



**M E A N**

**WEB FULL STACK DEVELOPER**

*Germán Caballero Rodríguez*  
*germanux@gmail.com*



# CSS3 introducción



# CSS3

# INDICE

- 1) Introducción a CSS3
- 2) Maquetación de una web
- 3) Animaciones CSS3
- 4) Gradientes

## Intro CSS3

- Con respecto a la estructura, básicamente cada navegador ordena los elementos por defecto de acuerdo a su tipo:
  - block (bloque) o inline (en línea).
- Esta clasificación está asociada con la forma en que los elementos son mostrados en pantalla.

## Intro CSS3

- Elementos block son posicionados uno sobre otro hacia abajo en la página.
- Elementos inline son posicionados lado a lado, uno al lado del otro en la misma línea, sin ningún salto de línea a menos que ya no haya más espacio horizontal para ubicarlos.

# Intro CSS3

## SELECTORES CSS3

- Permiten seleccionar elementos del HTML
  - **Selector universal** -> Se aplica a todas las etiquetas

```
* {}
```

- **Selector de tipo o etiqueta** -> Se aplica a todos las etiquetas div

```
div {}
```

- **Agrupación de selectores** -> Se aplica a todas las etiquetas h1 y h2

```
h1,h2 {}
```

# Intro CSS3

## SELECTORES CSS3

- **Selector descendente** -> Se aplica a todas las etiquetas p, que sean descendientes de una div, independientemente del grado (hijo, nieto,...)

```
div p {}
```

- **Selector de clase** -> Se aplica a las etiquetas, que tengan como atributo class, el valor cabecera

```
.cabecera {}
```

# Intro CSS3

## SELECTORES CSS3

- **Selector de Id** -> Se aplica a las etiquetas que tengan como atributo id, el valor cabecera

```
div#cabecera {}
```

- **Selector de hijos** -> Se aplica a los p, que sean hijos directos de un div

```
div > p {}
```

- **Selector adyacente** -> Se aplica a los p, que estén justo después de un div

```
div + p {}
```

- **Selector hermano** -> Se aplica a los p, que sean hermanos de un div y que estén después, sin ser tan estricto como el anterior

```
div ~ p {}
```



# Intro CSS3

## SELECTORES CSS3

- **Selector de atributos**

- Selecciona los div que tengan el atributo class

```
div[class]{}
```

- Selecciona los div que tengan en el atributo class el valor cabecera

```
div[class="cabecera"]
```

# Intro CSS3

## SELECTORES CSS3

- **Selector de atributos**

- Selecciona los div que tengan el atributo class

```
div[class]{}
```

- Selecciona los div que tengan en el atributo class el valor cabecera

```
div[class="cabecera"]
```

# Intro CSS3

## PSEUDO-CLASES CSS3

- **:link** -> Selecciona los enlaces que no hayan sido visitados por los usuarios
- **:visited** -> Selecciona los enlaces que han sido visitados al menos una vez

# Intro CSS3

## PSEUDO-CLASES CSS3

- **:hover** -> Selecciona el elemento sobre el cual pasa el ratón.
- **:active** -> Selecciona el elemento que activa el usuario, a través de un **click**.
- **:focus** -> Selecciona el elemento que tiene el foco del navegador.

■  
Más ejemplos: <https://goo.gl/I9I26N>

# Intro CSS3

## Selectors

<code>.class</code>	Selects all elements with <code>.class</code>
<code>#id</code>	Selects all elements with <code>#id</code>
<code>*</code>	Selects all elements
<code>div</code>	Selects all <code>&lt;div&gt;</code> elements
<code>div, p</code>	Selects all <code>&lt;div&gt;</code> and <code>&lt;p&gt;</code> elements
<code>div &gt; p</code>	Selects all <code>&lt;p&gt;</code> elements that are a direct descendant of a <code>&lt;div&gt;</code> element
<code>div + p</code>	Selects all <code>&lt;p&gt;</code> elements that are the next sibling of (i.e. placed directly after) <code>&lt;div&gt;</code> elements
<code>div ~ p</code>	Selects all <code>&lt;p&gt;</code> elements that follow, and are siblings of, <code>&lt;div&gt;</code> elements

# Intro CSS3

## Attributes

[attribute]	Selects all elements with the specified attribute
[attribute=value]	Selects all elements where the specified attribute is equal to 'value'
[attribute~value]	Selects all elements with an attribute containing the word 'value'
[attribute =value]	Selects all elements with an attribute list starting with 'value'
[attribute^=value]	Selects all elements with an attribute beginning with 'value'
[attribute\$=value]	Selects all elements with an attribute ending with 'value'
[attribute*=value]	Selects all elements with an attribute containing the substring 'value'

# Intro CSS3

## Pseudo-classes

<code>a:link</code>	Selects all unvisited links
<code>a:visited</code>	Selects all visited links
<code>a:hover</code>	Selects links on mouse hover
<code>a:active</code>	Selects the active link element

<code>p::after</code>	Insert content after <p> element
<code>p::before</code>	Insert content before <p> element
<code>input:checked</code>	Selects every checked <input> element
<code>input:disabled</code>	Selects every disabled <input> element
<code>p:empty</code>	Selects every <p> element with no children
<code>input:enabled</code>	Selects every enabled <input> element
<code>p:first-child</code>	Selects every <p> element that is the first child of its parent

