definen entre llaves: `{ }`
- La aplicación de un estilo sigue la siguiente estructura.

div	{	color	:	red	;	...	}
Selector/es		Propiedad		Valor			
		Declaración				Resto de declaraciones	

- Cada etiqueta tiene un estilo por defecto, que puede variar según el navegador empleado.

Contenido

- Introducción
- **Aplicando estilos**
- Buenas prácticas con CSS3
- Elementos fundamentales
- *Media queries* y responsividad
- Elementos extra
- Preprocesadores de CSS

Aplicando estilos

Clases e identificadores (i)

Selectores:

Los estilos se pueden aplicar a un/a:

- *Etiqueta* concreta : **div**, **p**, **h1**...
- *Clase (o class)* de elementos. Se identifica mediante un **punto** (.):

```
<div class="bloqueDiv">Este es mi div con clase</div>
```

– CSS: **.bloqueDiv** { color:  **red**; }

- *Elemento con identificador (id)*. Se identifica con **almohadilla** (#):

```
<div id="divEjemplo">Este es mi div con id</div>
```

– CSS: **#divEjemplo** { color:  **blue**; }

- Y existen muchas otras opciones.

Una etiqueta puede tener varias clases, pero solo un identificador.

Aplicando estilos

Clases e identificadores (ii)

- Una *etiqueta* puede llevar **tanto *id* como *class***:

```
<div id="divEjemplo" class="bloqueDiv">  
    Este es mi div con clase y con id  
</div>
```

— En CSS: `.bloqueDiv { color:  red; }`
`#divEjemplo { color:  blue; font-weight:bold; }`

- **El selector *id* tiene prioridad** sobre el selector *class* y este sobre el de etiqueta.
- Por lo tanto, en el ejemplo de arriba:
 - Todos los textos de los *div* con esa clase se verán ... **Rojos**
 - Y, si a ese *div* se le añade el mencionado *id*, se verán... **Azules y en negrita.**

Aplicando estilos

Estilos en cascada

- ¿Qué significa la declaración del siguiente estilo CSS?

```
div ul {font-weight:bold;}
```

- Es la manera de anidar o aplicar estilos en cascada.
 - *A espacio E*: el elemento E es un descendiente de A
 - *A > E*: el elemento E es hijo directo de A

- Por tanto, ¿qué significa la declaración del siguiente estilo?

```
section>div>ul {font-weight:bold;}
```

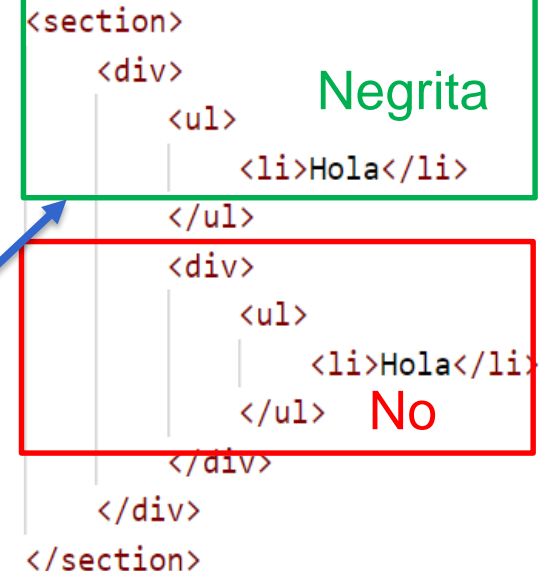
Coincidiría con toda sección, que tenga dentro un *div*, que contenga a su vez un *ul* (**todos de manera directa**).

```
<section>
  <div>
    <ul>
      <li>Hola</li>
    </ul>
  </div>
</section>
```

Negrita

```
<div>
  <ul>
    <li>Hola</li>
  </ul>
</div>
```

No



Contenido

- Introducción
- Aplicando estilos
- **Buenas prácticas con CSS3**
- Elementos fundamentales
- *Media queries* y responsividad
- Elementos extra
- Preprocesadores de CSS

Buenas prácticas con CSS3

Recomendaciones

Consultar ["Buenas prácticas para principiantes \(inglés\)"](#)

Conclusiones principales:

- Estructurar e **indentar bien** las hojas de estilos.
- Usar nombres de *class* e *id* **descriptivos**.
- **Lo primero: dar estilos a las etiquetas**. Después, usar en mayor medida *class* para así usar los mínimos *ids* posibles.
- En definitiva, ir **de lo más genérico a lo más específico**.

