

# CSS3

# Temario

1. Introducción a CSS3
2. Selectores
3. Pseudo-elementos
4. Pseudo-clases
5. Modelo de caja
6. Tratamiento de textos
7. Colores
8. Unidades de medida
9. Identificadores extendidos
10. Important

# Temario

- 11.Box-shadow
- 12.Border-radius
- 13.Text-shadow
- 14.Transiciones
- 15.Transformaciones
- 16.Media Queries
- 17.Responsive design
- 18.Emulación de terminales

# CSS3

- CSS, permite definir los estilos de pintado de los elementos HTML.
- En CSS se puede definir herencia entre estilos, los elementos dentro de otros (subelementos o hijos), tienen los estilos que tienen sus padres.
- No todas las propiedades se heredan

<http://www.w3.org/TR/css-2010/#properties>

# CSS3

- Ante una colisión de estilos, se resuelve
  - Con el estilo que tenga “!important”.
  - El del elemento.
  - Según lo genérico del selector, los mas concretos son los mas importantes.
  - El definido en ultimo lugar.

# 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 {}
```

# 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 descendente con universal** -> Se puede mezclar el selector universal con el resto de selectores

```
div * p {}
```

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

```
div.cabecera {}
```

# Selectores CSS3

- **Selector de Id** -> Se aplica a los div, 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 {}
```



# Selectores CSS3

- **Selector de hermanos** -> Se aplica a los div, que tengan como atributo id, el valor cabecera

```
div ~ p {}
```

# Selectores CSS3

- **Selector de atributos (Nuevo en CSS3)**

- 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"]
```

# Selectores CSS3

- **Selector de atributos**

- Selecciona los div que tengan en el atributo class como uno de los valores cabecera

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

- Selecciona los div que tengan en el atributo lang una serie de valores separados por guion, empezando por es

```
div[lang |= "es"]
```

# Selectores CSS3

- **Selector de atributos (Nuevo en CSS3)**

- Selecciona todos los elementos que tengan en el class cabecera exactamente

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

- Selecciona todos los elementos que tengan al final del class "cabecera" exactamente

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

- Selecciona todos los elementos que contengan en el class "cabecera"

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

# Selectores CSS3

- Compatibilidad

	version	Gecko 1.7~1.9 [2]	Opera 7.5~8.5	Opera 9~10	Safari 1.0~4.0.x	iCab 3.0 [1]	IE 8.0 Win	IE 7.0 Win	IE 5.5-6.0 Win	IE 5.23 Mac	Konq 3.5 [1]
*	CSS 2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
E	CSS 1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
E F	CSS 1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
E, F (grouping) [18]	CSS 1	OK	OK	OK	OK	OK	OK	OK	OK	B [19]	OK
E > F	CSS 2	OK	OK	OK	OK	OK	OK	OK	N	OK	OK
E + F	CSS 2	OK	OK	OK	OK	OK	OK	B [23]	N	OK	OK
E ~ F	CSS 3	OK	N	OK	P [3]	OK	OK	OK	N	OK	OK
E[foo]	CSS 2	OK	OK	OK	OK	OK	OK	S	N	N	OK
E[foo="bar"]	CSS 2	OK	OK	OK	OK	OK	OK	S	N	N	OK
E[foo~="bar"]	CSS 2	OK	OK	OK	OK	OK	OK	S	N	N	OK
E[foo^="bar"]	CSS 3	OK	N	OK	OK	OK	OK	S	N	N	OK
E[foo\$="bar"]	CSS 3	OK	N	OK	OK	OK	OK	S	N	N	OK
E[foo*="bar"]	CSS 3	OK	N	OK	OK	OK	OK	S	N	N	OK
E[hreflang="en"]	CSS 2	OK	OK	OK	OK	OK	S	S	N	N	OK
E.class	CSS 1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
E#id	CSS 1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
E#id.class	CSS 2	OK	OK	OK	OK	OK	OK	B	B	B	OK
E:not(s)	CSS 3	OK	N	OK [28]	OK	N	N	N	N	N	OK
	version	Gecko 1.7~1.9 [2]	Opera 7.5~8.5	Opera 9~10	Safari 1.0~4.0.x	iCab 3.0 [1]	IE 8.0 Win	IE 7.0 Win	IE 5.5-6.0 Win	IE 5.23 Mac	Konq 3.5 [1]

# PSEUDO-Elementos CSS3

- **:first-line** -> Permite seleccionar la primera línea de un texto.

```
//Pone en mayúsculas la primera línea de cada párrafo  
p:first-line { text-transform: uppercase; }
```

- **:first-letter** -> Permite seleccionar la primera letra de un texto

```
//Pone en mayúsculas la primera letra de cada párrafo  
p:first-letter { text-transform: uppercase; }
```

# PSEUDO-Elementos CSS3

- **:before** y **:after** -> Se emplean conjuntamente con la propiedad CSS **content**, para añadir texto antes y después de un elemento

```
//Antepone a los títulos H1, el texto "Capítulo –"  
h1:before { content: "Capítulo - "; }
```

```
//Postpone "." a los párrafos.  
p:after { content: "."; }
```

# PSEUDO-Elementos CSS3

- **::selection** -> selecciona el texto que ha seleccionado un usuario con su ratón o teclado

```
h1::selection{}
```

- En CSS3, cambian la sintaxis, empleándose **::** en lugar de **:**



# PSEUDO-CLASES CSS3

- **:first-child** -> Selecciona el primer elemento de un tipo dado, que sea hijo de un elemento.