# Intro CSS3

<code>p::first-letter</code>	Selects the first letter of every <p> element
<code>p::first-line</code>	Selects the first line of every <p> element
<code>p:first-of-type</code>	Selects every <p> element that is the first <p> element of its parent
<code>input:focus</code>	Selects the <input> element which has focus
<code>input:in-range</code>	Selects <input> elements with a value within a specified range
<code>input:invalid</code>	Selects all <input> elements with an invalid value
<code>p:lang(<i>language</i>)</code>	Selects all <p> elements with a lang attribute equal to 'language'
<code>p:last-child</code>	Selects every <p> element which is the last child of its parent
<code>p:last-of-type</code>	Selects every <p> element which is the last <p> element of its parent
<code>:not(p)</code>	Selects every element that is not a <p>
<code>p:nth-child(2)</code>	Selects every <p> element that is the second child of its parent



# Intro CSS3

<code>p:nth-last-child(2)</code>	Selects every <p> element that is the second child of its parent, counting from the last child
<code>p:nth-last-of-type(2)</code>	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<code>p:nth-of-type(2)</code>	Selects every <p> element that is the second <p> element of its parent
<code>p:only-of-type</code>	Selects every <p> element that is the only <p> element of its parent
<code>p:only-child</code>	Selects every <p> element that is the only child of its parent
<code>input:optional</code>	Selects <input> elements with no 'required' attribute

<code>input:out-of-range</code>	Selects <input> elements with a value outside a specified range
<code>input:read-only</code>	Selects <input> elements with the 'readonly' attribute specified

# Intro CSS3

<code>input:read-only</code>	Selects <input> elements with the 'readonly' attribute specified
<code>input:read-write</code>	Selects <input> elements with the 'readonly' attribute not specified
<code>input:required</code>	Selects <input> elements with the 'required' attribute specified
<code>:root</code>	Selects the documents root element
<code>::selection</code>	Selects the portion of an element that is selected by a user
<code>#id:target</code>	Selects the current active #id element
<code>input:valid</code>	Selects all <input> elements with a valid value

# Intro CSS3

odd y even (pares e impares)

```
p:nth-child(1) {  
  background: #999999;  
}  
p:nth-child(2) {  
  background: #CCCCCC;  
}  
p:nth-child(3) {  
  background: #999999;  
}  
p:nth-child(4) {  
  background: #CCCCCC;  
}
```

```
p:nth-child(odd) {  
  background: #999999;  
}  
p:nth-child(even) {  
  background: #CCCCCC;  
}
```

# Intro CSS3

## ESPECIFICIDAD DE SELECTORES

- Una de las mayores causas de confusión al seleccionar elementos CSS es la **especificidad**
- Estrictamente, no es algo propio de CSS3, porque funcionaba igual en CSS 2
- Entenderlo correctamente, hace que escribamos mejores selectores (lo más específicos posible)

# Intro CSS3

## ESPECIFICIDAD DE SELECTORES

- Cuando queremos escribir un selector CSS, debemos escribirlo de la forma más específica posible. Es decir: **de forma que aseguremos que solo el elemento o elementos que queremos seleccionar, son seleccionados.**
- Para evaluar la especificidad de un selector CSS, utilizamos un **sistema posicional.**
- Ante dos selectores que puedan aplicar a un mismo elemento, **siempre ganará el más específico de los dos.**

# Intro CSS3

## ESPECIFICIDAD DE SELECTORES



- **X** → Número de selectores de ID.
- **Y** → Número de selectores de clase, atributo o pseudoclase.
- **Z** → Número de selectores de elemento o pseudoelemento
- **IN** → Estilos aplicados inline (siempre son más específicos que los aplicados mediante hoja de estilos)
- **!** → Estilo que incluye **!important**. No es buena práctica
- Detalles:
  - El selector \* equivale a X=0, Y=0, Z=0
  - +, >, ~, not: no modifican la especificidad

# Intro CSS3

## ESPECIFICIDAD DE SELECTORES

- Veamos algunos ejemplos para x-y-z
  - \* → 0-0-0
  - div → 0-0-1
  - li > ul → 0-0-2
  - .myClass → 0-1-0
  - [type=checkbox] → 0-1-0
  - li[attr] → 0-1-1
  - form input[type=email] → 0-1-2
  - #myDiv → 1-0-0
  - #myDiv li.class a[href] → 1-2-2
  - style="..." → 1-0-0-0
  - !important → 1-0-0-0-0

# Intro CSS3

## ESPECIFICIDAD DE SELECTORES

- Ejemplo

- 

```
<div id="main">  
  <div id="content">  
    <p class="note">¿De qué color soy?<p>  
  </div>  
</div>
```

```
.note {  
  background-color:  
blue;  
}
```

```
div > div > p {  
  background-color: red;  
}
```

```
#content .note {  
  background-color:  
green;  
}
```