- Intentar aplicar estilos “*en cascada*” correctamente:

Si la regla queda muy larga o profunda:

```
div div ul { font-weight:bold; }
```

es aconsejable reducir asignando atributos *class* o *id*:

```
.menu-superior ul { font-weight:bold; }
```



Buenas prácticas con CSS3

Ejercicio 1




Consultar archivo: [bloques_anidados.html](#)

```
<div class=divCalendario">
  <div class="divHorasDiasMeses">
    <ul>
      <li>Horas</li>
      <li>Días</li>
      <ul>
        <li>Lunes</li>
        <li>Martes</li>
        <li>...</li>
      </ul>
      <li>Meses</li>
      <ul>
        <li>Enero</li>
        <li>Febrero</li>
        <li>...</li>
      </ul>
    </ul>
  </div>
</div>
```

1. ¿Cómo se aplicaría color de letra naranja solo a los primeros `` (Horas, Días y Meses)?
2. ¿Y negro solo a los segundos `` (Lunes, Martes, Enero...)?
3. ¿En qué **lugar de la hoja de estilos** añadirías esos estilos?
4. Y si anidamos un tercer `<div>` justo antes de las dos listas, ¿en que cambiaría?

Buenas prácticas con CSS3

Ejercicio 1 (solución)

1. ¿Cómo se aplicaría color de letra naranja solo a los primeros *li*?
 - Solución genérica: `div>div>ul` {color: orange; }
 - Solución más específica:
`.divHorasDiasMeses>ul` {color: orange; }
2. ¿Y negro solo a los segundos *li*? ¡Hay muchas soluciones!
 - `ul>ul` {color: black; }
3. ¿En qué lugar de la hoja de estilos añadirías esos estilos?
 - Son los más específicos, así que al final.
4. Y si anidamos un tercer *div* antes de las listas, ¿en que cambia?
 - Depende de cada caso. Por ejemplo, con la solución genérica de (1) aparece un borde negro pero los demás estilos no cambian, ya que sigue habiendo una secuencia de dos *div* y dos *ul* dentro.

Aplicando estilos

Estructura recomendada y uso de frameworks

Agrupamiento:

- El *mismo estilo* para *mismas etiquetas* o *clases*:

```
body, form, input { color:#000; font-size:12px; }
```

Flujo de estilos:

- Al comienzo del archivo, los estilos más generales.
- Conforme se avanza, se especifica con más detalle.
- Dividir **por bloques o en varias hojas** (una general, otras específicas).
- De forma ideal, una página solo cargaría los estilos que va a usar.

Uso de *frameworks* CSS: [Lista de frameworks CSS](#).

- Proporcionan estilos básicos. Nos permiten centrarnos en crear un diseño para los elementos propios de nuestra página.
- Usar un framework + **Crear una hoja propia** para personalizar.

Contenido

- Introducción
- Aplicando estilos
- Buenas prácticas con CSS3
- **Elementos fundamentales**
- *Media queries*
- Elementos extra
- Preprocesadores de CSS

Elementos fundamentales

Formato en los textos

Familias de fuentes

Se puede especificar más de una.
Desde la más específica a la más genérica.

```
body{  
  font-family: helvetica, sans-serif;  
}
```

Tamaños

Texto en px **Texto en em** Texto en %

Font-size. Tamaño de letra. Se suele usar *px*, *em* y *porcentaje* como *medida*.

```
.tam-em{ font-size: 1.6em; }
```