```
//Pone en rojo el primer párrafo, dentro de div  
div p:first-child {  
    color: red;  
}
```

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

# 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.

# PSEUDO-CLASES CSS3

- **:lang** -> Selecciona los elementos cuyo idioma se establece con la propiedad de la etiqueta **lang**.

```
//Pone en rojo los párrafos cuyo idioma sea español  
p:lang(es) { color: red; }
```

- El idioma de un elemento, se puede definir de varias formas:
  - Con la etiqueta **meta**, que afecta a toda la página.
  - Con el atributo **lang**, que afecta a la etiqueta donde se define y sus sub-etiquetas.

# Precedencia de Selectores

- Para resolver que estilo tiene prioridad sobre otros, el navegador hace un calculo sencillo, cuenta en el selector, los elementos que se encuentre de los siguientes tipos
  - Selector de ID (#)
  - Selector de clase o pseudoclase (.)
  - Selector de elemento html.
- Y luego compara, cumpliéndose que el que mas selectores de tipo **ID**, es el mas preferente, en caso de empate, el que mas selectores de **clase** o **pseudo-clase** y finalmente el que mas selectores de **HTML**.

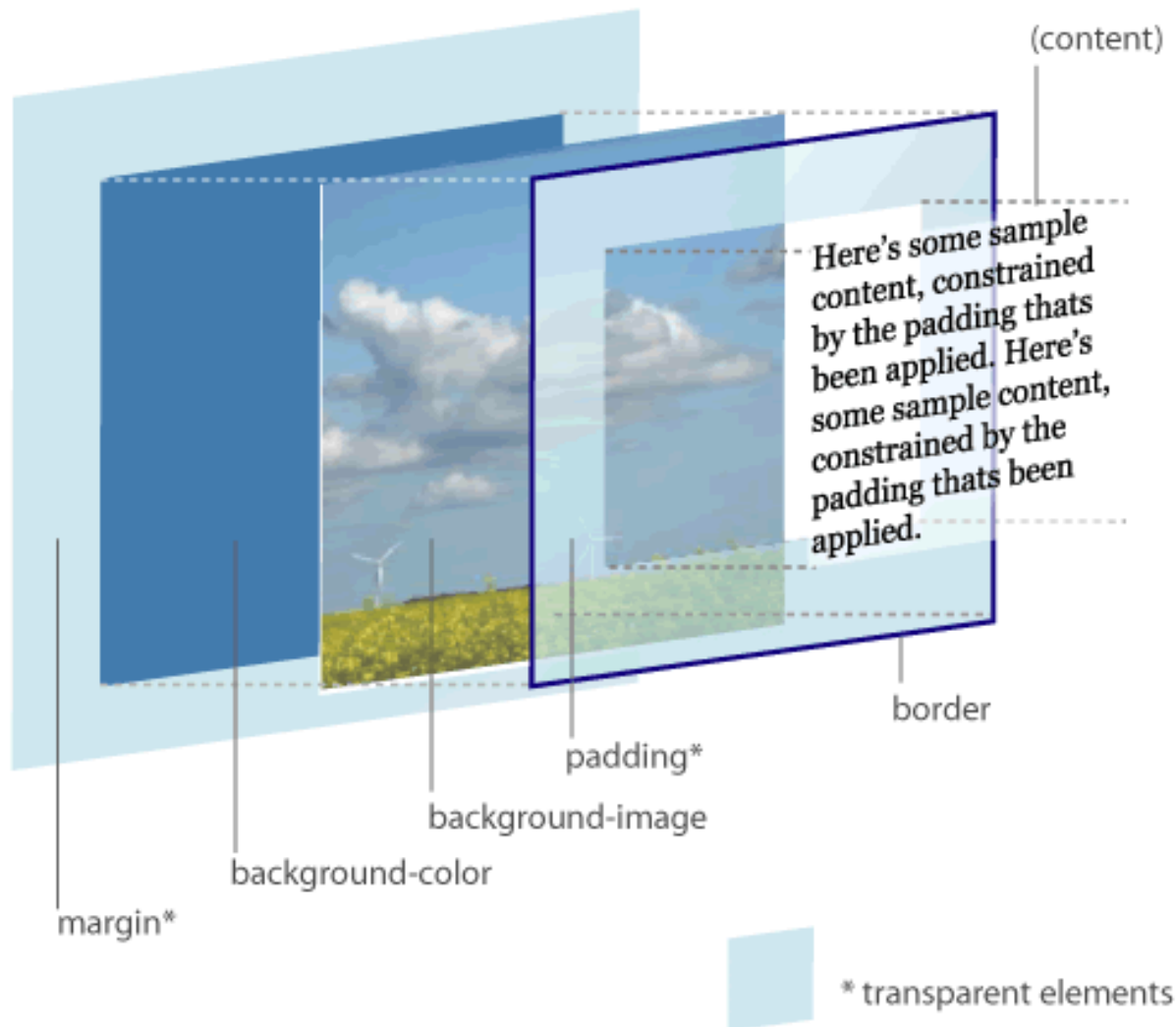
# Modelo de caja

- En CSS, todos los elementos se representan como cajas.

# Modelo de caja

- Partes de las cajas.
  - **Content:** Contenido HTML.
  - **Padding:** Espacio entre el contenido y el borde. (Relleno)
  - **Border:** Bordea el contenido y el padding
  - **Margin:** Espacio entre la caja y el resto de cajas adyacentes.
  - **background-color:** Color que se muestra por detrás del contenido y el padding.
  - **background-image:** Imagen que se muestra por detrás del contenido y el padding.

# Modelo de caja



# Modelo de caja

- El tamaño de un elemento, lo definen todas las partes de la caja.