# Intro CSS3

## ESPECIFICIDAD DE SELECTORES

- Ejemplo

- 

```
<div id="main">  
  <div id="content">  
    <p class="note">¿De qué color soy?<p>  
  </div>  
</div>
```

```
.note {  
  background-color:  
  blue;  
}
```

0-1-0

```
div > div > p {  
  background-color: red;  
}
```

0-0-3

```
#content .note {  
  background-color:  
  green;  
}
```

1-1-0

# Intro CSS3

## IDENTIFICADORES EXTENDIDOS

- Algunas propiedades de CSS, están en fase de borrador, es decir no hay obligatoriedad de implementarla por parte de los navegadores, aunque algunos las implementan.
- Cuando esto ocurre, se ha de acceder a estas propiedades con identificadores propios del navegador.
- Sintaxis

```
propiedad /* Oficial */  
-ms-propiedad /* Internet Explorer */  
-moz-propiedad /* Mozilla Firefox */  
-o-propiedad /* Opera */  
-webkit-propiedad /* Chrome, Safari */
```

# Intro CSS3

## BORDER-RADIUS

- Propiedad que permite añadir bordes redondeados a las cajas.
- Sintaxis

```
border-radius: <radio>;
```

- Algún ejemplo

```
border-radius: 5px;
```

# Intro CSS3

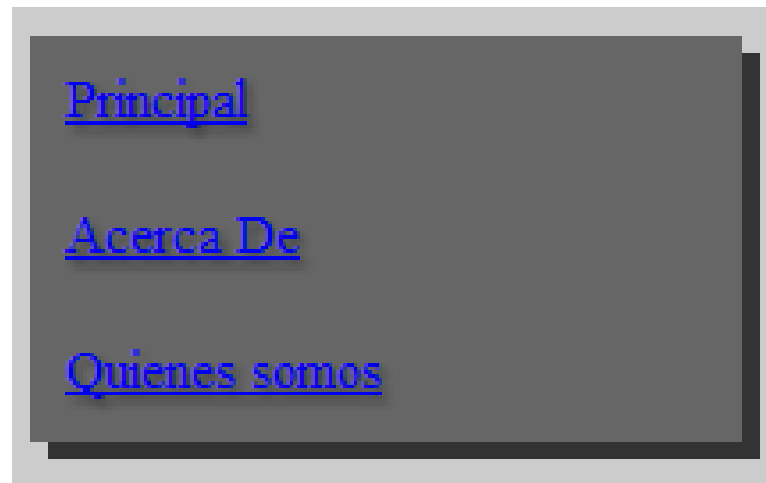
## BOX-SHADOW

- Propiedad que permite aplicar una sombra a las cajas.
- Sintaxis

```
box-shadow: <inset>  
            <desplazamiento horizontal>  
            <desplazamiento vertical>  
            <difuminado>  
            <color RGB>;
```

- Algún ejemplo

```
box-shadow: 5px 9px 3 #000;
```



# Intro CSS3

## TEXT-SHADOW

- Propiedad que permite aplicar una sombra a los textos.
- Sintaxis

```
text-shadow: <desplazamiento horizontal>  
            <desplazamiento vertical>  
            <difuminado>  
            <color RGB>;
```

- Ejemplos

```
text-shadow: 2px 2px 3 #000;
```

**TITULO**

# Intro CSS3

## COLUMNS

- Esta propiedad/propiedades son bastante útiles para cuando se quiere separar en columnas un texto o un grupo de elementos:
- Se puede dividir en varias propiedades:

***column-count:*** define el numero de columnas.

***column-rule:*** añade un separador entre las columnas, mismo sintaxis que un borde.

***column-gap:*** la distancia que hay entra las columnas.

***column-width:*** define el ancho de cada columna, por defecto es auto.

# Maquetación de una web

## La propiedad float de CSS

- Los diseñadores de páginas web comenzaron la transición de la maquetación con tablas
- Actualmente está casi abandonada y totalmente desaconsejada.
- La maquetación con CSS ofrece más posibilidades
- Sin el uso de la propiedad float, la maquetación con CSS resultaría imposible.

# Maquetación de una web

## La propiedad float de CSS

- Una de las diferencias más notables es que para colocar un elemento junto a otro en el plano horizontal, con las tablas bastaba con colocar dos celdas adyacentes en una misma fila.
- La forma de hacer lo mismo con CSS, es empleando la propiedad float.



# Maquetación de una web

## La propiedad float de CSS

- La propiedad float de CSS es la que permite que los elementos floten sobre el lado que se indica.
- El resto de los elementos que no tienen la propiedad float definida se encuentran alrededor.
- Por defecto, los elementos cuya propiedad float no se encuentra definida, es none.

# Maquetación de una web

## La propiedad float de CSS

- El primero define que el elemento fluirá hacia la izquierda.
- El segundo a la derecha, mientras que el tercero quita la propiedad.

Sintaxis:

```
1 float: left;  
2 float: right;  
3 float: none;
```

## Maquetación de una web

### Características de la propiedad float

- Los elementos cuya propiedad float tengan el valor left, flotarán a la izquierda **hasta el límite del elemento que los contenga** o hasta el límite marcado por el padding del mismo.