Decoración del texto

Texto subrayado ~~Texto tachado~~

```
.underline{text-decoration: underline;}
```

Font-weight

Texto negrita Texto normal

```
.negrita{ font-weight: bold; }
```

Alineación

Texto centrado.

```
.divTextoCentrado{  
  text-align:center;
```

Colores

Texto verde Texto amarillo Texto azul

```
.verde{ color: ■ green; }  
.amarillo{ color: ■ rgb(255, 255, 0); }  
.azul{ color: ■ hsl(240, 100%, 50%); }
```

Ver formato_textos.html/formato_textos.css

Elementos fundamentales

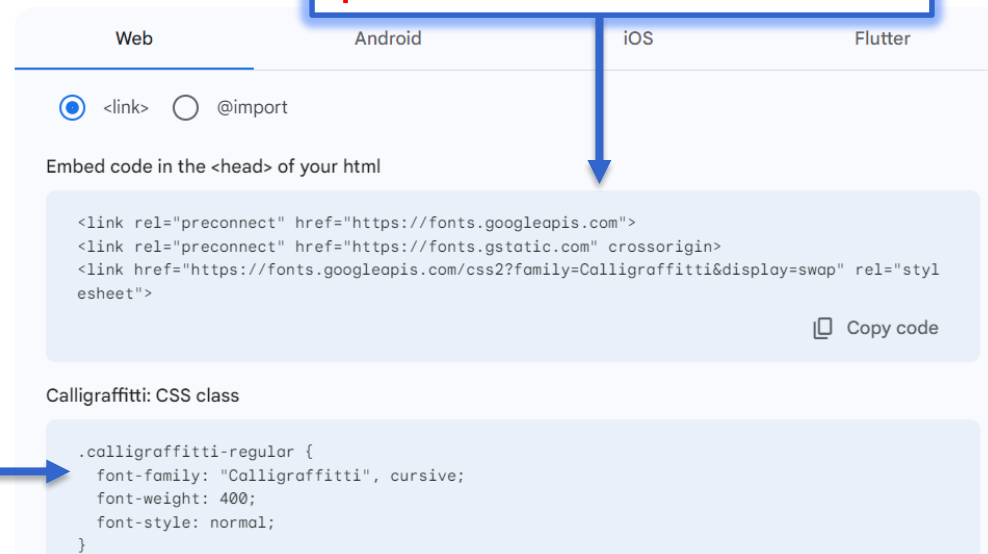
Formato en los textos: @font-face

- **@font-face** : Se usa para especificar fuentes a usar en una web, aunque no estén instaladas en el sistema del usuario.
- Google proporciona gran diversidad de [fuentes](#) así como un [tutorial para su uso](#).
- **Para incluir una fuente:**

1. Copia el `<link>` y ponlo dentro del `<head>`



2. Usa la fuente en tu archivo CSS.



Elementos fundamentales

Colores y fondo (i)

Colores: Diversas formas de representarlo. Entre ellas: ([+ info](#))

- **Nombre:** *red, blue*, etc ([colores](#) predefinidos);
- **RGB** (*Red, Green, Blue, [Alpha]*): `rgb(0,0,0);` `rgba(0,0,0,0.5);`
- **HSL** (*Hue, Saturation, Lightness, [Alpha]*): `hsla(0%,0%,0%,0.5);`
- **Hexadecimal:** `#00F3A1`

Aplicación de gestión de inve

Fondos: Puede tener fondo cualquier elemento: *html, div*, etc. ACTUALIZA

- Un fondo puede ser:
 - un **color** (*background-color*)
 - una **imagen** (*background-[image/position/repeat]*)
 - un **gradiente**: <http://www.colorzilla.com/gradient-editor/>
- ```
background: linear-gradient(to right,
 rgba(60, 110, 50, 10) 0%,
 rgba(50, 116, 45, 0) 100%);
```
- Consultar los archivos **fondos.html** y **fondos.css**.