```
div {  
    width: 300px;  
    padding-left: 50px;  
    padding-right: 50px;  
    margin-left: 30px;  
    margin-right: 30px;  
    border: 10px solid black;  
}
```

- Ancho total = 30px + 10px + 50px + 300px + 50px + 10px + 30px = 480 píxel



# Modelo de caja: DISPLAY

- Se puede organizar los elementos por tipo, siendo los tipos posibles.
  - **Línea:** Solo ocupan el espacio necesario para mostrar el contenido.
    - a, b, br, img, input, label, select, span, strong, sub, sup, textarea.
  - **Bloque:** Ocupan toda la línea.
    - div, form, h1, h2, h3, h4, h5, h6, hr, menu, ol, p, table, ul.

# Modelo de caja: DISPLAY

- La naturaleza de los elementos, se puede modificar con la propiedad **display**. los valores posibles son:

- Inline (elemento de línea)
- Block (elemento de bloque)
- list-item
- run-in
- inline-block
- Table
- inline-table
- table-row-group
- table-header-group
- table-footer-group
- table-row
- table-column-group
- table-columntable-cell
- table-caption
- None (No genera caja)
- inherit

# Modelo de caja: DISPLAY

- Ejemplo de elemento **inline**

```
[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.
```

- Ejemplo de varios elementos **inline** seguidos

```
[display: inline] Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem  
ipsum! Donec mollis nunc in leo! Vivamus fermentum!
```

# Modelo de caja: DISPLAY

- Ejemplo de elemento **block**

```
[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.
```

- Ejemplo de varios elementos **block** seguidos

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

```
[display: block] Lorem ipsum
```

```
[display: block] Donec mollis nunc in leo
```

```
[display: block] Vivamus fermentum
```

# Modelo de caja: DISPLAY

- Ejemplo de varios elementos **inline** seguidos, con un elemento **inline-block** entre medias

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
```

mollis nunc in leo. luctus viverra nisi. 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.

# Modelo de caja: OVERFLOW

- Otra propiedad interesante para el modelo de caja es la propiedad **overflow**, que indica si los elementos contenidos por este elemento, pueden o no salir de su espacio.
- Los posibles valores son
  - **visible**
  - **hidden**
  - **scroll**
  - **auto**

# Modelo de caja: OVERFLOW

- **visible:** el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.
- **hidden:** el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento. Si el elemento no tiene tamaño, se expande el elemento para que contenga todo.

# Modelo de caja: OVERFLOW

- **scroll**: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de scroll que permiten visualizar el resto del contenido.
- **auto**: el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.



# Modelo de caja: POSITION

- La ultima consideración a tener en cuenta sobre los elementos cuando se sitúan en un documento HTML, es su **posición**. Esta puede ser de varios tipos:
  - **Normal o estático**
  - **Relativo**
  - **Absoluto**
  - **Fijo**
  - **Flotante**

# Modelo de caja: POSITION

- **Normal o estático:**
  - Se considera:
    - Si es de bloque o línea
    - Los atributos **width** y **height**
    - Su contenido
  - Se ignoran
    - Los atributos **top**, **right**, **bottom** y **left**.

# Modelo de caja: POSITION

- **Relativo:**
  - Se mantiene el espacio que ocuparía el elemento sin desplazamiento
  - Se desplaza el elemento relativo a la posición que ocuparía de posicionarse de forma estática.
  - No afecta a otros elementos.

# Modelo de caja: POSITION

- **Relativo:**
  - Se consideran:
    - Los atributos **top**, **right**, **bottom** y **left**.
    - Los valores positivos van hacia abajo y la derecha.

# Modelo de caja: POSITION

- **Absoluto:**
  - El elemento sale del flujo de colocación relativo de la pagina, pasa a otro plano.
  - Sin desplazamiento, se coloca en el mismo lugar, pero no se reserva su posición, por lo que puede tapar otros elementos.
  - Al no reservar su posición, los demás elementos se ven afectados.

# Modelo de caja: POSITION

- **Absoluto:**
  - Se referencia al primer elemento del árbol con **position** distinta de **static** y se referencia a el.
  - Se consideran:
    - Los atributos **top**, **right**, **bottom** y **left**.

# Modelo de caja: POSITION

- **Fijo:**
  - Es como absoluto, pero siempre referenciado a la pantalla, no al primer elemento no **static**.
  - No desaparece al hacer **scroll** con el navegador.

# Modelo de caja: FLOAT

- **Flotante:**
  - Se establece con la propiedad **float**, no con la **position**.
  - Pasa a otro plano como el absoluto y el fijo, pero ese plano solo se puede referir a la derecha o izquierda.
  - Se van colocando las cajas una a la derecha de otra (o viceversa) hasta que no caben mas en la línea, con lo que las siguientes pasan a otra línea.