# Maquetación de una web

```
<head>
<style>

#caja1 {
width: 190px;
border: solid 1px #000;
background-color: #ccc;
padding: 5px;
font-family: arial;
margin-right: 10px;
float:left; }

#caja2 {
width: 320px;
border: solid 1px #00f;
background-color: #99f;
padding: 5px;
font-family: verdana;
float:left; }

</style>
</head>
<body>
<div id="caja1">
<p>Pellentesque habitant morbi tristique senectus et netu
</div>
<div id="caja2">
<p>Aenean ultricies mi vitae est. Mauris placerat eleifem
</div>
</body>
```

## Maquetación de una web

# Características de la propiedad float

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Suspendisse tortor Vestibulum feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Eu libero Donec sit amet quam egestas semper.

Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Aenean ultricies mi vitae est Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Wisi Vestibulum erat, condimentum sed, commodo vitae, ornare sit amet, WISI. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus.

# Maquetación de una web

## Propiedad display

- Establece el tipo de la caja que genera cada elemento.
- La propiedad display es tan compleja que casi ningún navegador es capaz de mostrar correctamente todos sus valores.
- Display es **none** que hace que el elemento no genere ninguna caja.
- Si se utiliza la propiedad display: none sobre un elemento, todos sus descendientes también desaparecen por completo de la página.

# Maquetación de una web

## Propiedad display

- Si se quiere hacer un elemento invisible, es decir, que no se vea pero que siga ocupando el mismo sitio, se debe utilizar la propiedad visibility.
- La propiedad display: none se utiliza habitualmente en aplicaciones web dinámicas creadas con JavaScript y que muestran/ocultan contenidos cuando el usuario realiza alguna acción



# Maquetación de una web

## Propiedad display

- Los otros dos valores más utilizados son block e inline que hacen que la caja de un elemento sea de bloque o en línea respectivamente.

**[display: block]** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non sem quis tellus vulputate lobortis. Vivamus fermentum, tortor id ornare ultrices, ligula ipsum tincidunt pede, et blandit sem pede suscipit pede. Nulla cursus porta sem. Donec mollis nunc in leo.

**[display: inline]** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non sem quis tellus vulputate lobortis. Vivamus fermentum, tortor id ornare ultrices, ligula ipsum tincidunt pede, et blandit sem pede suscipit pede. Nulla cursus porta sem. Donec mollis nunc in leo.



# Maquetación de una web

## Propiedad display

- **Inline-block** crea cajas que son de bloque y en línea de forma simultánea.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non sem quis tellus vulputate lobortis. Vivamus fermentum, tortor id ornare ultrices, ligula ipsum tincidunt pede, et blandit sem pede suscipit pede. Nulla cursus porta sem. Donec

[display: inline-block, width: 25%]  
Quisque semper, magna  
sed pharetra  
tincidunt, quam urna  
dapibus dolor, a  
dignissim sem neque  
id purus. Etiam  
luctus viverra nisi.

mollis nunc in leo. Integer lobortis accumsan felis. Cras

venenatis. Morbi cursus, tellus vitae iaculis pulvinar, turpis nibh posuere nisl, sed vehicula massa orci at dui. Morbi pede ipsum, porta quis, venenatis et, ullamcorper in, metus. Nulla facilisi. Quisque laoreet molestie mi. Ut mollis elit eget urna.

# Maquetación de una web

## Márgenes

- Cada capa tiene cuatro márgenes, los que por defecto tienen un valor cero.
- En el código CSS, el margen se establece mediante el atributo “margin”

# Maquetación de una web

## Márgenes

```
/* El siguiente es el método natural de establecer los
márgenes */
#identificador {
margin-top: 10px;
margin-right: 10px;
margin-bottom: 10px;
margin-left: 10px;
}
/* Su forma abreviada sería la siguiente: */
#identificador {
margin: 10px;
}
/* Cuando los márgenes superior e inferior tienen las
mismas dimensiones y el izquierdo y el derecho diferentes
pero iguales entre si, el código abreviado es el siguiente,
siendo la primera cifra la que corresponde a los márgenes
superior e inferior y la segunda a la izquierda y derecha */
#identificador {
margin: 10px 5px;
}
```

# Maquetación de una web

## Márgenes

/\* Cuando los márgenes derecho e izquierdo tienen el mismo valor, y los márgenes superior e inferior son diferentes y además, diferentes entre si, la forma de abreviar es la que sigue, siendo la primera cifra del margen superior, la segunda de los márgenes laterales y la tercera del margen inferior \*/

```
#identificador {  
margin: 5px 15px 2px;  
}
```

/\* Cuando todos los márgenes son diferentes, el modo de abreviar es el que describimos a continuación, siendo las cifras de los márgenes superior, derecho, inferior e izquierdo respectivamente: \*/

```
#identificador {  
margin: 3px 10px 6px 8px;  
}
```

# Maquetación de una web

## Relleno (Padding):

- El relleno se determina del mismo modo que los márgenes.
- Al igual que en el caso de los márgenes, el relleno se determina lado por lado
- Se puede emplar el método abreviado, que es igual que el empleado en los márgenes.