| DATOS DEL INVESTIGADOR   |  |
|--------------------------|--|
| Apellidos, Nombre:       |  |
| Centro:                  |  |
| Departamento:            |  |
| Categoría Docente:       |  |
| Correo electrónico:      |  |
| Grupo Investigación:     |  |
| Instituto Investigación: |  |

| DATOS SICA ACTUALIZACIÓN     |                                   |
|------------------------------|-----------------------------------|
| Datos personales, foto, etc. | <input type="button" value="SÍ"/> |

| APORTACIÓN DE CALIDAD                                     |                                   |
|-----------------------------------------------------------|-----------------------------------|
| ¿Cuenta con una aportación a la Evaluación de la Calidad? |                                   |
| Indicar los criterios de conocimiento elegidos.           | <input type="button" value="SÍ"/> |
| En caso afirmativo, tener en cuenta los criterios CNEAI.  |                                   |

# Elementos fundamentales

## Márgenes y bordes

**Márgenes:** Hay dos tipos de márgenes:

- **Padding** → margen interno desde el borde del bloque al contenido.
- **Margin** → margen desde el borde del bloque a elementos externos.

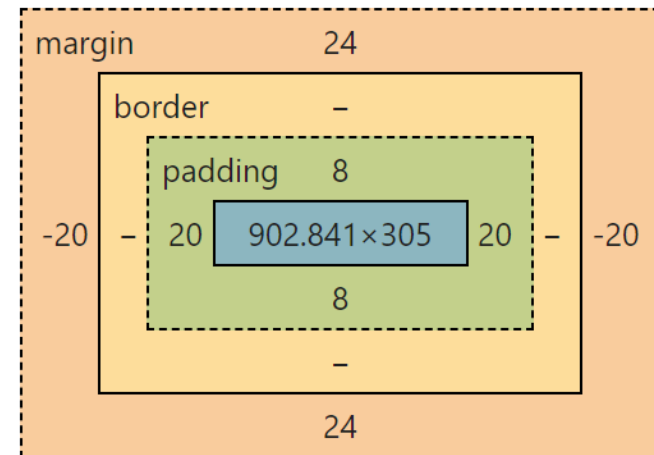
Dos opciones de aplicación del estilo (*ídem para padding*):

- Por separado: `margin-top: 4px; margin-right: 4px...`
- En conjunto: `margin: 4px;`

**Bordes:**

- `border-width`: px, em, thin,...
- `border-style`: none, dashed, solid,...
- `border-color`.
- Otra opción de uso más directa es:

`border: 1px solid ■ red;`



# Elementos fundamentales

## Posicionando los elementos (i): Centrar básico

### Centrado básico de un bloque o una imagen:

- HORIZONTAL:

#### Centrando una imagen

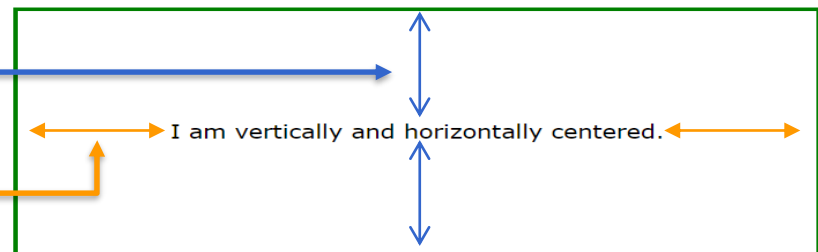
Para centrar un bloque, ajusta los márgenes a auto. En el caso de la imagen, ponle `display: block`.

```
img {
 display: block;
 margin-left: auto;
 margin-right: auto;
}
```



- VERTICAL: Varias opciones, por ejemplo:

```
.center {
 padding: 70px 0;
 border: 3px solid green;
 text-align: center;
}
```



Más información sobre posicionamiento de elementos y alineación:

- CSS tips & tricks de W3C [+info](#)
- W3Schools [+info](#)