# Modelo de caja: FLOAT

- **Flotante:**
  - Pueden tapar otro contenido que este en el plano **fijo** o **absoluto**.
  - Se ignoran
    - Los atributos **top**, **right**, **bottom** y **left**.

# Tratamiento de textos













- CSS, permite trabajar los texto a través de una serie de propiedades que los afectan.
- **text-align** -> Establece la alineación del contenido del elemento (left | right | center | justify | inherit).
- **line-height** -> Establece la altura de la línea de texto (normal | numero | unidad de medida | porcentaje | inherit).

# Tratamiento de textos

- **text-decoration** -> Establece la decoración del texto (none | underline (subrayado) | overline (superayado) | line-through (tachado) | blink (parpadeante) | inherit).
- **text-transform** -> transforma el texto original (capitalize (primera letra de cada palabra en mayúsculas) | uppercase | lowercase | none | inherit)

# Tratamiento de textos

- **vertical-align** -> Alineación vertical del contenido (baseline | sub | super | top | text-top | middle | bottom | text-bottom | porcentaje | unidad de medida | inherit)

 Ayuda <b>baseline</b>	 Ayuda <b>sub</b>	 Ayuda <b>super</b>
 Ayuda <b>top</b>	 Ayuda <b>text-top</b>	 Ayuda <b>middle</b>
 Ayuda <b>bottom</b>	 Ayuda <b>text-bottom</b>	 <b>medida 1em</b> Ayuda
 <b>medida -1em</b> Ayuda	 <b>porcentaje 10%</b> Ayuda	 <b>porcentaje 80%</b> Ayuda

# Tratamiento de textos

- **text-indent** -> Tabula desde la izquierda la primera línea del texto (unidad de medida | porcentaje | inherit).
- **letter-spacing** -> Permite establecer el espacio entre las letras que forman las palabras del texto (normal | unidad de medida | inherit).
- **word-spacing** -> Permite establecer el espacio entre las palabras que forman el texto (normal | unidad de medida | inherit).

# Tratamiento de textos

- **white-space** -> Establece el tratamiento de los espacios en blanco del texto.
  - **Normal**.
  - **pre**: Se respetan espacios en blanco y nuevas líneas. Si la línea es larga, se sale del espacio asignado.
  - **nowrap**: Elimina espacios en blanco de mas y nuevas líneas. Si la línea es larga, se sale del espacio asignado.
  - **pre-wrap**: Se respetan espacios en blanco y nuevas líneas, ajustando cada línea al espacio asignado.

# Tratamiento de textos

- **white-space** -> Establece el tratamiento de los espacios en blanco del texto.
  - **pre-line**: Elimina espacios en blanco de mas y respeta nuevas líneas, ajustando cada línea al espacio asignado.
  - **Inherit**: Hereda del padre.

# COLORES

- Hay cuatro formas de definir los colores en CSS.
  - **RGB**: Red, Green y Blue.
  - **RGBA**: Red, Green, Blue y Alpha.
  - **HSL**: Hue, Saturation y Lightness (Tono, Saturación y Brillo)
  - **HSLA**: Hue, Saturation, Lightness y Alpha (Tono, Saturación, Brillo y Transparencia)



# COLORES

- La transparencia (Alpha) se expresa en tanto por 1, siendo 0 la máxima transparencia y 1 máxima opacidad.

```
body{  
    background-color:rgba(100,20,40,0.5);  
}
```

# COLORES HSL

- El tono puede tomar valores entre 0 y 360, siendo estos una referencia
  - 0 y 360 rojo.
  - 120 verde.
  - 240 azul.
- Los otros dos valores, pueden tomar valores en tanto por ciento.



```
body{  
    background-color:rgba(100,20,40,0.5);  
}
```

# COLORES HSL

- Un ejemplo puede ser.

```
body{  
    background-color: hsl(360,100%,20%);  
}
```

- También se puede añadir un canal alpha (transparencia), en este caso en tanto por ciento.

```
body{  
    hsla(300,130%,65%,10%);  
}
```

# UNIDADES DE MEDIDA

- Se emplean, para definir la altura, anchura y márgenes de los elementos y el tamaño de las letras del texto.
- Se indican con un valor numérico, seguido de unas unidad de medida.
- Existen dos grupos de unidades.
  - Absolutas.
  - Relativas.

# UNIDADES DE MEDIDA

- Absolutas
  - **in**, pulgadas (2.54cm).
  - **cm**, centímetros.
  - **mm**, milímetros.
  - **pt**, puntos ( $1\text{pulgada} / 72 = 0.35\text{mm}$ )
  - **pc**, picas ( $12\text{ puntos} = 4.23\text{ mm}$ )

# UNIDADES DE MEDIDA

- Relativas
  - **em**, relativa al tamaño de letra del elemento.
  - **ex**, relativa a la altura de la letra x minúscula de la tipografía empleada.
  - **px**, pixel, relativa a la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

# UNIDADES DE MEDIDA

- Normalmente los navegadores, emplean por defecto un tamaño de textos de 16px, por lo que una medida de 1em, por defecto equivaldrá a 16px.
- Por tanto cuando se emplea **em**, es importante conocer el valor establecido para la propiedad **font-size**.

```
/*Margenes de 16px, la mitad del tamaño de letra establecido*/  
p { font-size: 32px; margin: 0.5em; }
```

# 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-css3 /* Oficial */  
-moz-propiedad-css3 /* Mozilla Firefox */  
-o-propiedad-css3 /* Opera */  
-webkit-propiedad-css3 /* Chrome, Safari */
```



# ! IMPORTANT

- Es habitual que en la creación de los estilos CSS, se necesite hacer que un estilo concreto tenga mas peso que otro estilo que lo esta pisando, para ello, se ha de añadir la expresión **!important** a la propiedad que deseemos sea la que se aplique.