# Maquetación de una web

## Borde (Border):

- Border-color:
  - determinar el color de los cuatro bordes
- Border-width:
  - Determinar el ancho de todos los bordes en una sola línea de código

# Maquetación de una web

## Bordes redondeados

- Esta propiedad se llama “border radius”.
- Las propiedades completas de CSS para bordes redondeados para todos los navegadores:

```
#caja {  
  background-color: #F00;  
  height: 200px;  
  width: 200px;  
  padding: 10px;  
  /*para Firefox*/  
  -moz-border-radius: 15px 15px 15px 15px;  
  /*para Safari y Chrome*/  
  -webkit-border-radius: 15px 15px 15px 15px;  
  /* para Opera */  
  border-radius: 15px 15px 15px 15px;  
  /* para IE */  
  behavior:url(border.htc);  
}
```

# Maquetación de una web

## Ejercicio

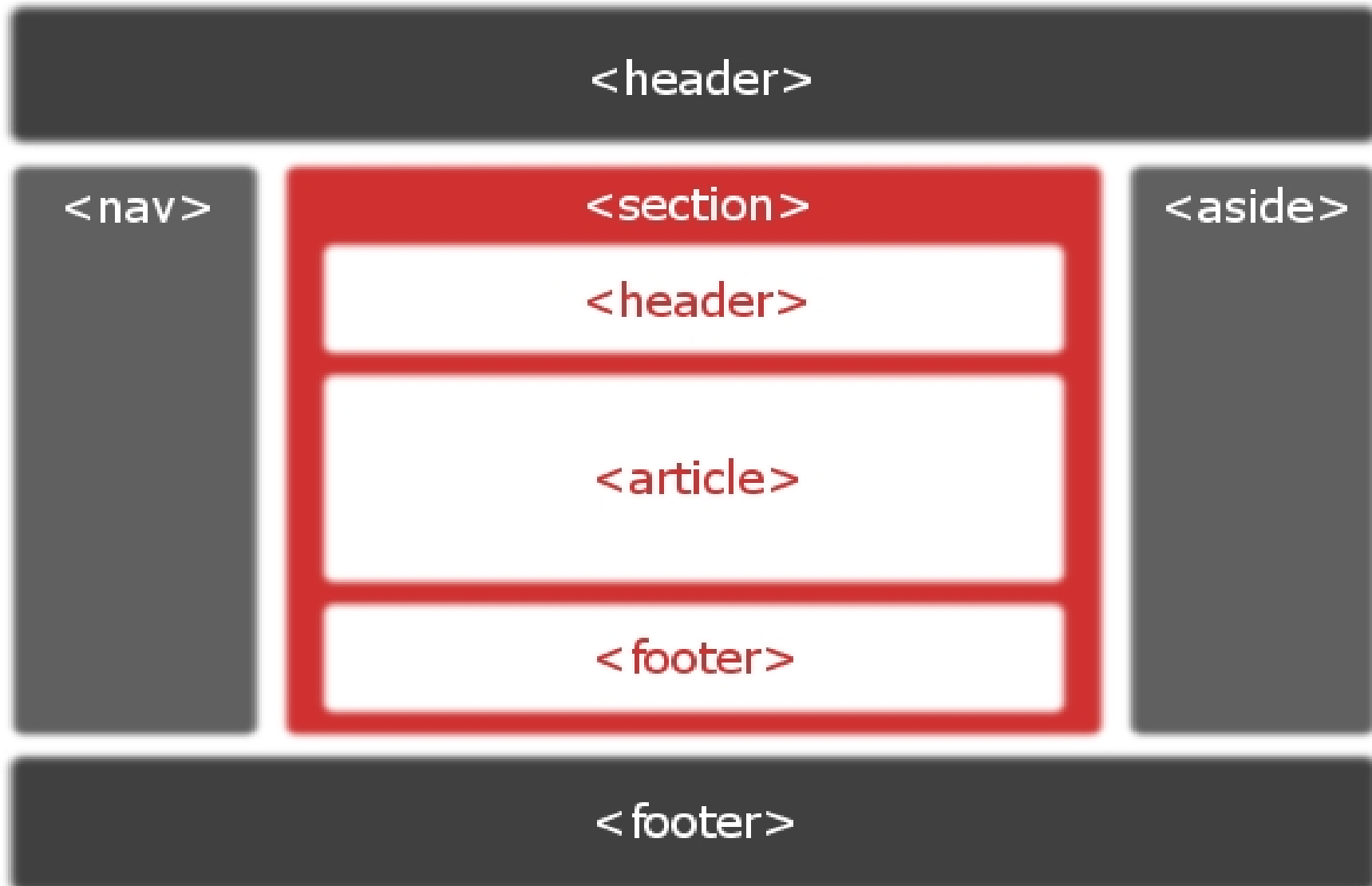
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Bordes Redondeados CSS3</title>
<style type="text/css">
#caja {
  background-color: #F00;
  height: 200px;
  width: 200px;
  /*para Firefox*/
  -moz-border-radius: 15px 15px 15px 15px;
  /*para Safari y Chrome*/
  -webkit-border-radius: 15px 15px 15px 15px;
  /*para IE */
  behavior:url(border.htc);
  /* para Opera */
  border-radius: 15px 15px 15px 15px;
}
</style>
</head>