# Elementos fundamentales

## Posicionando los elementos (ii): CSS Flexbox

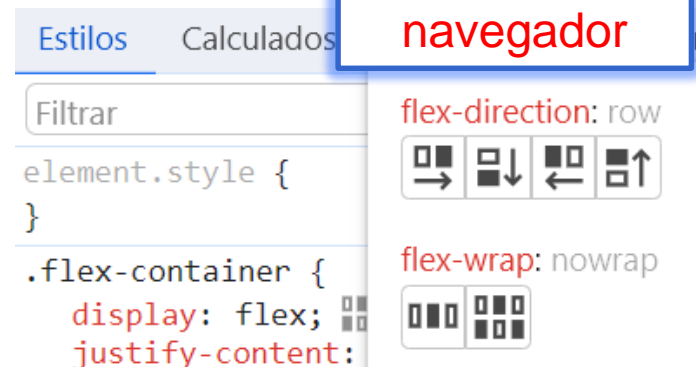
### Flexbox:

#### Guía de flexbox



- Facilita la creación de estructuras responsivas.  
Un contenedor se hace *flexible* con: `display: flex;`
- Las propiedades del contenedor padre más usadas son:
  - *Flex-direction*: dirección de elementos (horizontal o vertical).
  - *Flex-wrap*: cómo el contenedor *envuelve* los elementos.
  - Alineación de elementos (depende de *flex-direction*):
    - *justify-content*: en horizontal.
    - *align-items*: en vertical.
- Las propiedades para los hijos incluyen: *order*, *flex-basis*...

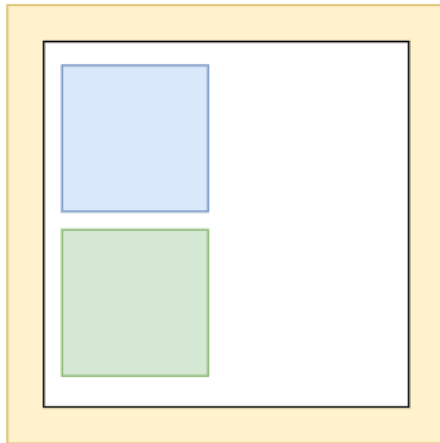
Consultar [estructura\\_básica.html](#)



Pruébalo en el navegador

# Elementos fundamentales

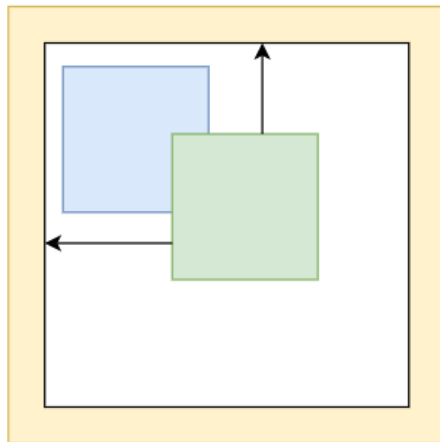
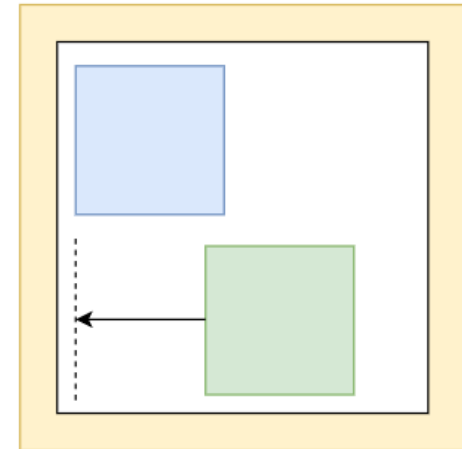
## Posicionando los elementos (iii): position/float