```
section#Barralidiomas{  
    display: block !important;  
}
```

# BOX-SHADOW

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

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

- Algún ejemplo

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



# BOX-SHADOW

# CSS3 Box-shadow - <b>Candidate Recommendation</b>									
<i>Method of displaying an inner or outer shadow effect to elements</i>									
Resources: <a href="#">MDN article</a> <a href="#">Live editor</a> <a href="#">Demo of various effects</a> <a href="#">Visual browser comparison</a>									
<a href="#">Information page</a>									
<a href="#">Show all versions</a>	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
								10.0	2.1 -webkit-
	6.0	3.6 -moz-				3.2 -webkit-		11.0	2.2 -webkit-
	7.0	9.0				4.0-4.1 -webkit-		11.1	2.3 -webkit-
	8.0	10.0	17.0	5.0 -webkit-		4.2-4.3 -webkit-		11.5	3.0 -webkit-
Current	9.0	11.0	18.0	5.1	11.6	5.0	5.0-6.0	12.0	4.0 -webkit-
Near future	10.0	12.0	19.0	5.2	12.0				
Farther future		13.0	20.0						

*Usage stats:	Global
Support:	70.91%
Partial support:	2.08%
Total:	72.99%

# Border-radius

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

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

- Algún ejemplo

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



# Border-radius

- Se pueden definir también las siguientes propiedades
  - border-bottom-left-radius,
  - border-bottom-right-radius,
  - border-top-left-radius
  - border-top-right-radius

# Border-radius

# CSS3 Border-radius (rounded corners) - <b>Candidate Recommendation</b>						*_Usage stats:		Global	
<i>Method of making the border corners round</i>						Support:		73.33%	
Resources: <a href="#">Border-radius CSS Generator</a> <a href="#">Detailed compliance table</a> <a href="#">Information page</a>						Partial support:		0.08%	
						Total:		73.41%	
<a href="#">Show all versions</a>	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
								10.0	2.1 <small>-webkit-</small>
	6.0	3.6 <small>-moz-</small>				3.2 <small>-webkit-</small>		11.0	2.2
	7.0	9.0				4.0-4.1		11.1	2.3
	8.0	10.0	17.0	5.0		4.2-4.3		11.5	3.0
Current	9.0	11.0	18.0	5.1	11.6	5.0	5.0-6.0	12.0	4.0
Near future	10.0	12.0	19.0	5.2	12.0				
Farther future		13.0	20.0						

# TEXT-SHADOW

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

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

- Algunos ejemplos

```
text-shadow: 10px 10px 5px rgba(0,0,0,.3);
```

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

**TITULO**

**TITULO**

# TEXT-SHADOW

## # CSS3 Text-shadow - **Working Draft**

*Method of applying one or more shadow or blur effects to text*

Resources: [Mozilla hacks article](#) [Live editor](#) [Information page](#)

### \*Usage stats:

### Global

Support:	59.81%
Partial support:	2.01%
Total:	61.82%

<a href="#">Show all versions</a>	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
								10.0	2.1
	6.0	3.6				3.2		11.0	2.2
	7.0	9.0				4.0-4.1		11.1	2.3
	8.0	10.0	17.0	5.0		4.2-4.3		11.5	3.0
Current	9.0	11.0	18.0	5.1	11.6	5.0	5.0-6.0	12.0	4.0
Near future	10.0	12.0	19.0	5.2	12.0				
Farther future		13.0	20.0						



# TRANSITION

- Es una propiedad en fase de borrador, por lo que se necesita definirla con los identificadores extendidos.
- Normalmente suele estar asociado a los eventos del ratón.
- Se puede dividir en varias propiedades:

```
transition-property: font-size;  
transition-duration: 1s;  
transition-delay: 0.2s;  
transition-timing-function: ease;
```

# TRANSITION

- La lista de propiedades, a las que se puede aplicar transiciones esta en
  - <http://www.w3.org/TR/css3-transitions/#animatable-types>
- Los posibles valores de la propiedad **transition-timing-function** son
  - ease
  - linear
  - ease-in
  - ease-out
  - ease-in-out
  - cubic-bezier

# TRANSITION

- Si las propiedades afectadas por las transiciones son alteradas por javascript, también se aplica la transición.
- Se puede definir un listener, para realizar alguna operación al terminar la transición, el evento es
  - `transitionend`
  - `webkitTransitionEnd`

# TRANSFORMATION

- Se pueden aplicar transformaciones a los elementos, modificando el aspecto de la caja.
- Las transformaciones disponibles son
  - **skew(angulo)**: Inclina la caja el ángulo.
  - **scale(x, y)**: Escala la caja en X e Y.
  - **rotate(angulo)**: Rota la caja un ángulo en sentido horario.
  - **translate(x, y)**: Traslada la caja en X e Y.

# TRANSFORMACIONES

# CSS3 Transforms - Working Draft									
								*Usage stats:	Global
								Support:	72.99%
Method of transforming an element including rotating, scaling, etc.									
Resources: <a href="#">Live editor</a> <a href="#">MDN article</a> <a href="#">Workaround script for IE</a> <a href="#">Information page</a> <a href="#">Converter for IE</a>									
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
								10.0	2.1 -webkit-
	6.0	3.6 -moz-				3.2 -webkit-		11.0	2.2 -webkit-
	7.0	9.0 -moz-				4.0-4.1 -webkit-		11.1	2.3 -webkit-
	8.0	10.0 -moz-	17.0 -webkit-	5.0 -webkit-		4.2-4.3 -webkit-		11.5	3.0 -webkit-
Current	9.0 -ms-	11.0 -moz-	18.0 -webkit-	5.1 -webkit-	11.6 -o-	5.0 -webkit-	5.0-6.0	12.0	4.0 -webkit-
Near future	10.0 -ms-	12.0 -moz-	19.0 -webkit-	5.2 -webkit-	12.0 -o-				
Farther future		13.0 -moz-	20.0 -webkit-						

# Degradados

- Se pueden aplicar degradados a los fondos, con las funciones
  - linear-gradient
  - -prefix-linear-gradient

```
background: -prefix-linear-gradient(left, red, orange, yellow, green, blue);  
background: linear-gradient(to right, red, orange, yellow, green, blue);
```

- radial-gradient

```
radial-gradient(red 5%, yellow 25%, #1E90FF 50%);
```

# Degradados

- También, se pueden repetir degradados, para crear cuadrículas o círculos concéntricos
  - repeating-linear-gradient

```
background: repeating-linear-gradient(-45deg, red, red 5px, white 5px, white 10px);
```

- repeating-radial-gradient

```
background: repeating-radial-gradient(black, black 5px, white 5px, white 10px);
```

# Declaración de fuentes

- Hasta CSS3, las fuentes a emplear, eran las instaladas en el equipo del cliente, a partir de ahora, se pueden definir fuentes en línea.

```
@font-face {  
  [font-family: <family-name>;]?  
  [src: [ <uri> [format(<string>#)]? | <font-face-name> ]#;]?  
  [unicode-range: <urange>#;]?  
  [font-variant: <font-variant>;]?  
  [font-feature-settings: normal | <feature-tag-value>#;]?  
  [font-stretch: <font-stretch>;]?  
  [font-weight: <weight>;]  
  [font-style: <style>;]  
}
```



# Declaración de fuentes

- Las propiedades a definir en la fuente son:
  - **<family-name>**: especifica el nombre de la fuente que se utilizará.
  - **<uri>**: URL para de la ubicación remota de la fuente, o nombre de la fuente en el ordenador del autor ("FontName")
  - **<font-variant>**: variante de una fuente anterior.
  - **<font-stretch>**: especifica la anchura de la fuente.
  - **<weight>**: especifica el peso de la fuente (light, normal, bold ...)
  - **<style>**: especifica el estilo de la fuente (normal, italic o oblique)

# Multi column

- Se permite la definición de la forma de relleno de un elemento, en columnas.

```
div { column-count:2; }
```

- Que se vería de la siguiente forma

Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna aliqua. Ut  
enim ad minim veniam, quis nostrud  
exercitation ullamco laboris nisi ut aliquip ex ea

commodo consequat. Duis aute irure dolor in  
reprehenderit in voluptate velit esse cillum  
dolore eu fugiat nulla pariatur. Excepteur sint  
occaecat cupidatat non proident, sunt in culpa  
qui officia deserunt mollit anim id est laborum

# Multi column

- También se puede definir el ancho que deberá ocupar una columna y que adapte el contenido

```
div { column-width:10em; }
```

- Que se vería de la siguiente forma

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute	irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint	occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
--	--	--	--

# Multi columna

- Y también aplicar a un componente dentro del aquellos que se irán formateados en columnas, para que aparezca ocupando todo el espacio

```
column-span: all;  
text-align: center;
```

- Que se vería de la siguiente forma

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit,

quis nostrud exercitation  
ullamco laboris nisi ut aliquip  
ex ea commodo consequat.  
Duis aute irure dolor in

sed do eiusmod tempor  
incididunt ut labore et dolore

Loren ipsum

reprehenderit in voluptate  
velit esse cillum dolore eu  
fugiat nulla pariatur.  
Excepteur sint occaecat

magna aliqua. Ut enim ad  
minim veniam,

cupidatat non proident, sunt in  
culpa qui officia deserunt  
mollit anim id est laborum

# Multi column

- Y definir el espacio entre columnas con **column-gap**, o establecer un trazo divisorio entre columnas con **column-rule**