<body>
<h2>Bordes Redondeados CSS3 </h2>
<div id="caja"></div>
</body>
</html>
```



# Maquetación de una web

## Ejercicio 9 (herencia CSS3)



# Maquetación de una web

## Ejercicio 10

- En el ejercicio anterior:
  - En NAV crear un menú para acceder a tres páginas web (Ejercicio\_10.1.html, Ejercicio\_10.2.html y Ejercicio\_10.3.html) que compartan el mismo estilo CSS3.
  - En la barra lateral <aside> una lista con enlaces otras webs (p.ej. google, facebook y yahoo)
  - Cada una de las páginas llevarán artículos aleatorios.

## Animaciones CSS3

Las animaciones CSS3 permiten animar la transición entre un estilo CSS y otro.

- Las animaciones constan de dos componentes:
  - Un estilo que describe la animación
  - Un conjunto de fotogramas que indican su estado inicial y final
  - Así como posibles puntos intermedios en la misma.

# Animaciones CSS3

- Las animaciones CSS tienen tres ventajas principales sobre las técnicas tradicionales de animación basada en scripts:
  - Es muy fácil crear animaciones sencillas
  - La animación se muestra correctamente, incluso en equipos poco potentes.
  - Al ser el navegador quien controle la secuencia de la animación, permitimos que optimice el rendimiento

# Animaciones CSS3

- Para crear una secuencia de animación CSS usaremos la propiedad **animation** y sus sub-propiedades.
- Con ellas podemos configurar:
  - El ritmo
  - La duración de la animación
  - Otros detalles sobre la secuencia de la animación.

# Animaciones CSS3

- animation-delay
  - Tiempo de retardo entre el momento en que el elemento se carga y el comienzo de la secuencia de la animación.
- animation-direction
  - Indica si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia o si debe comenzar desde el principio al llegar al final.
- animation-duration
  - Indica la cantidad de tiempo que la animación consume en completar su ciclo (duración).

# Animaciones CSS3

- animation-iteration-count
  - El número de veces que se repite. Podemos indicar infinite para repetir la animación indefinidamente.
- animation-play-state
  - Permite pausar y reanudar la secuencia de la animación
- animation-timing-function
  - Indica el ritmo de la animación, es decir, como se muestran los fotogramas de la animación, estableciendo curvas de aceleración.
- animation-fill-mode
  - Especifica qué valores tendrán las propiedades después de finalizar la animación

# Animaciones CSS3

- Una vez configurado el tiempo de la animación, necesitamos definir su apariencia.
- Esto lo haremos estableciendo dos fotogramas más usando la regla `@keyframes`.
  - Cada fotograma describe cómo se muestra cada elemento animado en un momento dado durante la secuencia de la animación.
- Desde que se define el tiempo y el ritmo de la animación, el fotograma usa **percentage** para indicar en qué momento de la secuencia de la animación tiene lugar.
  - 0% es el principio, 100% es el estado final de la animación.
  - Estos dos momentos tienen alias especiales: **from** y **to**.
- Además puedes, opcionalmente, incluir fotogramas que describan pasos intermedios entre el punto inicial y final de la animación.



# Animaciones CSS3

- `h1 {`
- `animation-duration: 3s;`
- `animation-name: slidein;`
- `}`

# Animaciones CSS3

- @keyframes slidein {
- from {
- margin-left: 100%;
- width: 300%
- }
- 
- to {
- margin-left: 0%;
- width: 100%;
- }
- }

# Animaciones CSS3

- Añadiendo otro fotograma
- 75% {
- font-size: 300%;
- margin-left: 25%;
- width: 150%;
- }

# Animaciones CSS3

## Haciendo que se repita

- `h1 {`
- `animation-duration: 3s;`
- `animation-name: slidein;`
- `animation-iteration-count: infinite;`
- `}`

# Animaciones CSS3

- h1 {
- animation-duration: 3s;
- animation-name: slidein;
- animation-iteration-count: infinite;
- animation-direction: alternate;
- }

# Animaciones CSS3

## Usando eventos de animación

- Podemos tener un control mayor sobre las animaciones haciendo uso de eventos de animación.
- Dichos eventos, representados por el objeto `AnimationEvent`, se pueden usar para detectar:
  - Cuándo comienza la animación
  - Cuándo termina
  - Cuándo comienza una iteración.
- Cada evento incluye el momento en el que ocurrió, así como el nombre de la animación que lo desencadenó.

## Animaciones CSS3

# Añadiendo detectores de eventos a la animación

- Podemos usar Javascript para escuchar los tres posibles eventos de animación.
  - function setup() {
  - var e = document.getElementById("watchme");
  - e.addEventListener("animationstart", listener, false);
  - e.addEventListener("animationend", listener, false);
  - e.addEventListener("animationiteration", listener, false);
  - 
  - var e = document.getElementById("watchme");
  - e.className = "slidein";
  - }

# Animaciones CSS3

## Recibiendo los eventos

- Los eventos, al irse disparando, llamarán a la función listener().
- 
- ```
function listener(e) {
```
- ```
  var l = document.createElement("li");
```
- ```
  switch(e.type) {
```
- ```
    case "animationstart":
```
- ```
      l.innerHTML = "Iniciado: tiempo transcurrido " + e.elapsedTime;
```
- ```
      break;
```
- ```
    case "animationend":
```
- ```
      l.innerHTML = "Finalizado: tiempo transcurrido " + e.elapsedTime;
```
- ```
      break;
```
- ```
    case "animationiteration":
```
- ```
      l.innerHTML = "Nueva iteración comenzó a los " + e.elapsedTime;
```
- ```
      break;
```
- ```
  }
```
- ```
  document.getElementById("output").appendChild(l);
```
- ```
}
```



# Animaciones CSS3

## ANIMATION

- Podemos especificar la curva que seguirá la animación con `animation-timing-function`. Puede tomar los siguientes valores
  - **ease**: empieza lento, luego acelera, y termina lento. Es el valor por defecto.
  - **linear**: misma velocidad de principio a fin.
  - **ease-in**: empieza lento
  - **ease-out**: termina lento
  - **ease-in-out**: empieza y termina lento
  - **cubic-bezier(x, y, z, n)**: permite especificar nuestros propios parámetros siguiendo una función de Bezier



# Animaciones CSS3

- Podemos obtener un doble efecto en nuestra animación