**Static:** El valor por defecto. Se sigue el flujo normal, y no le afecta *left*, *right*, *top* o *bottom*.

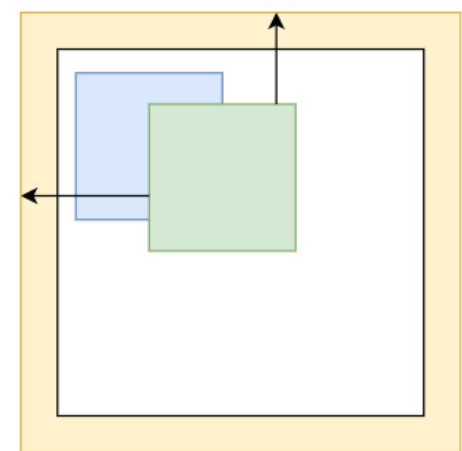
**Relative:** Se posiciona de forma relativa a su posición actual con *left*, *right*, *top* o *bottom*.

Ver [posición\\_elems.html/.css](http://posición_elems.html/.css)



**Absolute:** toma de referencia el ancestro (no static) más cercano. Acepta *left*, *right*, *top* o *bottom*

**Fixed:** toma como referencia el navegador, por lo que se desplaza con el *scrolling*.



Además, se puede hacer que los elementos “floten”: propiedad [float](#).

# Elementos fundamentales

## Posicionando los elementos (iv)

- **Ejercicio:** [calculadora.html](#) - [estilos-float/estilos-flex.css](#)

Probar usando “Inspeccionar” del navegador:

- ¿Qué ocurre si desactivamos la propiedad *float* de *.boton*?
- ¿Cambia algo si desactivamos *overflow: hidden;* en *.botones*?

[Más información sobre elementos flotantes](#)

- Haz que el botón igual, en lugar de estar centrado, aparezca de forma relativa a 90px del lado izquierdo de la calculadora.

**Intercambia ahora la hoja de estilo por la de *flex*:**

- ¿Cómo se consigue la separación entre los botones de una misma línea y cómo se conseguía en la versión con *float*?
- Elimina una operación en la primera fila y añade un nuevo número en la segunda fila. ¿Qué ocurre con cada versión *css*?

# Elementos fundamentales

## Pseudo elementos/clases (i)

Pseudoelementos Dan estilos a partes específicas de un elemento

| Pseudoelemento | Ejemplo        | Descripción                                 |
|----------------|----------------|---------------------------------------------|
| ::first-letter | p:first-letter | Selecciona la primera letra de los párrafos |
| ::first-line   | p:first-line   | Selecciona la primera línea de los párrafos |
| ::before       | p:before       | Inserta contenido antes de cada párrafo     |
| ::after        | p:after        | Inserta contenido después de cada párrafo   |

Pseudoclases Definen estilos en ciertos estados especiales

| Pseudoclase  | Ejemplo       | Descripción                                             |
|--------------|---------------|---------------------------------------------------------|
| :link        | a:link        | Selecciona todos los enlaces sin visitar                |
| :visited     | a:visited     | Selecciona todos los enlaces visitados                  |
| :active      | a:active      | Selecciona todos los enlaces activos                    |
| :hover       | a:hover       | Selecciona los enlaces “pisados” por el ratón           |
| :focus       | input:focus   | Selecciona el input con el foco                         |
| :first-child | p:first-child | Selecciona los párrafos que son primer hijo de su padre |



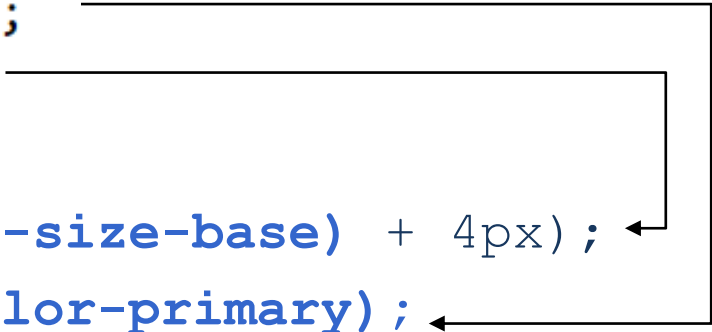
# Elementos fundamentales

## Pseudo elementos/clases (ii)

- `:root`: pseudo-clase para definir *Custom Properties*:

```
:root {
 --color-primary: ■royalblue;
 --font-size-base: 16px; }

.header {
 font-size: calc(var(--font-size-base) + 4px);
 background-color: var(--color-primary);
}
```