```
column-count: 3;  
column-rule: 1px dotted black;
```

- Que se vería de la siguiente forma

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit,  
sed do eiusmod tempor  
incididunt ut labore et dolore  
magna aliqua. Ut enim ad  
minim veniam, quis nostrud

exercitation ullamco laboris  
nisi ut aliquip ex ea commodo  
consequat. Duis aute irure  
dolor in reprehenderit in  
voluptate velit esse cillum  
dolore eu fugiat nulla pariatur.

Excepteur sint occaecat  
cupidatat non proident, sunt in  
culpa qui officia deserunt  
mollit anim id est laborum

# MEDIA QUERIES

- Un nuevo elemento de CSS3, que permite aplicar estilos en una pagina dependiendo de determinadas variables del entorno.
- La sintaxis de una media query, es

```
@media <media type> and (<propiedad>: <valor>) {  
    /* Aquí van las reglas CSS necesarias */  
}
```

- Por ejemplo

```
@media screen and (max-width: 320px) {  
    /* Aquí van las reglas CSS necesarias */  
}
```

# MEDIA QUERIES

- También puede hacerse sobre la etiqueta **link** que importa un CSS

```
<link rel="stylesheet"  
      media="media screen and (max-width: 480px)"  
      type="text/css" href="PantallasPequeñas-320px.css" />
```

```
@import url("style.css") screen and (max-width: 480px);
```

# MEDIA QUERIES

- **Media Types**
  - **screen**: Todos los visores que pagina que tienen pantalla.
  - **projection**: Los proyectores.
  - **print**: Las impresoras.
  - **all**: Incluye todos los dispositivos.



# MEDIA QUERIES

- **Propiedades**

- **Min-width:** Permite aplicar un estilo sobre un elemento cuyo ancho menos es el indicado (empleado para definir estilos para pantallas mas grandes del valor indicado)
- **Max-width:** Permite aplicar un estilo sobre un elemento cuyo ancho mayor es el indicado (empleado para definir estilos para pantallas con el ancho menor al indicado)
- **orientation:** Permite consultar la orientación de la pantalla, sus valores son: **portrait** (vertical) y **landscape** (apaisado u horizontal).

# MEDIA QUERIES

- **Propiedades**
  - **aspect-ratio:** Relación de aspecto de la pantalla, expresada como 16:9.
  - **resolution:** Resolución de la pantalla.

# MEDIA QUERIES

Feature	Value	Min/Max	Description
width	Length	Yes	Width of display area
height	Length	Yes	Height of display area
device-width	Length	Yes	Width of device
device-height	Length	Yes	Height of device
orientation	portrait orlandscape	No	Orientation of device
aspect-ratio	Ratio (w/h)	Yes	Ratio of width to height, expressed as two integers separated by a slash (e.g., 16/9)
device-aspect-ratio	Ratio (w/h)	Yes	Ratio of device-width to device-height
color	Integer	Yes	Number of bits per color component (if not color, the value is 0)
color-index	Integer	Yes	Number of entries in the output device's color lookup table
monochrome	Integer	Yes	Number of bits per pixel in the monochrome frame buffer (if not monochrome, the value is 0)
resolution	Resolution	Yes	Density of pixels of output device, express as integer followed by dpi(dots per inch) or dpcm (dots per centimeter)
scan	progressiveor interlace	No	Scanning process used by TV devices
grid	0 or 1	No	If set to 1, the device is grid-based, such as a teletype terminal or phone display with only one fixed font (all other devices are 0)

# MEDIA QUERIES

- Los navegadores que soportan **Media Queries**, son todos los modernos, esto excluye a IE6, IE7 e IE8.
- Para integrar la funcionalidad que proporcionan las **Media Queries** en esos navegadores, se pueden emplear algunas librerías de **javascript** como
  - Modernizr.
  - Respond.

# RESPONSIVE DESIGN

- También conocido como **Diseño Web adaptable**, es una técnica de diseño centrada en el usuario, que permite que este tenga una navegación placentera por las paginas web de un sitio, independientemente del tamaño de la pantalla.
- Las técnicas, se basan en HTML y principalmente CSS, permitiendo que **la web se adapte al ancho de pantalla**, respondiendo ante los cambios del tamaño la misma.

# RESPONSIVE DESIGN

- Dado el gran número de medios y navegadores existentes no es factible crear una versión de nuestra web para cada uno de ellos, por lo que necesitamos un sistema para que nuestra web se adapte al contexto en el que está siendo vista, optimizándose de manera automática.

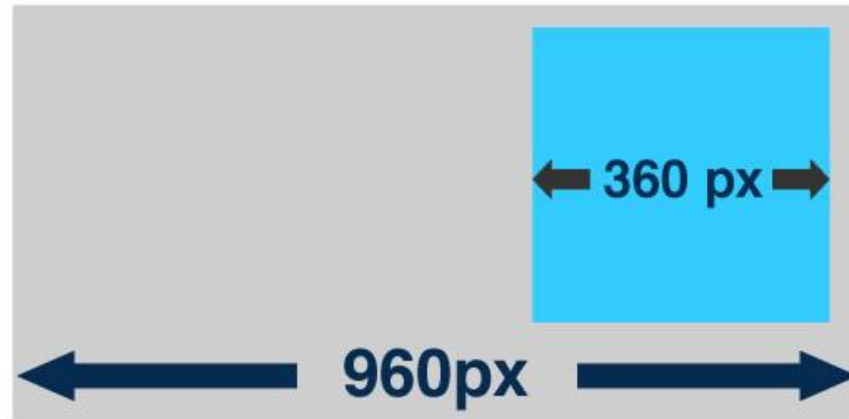
# RESPONSIVE DESIGN

- El **primer concepto** en el que se apoya el Diseño Web Adaptable es en abandonar los anchos fijos de nuestra web.
- Los anchos deberán ser fluidos.
- En lugar de diseñar nuestra web basándonos en valores fijos (por ejemplo width: 960px), el diseño fluido está pensado en términos de proporciones.

# RESPONSIVE DESIGN

- Para saber el ancho de un elemento tendremos que dividir el ancho inicial del mismo entre el ancho del elemento padre.

Se cambia `width="360px"` por `width="37,5%"`



$$360/960 = 37,5\%$$



# RESPONSIVE DESIGN

- También se han de adaptar los tamaños de las fuentes siguiendo la misma técnica.

Si el tamaño general es de 16px, y tenemos un título de 22px, su nuevo tamaño será de  $22/16 = 1.375\text{em}$

- Para las imágenes y elementos de ancho fijo, la misma técnica, no se puede emplear, y lo que hay que plasmar, es que la imagen adquiriera el 100% del tamaño del padre, modificando el tamaño del padre.