```
div {
  width: 100px;
  height: 50px;
  background-color: red;
  font-weight: bold;
  position: relative;
  -webkit-animation-name: mymove; /* Chrome,
    Safari, Opera */
  -webkit-animation-duration: 5s; /* Chrome,
    Safari, Opera */
  animation-name: mymove;
  animation-duration: 5s;
}

/* Chrome, Safari, Opera */
#div1 {-webkit-animation-timing-function: linear;}
#div2 {-webkit-animation-timing-function: ease;}
#div3 {-webkit-animation-timing-function: ease-in;}
#div4 {-webkit-animation-timing-function: ease-
  out;}
#div5 {-webkit-animation-timing-function: ease-in-
  out;}

/* Standard syntax */
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
```

```
/* Chrome, Safari, Opera */
@-webkit-keyframes mymove {
  from {left: 0px;}
  to {left: 300px;}
}

/* Standard syntax */
@keyframes mymove {
  from {left: 0px;}
  to {left: 300px;}
}
```

# Gradientes CSS3

## GRADIENT: LINEAR

- Gradiente lineal: Requieren al menos dos colores (inicio y fin).
- Si no se especifica dirección, es de arriba a abajo

```
#grad {  
    background: red; /* For browsers that do not support gradients */  
    background: -webkit-linear-gradient(red, yellow); /* For Safari 5.1 to 6.0 */  
    background: -o-linear-gradient(red, yellow); /* For Opera 11.1 to 12.0 */  
    background: -moz-linear-gradient(red, yellow); /* For Firefox 3.6 to 15 */  
    background: linear-gradient(red, yellow); /* Standard syntax */  
}
```



# Gradientes CSS3

- Podemos indicar de izquierda a derecha indicando left como punto inicial

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(left, red , yellow); /* For Safari 5.1 to  
    6.0 */  
  background: -o-linear-gradient(right, red, yellow); /* For Opera 11.1 to 12.0  
    */  
  background: -moz-linear-gradient(right, red, yellow); /* For Firefox 3.6 to 15  
    */  
  background: linear-gradient(to right, red , yellow); /* Standard syntax */  
}
```



# Gradientes CSS3

- Podemos generar un gradiente en diagonal, especificando las posiciones horizontales y verticales de inicio

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(left top, red, yellow); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(bottom right, red, yellow); /* For Opera 11.1 to 12.0 */  
  /*  
  background: -moz-linear-gradient(bottom right, red, yellow); /* For Firefox 3.6 to 15 */  
  /*  
  background: linear-gradient(to bottom right, red, yellow); /* Standard syntax */  
}
```



# Gradientes CSS3

- Podemos especificar un ángulo entre la horizontal y la línea del gradiente

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(-90deg, red, yellow); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(-90deg, red, yellow); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(-90deg, red, yellow); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(-90deg, red, yellow); /* Standard syntax */  
}
```



# Gradientes CSS3

- Podemos especificar un ángulo entre la horizontal y la línea del gradiente

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(-90deg, red, yellow); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(-90deg, red, yellow); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(-90deg, red, yellow); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(-90deg, red, yellow); /* Standard syntax */  
}
```



# Gradientes CSS3

- Podemos usar más de dos colores

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(red, yellow, green); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(red, yellow, green); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(red, yellow, green); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(red, yellow, green); /* Standard syntax */  
}
```