A diagram with two horizontal lines. The top line starts at the end of the line `--font-size-base: 16px; }` and extends to the right, then turns down and left, ending with an arrow pointing to the `var(--font-size-base)` in the `font-size` property of the `.header` rule. The bottom line starts at the end of the line `var(--color-primary);` and extends to the right, then turns down and left, ending with an arrow pointing to the `var(--color-primary)` in the `background-color` property of the `.header` rule.

**Ejemplo:** Prueba a visualizar el archivo [dropdown.html](#)

- Analiza el código, ¿cómo se logra que se muestre la lista?
- ¿Y cómo hacer que, mientras se pulsa un enlace, este se muestre en **violeta**?
- Convierte al blanco en un *custom property* y aplícalo.

# Contenido

- Introducción
- Aplicando estilos
- Buenas prácticas con CSS3
- Elementos fundamentales
- ***Media queries y responsividad***
- Elementos extra
- Preprocesadores de CSS

# Media queries y responsividad

## Consulta y propiedades

**Media queries:** Permiten **aplicar estilos distintos** según:

- El *dispositivo* que accede (*screen* (o *all*) es el más común):

```
@media all{
 /*estilos para todos los medios */
}

@media print{
 /*estilos para imprimir la página */
}
```

- Ciertas *características* y parámetros específicos:

```
@media screen and (min-width:700px)
 and (max-width: 1000px){
 /* dispositivos con ancho de
 pantalla entre 700 y 1000px; */
}
```

Otras características seleccionables son:

- Orientación (*landscape*, *portrait*).
- Ratio de ancho/alto.



Desktop



Tablet



Phone

Dibujos de  
w3schools.com

# Media queries y responsividad

## Aplicación *media query* y Responsive web design

- **Para aplicar *media queries* tenemos dos opciones:**
  - Se añaden en un mismo fichero CSS. Las distintas *media queries* se van comprobando y se aplican las que cumplan la condición.
  - Se añaden unos ficheros de estilos u otros según el media query. Ej.:

```
<link rel="stylesheet"
```

```
 media="screen and min-width: 768px" href="style-768.css">
```

```
<link rel="stylesheet"
```

```
 media="screen and min-width: 1200px" href="style-1200.css">
```

- Importante para maquetar pensando en todos los dispositivos
- Para ampliar conocimientos: Media queries

## Responsive Web Design

- Permite diseñar una web que se adapta al redimensionar.
- Recomendación para móviles:

```
<meta name="viewport" content="width=device-width,
 initial-scale=1.0">
```

# Contenido

- Introducción
- Aplicando estilos
- Buenas prácticas con CSS3
- Elementos fundamentales
- *Media queries* y responsividad
- **Elementos extra**
- Preprocesadores de CSS

# Elementos extra

## Sombras y múltiples columnas

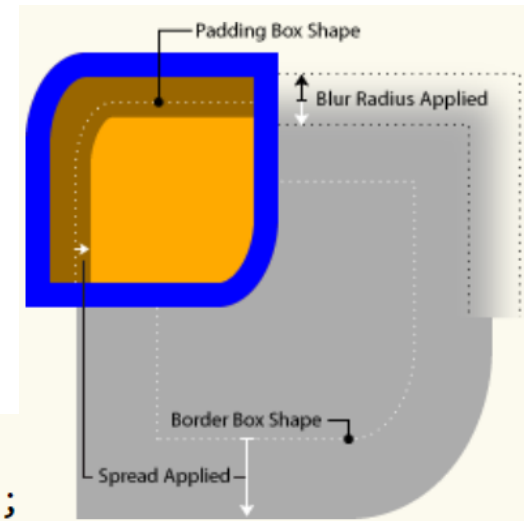
- **Nota inicial:** probar estos elementos en <http://css3generator.com/>

### Sombras:

- En CSS3 se puede añadir una sombra a cualquier bloque, mediante la propiedad *box-shadow*.
- Las propiedades son:
  - la longitud horizontal y vertical de la sombra.
  - difuminación (*blur*) y extensión (*spread*).
  - el color de la sombra y si es interna (*inset*).