# Gradientes CSS3

- Podemos repetir un patrón mediante *repeating-linear-gradient()*

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  /* Safari 5.1 to 6.0 */  
  background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /* Opera 11.1 to 12.0 */  
  background: -o-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /* Firefox 3.6 to 15 */  
  background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /* Standard syntax */  
  background: repeating-linear-gradient(red, yellow 10%, green 20%);  
}
```



# Gradientes CSS3

- Definido por su centro
- Sigue el siguiente patrón

*background: radial-gradient(<shape> <size> at <position>, start-color, ..., last-color);*

- Por defecto los valores son:
  - shape: ellipse
  - size: farthest-corner
  - position: center

Hay 4 posibles valores:

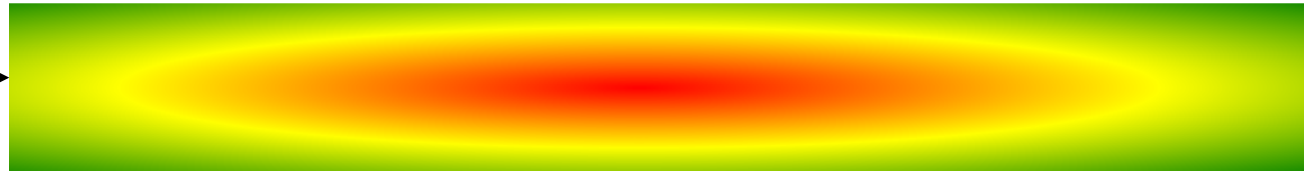
- *closest-side*
- *farthest-side*
- *closest-corner*
- *farthest-corner*

Más información en: <https://goo.gl/IdC2CS>

# Gradientes CSS3

- Tomando valores por defecto, tenemos esto

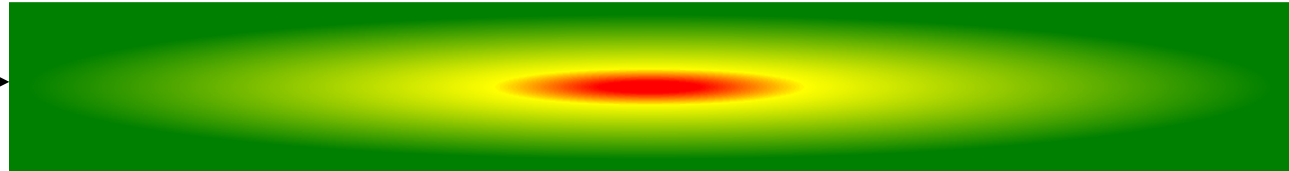
```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-radial-gradient(red, yellow, green); /* Safari 5.1 to 6.0 */  
  background: -o-radial-gradient(red, yellow, green); /* For Opera 11.6 to 12.0 */  
  /*  
  background: -moz-radial-gradient(red, yellow, green); /* For Firefox 3.6 to 15 */  
  /*  
  background: radial-gradient(red, yellow, green); /* Standard syntax */  
}
```



# Gradientes CSS3

- Podemos variar el espaciado entre colores

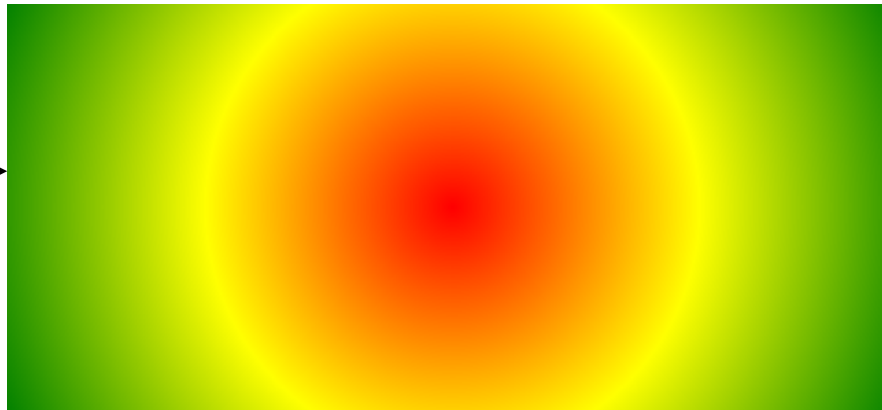
```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-radial-gradient(red 5%, yellow 15%, green 60%); /* Safari 5.1-6.0 */  
  background: -o-radial-gradient(red 5%, yellow 15%, green 60%); /* For Opera 11.6-12.0 */  
  background: -moz-radial-gradient(red 5%, yellow 15%, green 60%); /* For Firefox 3.6-15 */  
  background: radial-gradient(red 5%, yellow 15%, green 60%); /* Standard syntax */  
}
```



# Gradientes CSS3

- Podemos cambiar la forma

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-radial-gradient(circle, red, yellow, green); /* Safari */  
  background: -o-radial-gradient(circle, red, yellow, green); /* Opera 11.6 to 12.0 */  
  background: -moz-radial-gradient(circle, red, yellow, green); /* Firefox 3.6 to 15 */  
  background: radial-gradient(circle, red, yellow, green); /* Standard syntax */  
}
```



# Gradientes CSS3

## GRADIENT

- Cosas a tener en cuenta:
  - En la actualidad, el uso de gradientes en web está **casi en desuso**. Se prefieren colores sólidos.
  - Internet Explorer 9 e inferiores **no soportan** el uso de gradientes: <https://goo.gl/77yWxT>
  - La sintáxis para el soporte de gradientes en navegadores ha variado durante el tiempo, y es bastante lío: <https://goo.gl/1ajcgR>
  -