[horizontal offset] [vertical off.] [blur] [spread] [color] [inset]

```
box-shadow: 64px 64px 12px 40px rgba(0,0,0,0.4),
 12px 12px 0px 8px rgba(0,0,0,0.4) inset;
```



### Múltiples columnas:

- *column-[count/gap]*: adapta nuestro bloque como multicolumna.

# Elementos extra

## Esquinas redondeadas y animaciones

### Esquinas redondeadas:

- Para *redondear las esquinas* de un elemento con borde.
- Se puede añadir con:
  - `border-radius: 5px;`
  - `border-radius: arriba-izq arriba-der abajo-der abajo-izq;`



```
.tab{
 -webkit-border-radius: 10px 10px 0px 0px;
 border-radius: 10px 10px 0px 0px;
 /* Se detalla para cada esquina individualmente */
}
```

### Animaciones:

- Para *animar los objetos estáticos* (todas las propiedades):
  - Se define un *keyframe rule*, donde se crea la animación.
  - Luego se añade a *animation-name* la regla creada.

# Contenido

- Introducción
- Aplicando estilos
- Buenas prácticas con CSS3
- Elementos fundamentales
- *Media queries* y responsividad
- Elementos extra
- **Preprocesadores de CSS**



# Preprocesadores de CSS

## Introducción

### Preprocesadores:

- Son *lenguajes* que, al ser compilados, genera un CSS más eficiente y con menos esfuerzo del programador.
- Evita tareas repetitivas, anida estilos, permite definir macros y funciones, entre otras características.
- Los más usados son **LESS** y **SASS**, pero hay otros como *Stylus* ([Comparativa](#)).
- Las web oficiales con información de los mismos son:
  - **SASS:** (*Syntactically Awesome Style Sheets*) <http://sass-lang.com/>
  - **LESS:** (*Leaner Style Sheets*) <http://lesscss.org/>

# Preprocesadores de CSS

## LESS y SASS



### LESS

- Dos opciones de uso:
  - Compilar con *lessc* previamente (**recomendado**).
  - Dejar que el navegador interprete el código *.less*, añadiendo el código en una etiqueta *style* (*on the fly*).
- Mejoras:
  - *Uso de variables, funciones, operaciones aritméticas...*
  - *Herencia, mejora de anidación...*

### SASS

- Hay dos tipos de ficheros con distinto formato:
  - *scss*. Formato más parecido a CSS.
  - *sass*. Cambia levemente la sintaxis, al no usar llaves de anidamiento ni puntos y comas de cierre.
- Hay que compilar el archivo y genera un *.css*.



# Referencias

## *Bibliografía:*

- 📖 **Basics of Web Design: HTML5 & CSS3.**  
Terry Felke-Morris. Addison-Wesley, 2ª Ed., 2014
- 📖 **Responsive Web Design with HTML and CSS: Building future-proof responsive websites using the latests HTML5 and CSS techniques.**  
B. Frain. Packt Publishing, 4ª Ed, 2022.

## *Recursos web:*

- **Validador CSS:** <http://jigsaw.w3.org/css-validator/>
- **Características soportadas** en cada navegador: <http://caniuse.com>
- **Información/Noticias** sobre CSS3: <http://www.w3.org/Style/CSS/>
- **Material icons:** Fáciles de usar, disponibles PNG y SVG, modificables con CSS.
  - [https://developers.google.com/fonts/docs/material\\_icons](https://developers.google.com/fonts/docs/material_icons)Además, se pueden hacer uso de ellos y editarlos con Figma:  
<https://www.figma.com/community/file/1014241558898418245>