```
img, video, object { max-width:100%; } //Navegadores modernos  
img, video, object {width:100%; } //IE6 e IE7
```

# RESPONSIVE DESIGN

- La aproximación de la rejilla flexible, define que han de emplearse

- **Fuentes flexibles:** Basar las medidas en em.

$\text{<Valor en em > = <tamaño deseado> / <tamaño definido por el navegador (16px)>}$

- **Contenedores flexibles:**

- Como medida general, suele adoptarse como ancho base un 90% del área visible de la pantalla,

- **Márgenes flexibles:**

- Se define en porcentaje, teniendo en cuenta que el tamaño de referencia es el del contenedor.

- **Rellenos flexibles:**

- **Imágenes flexibles:**

# RESPONSIVE DESIGN

- Con este concepto, no es suficiente, ya que una pagina pensada para pantallas grandes, intentara abarcar todo lo posible la pantalla, para incluir los contenidos, y no tener que introducir scroll.
- Llegado ese caso, si tuviéramos un Layout (distribución de espacios en la pantalla) con 3 columnas, aunque podamos minimizar el ancho de las mismas y sus contenidos, acabaríamos presentando algo tan pequeño al usuario de un **smartphone** o **tablet**, que lo forzaría a hacer zoom.

# RESPONSIVE DESIGN

- El **segundo concepto** en el que se apoya el Diseño Web Adaptable son las **Media Queries** de CSS3.
- Este componente de CSS3, permite obtener preguntando al navegador, cuales son sus características, y según cuales sean, se aplicara un **layout** u otro, en realidad, se pueden aplicar a hojas de estilos completas.

```
@media screen and (max-width: 320px) {  
    /* Aquí van las reglas CSS necesarias */  
}
```

# RESPONSIVE DESIGN

- Se recomienda que los anchos de pantalla contemplados sean
  - 320px
  - 480px
  - 600px
  - 768px
  - 900px
  - 1200px

# RESPONSIVE DESIGN

- El tercer y ultimo paso, es indicar al navegador que emplee el ancho del medio (el monitor, smartphone, tablet,...) como ancho inicial, anulando la escala.

```
<meta name="viewport"  
      content="width=device-width;  
              initial-scale=1.0;  
              maximum-scale=1.0;  
              user-scalable=0;" />
```

# EMULADORES DISTINTOS TERMINALES

- El navegador de Google **Chrome**, desde hace unas cuantas versiones, incluye un emulador para poder visualizar las paginas desde el propio navegador, emulando que se trata de distintos terminales móviles.
- Para activarlo, hay que abrir las opciones de desarrollador (F12).
- Seleccionar la opción “**Show console**”.



# EMULADORES DISTINTOS

## TERMINALES

- En la sección de **emulation**, seleccionar el **device** y **emulate**.

Console	Search	Emulation	Rendering
Device	<input type="text" value="Amazon Amazon Kindle Fire HD 7"/>		
Screen	<input type="button" value="Emulate"/> <input type="button" value="Reset"/>		
User Agent	Viewport: 1280 × 800, devicePixelRatio = 1.5		
Sensors	User agent: Mozilla/5.0 (Linux; U; Android 2.3.4; en-us; Kindle Fire HD Build/G		



# EMULADORES DISTINTOS TERMINALES

- La anterior prueba, esta bien para probar como queda la pagina visualizada en **Chrome** desde distintos terminales, pero no deja de ser **Chrome**, es posible que se quiera probar en otros navegadores, para ello, se puede crear una pagina de **tester**, que incluya todas las resoluciones, empleando **iframes**.

# EMULADORES DISTINTOS TERMINALES

- Tester

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Responsive Design Testing</title>
  <style>
    body { margin: 20px; font-family: sans-serif; overflow-x: scroll; }
    .wrapper { width: 6000px; }
    .frame { float: left; }
    h2 { margin: 0 0 5px 0; }
    iframe { margin: 0 20px 20px 0; border: 1px solid #666; }
  </style>
</head>
```

# EMULADORES DISTINTOS TERMINALES

- Tester

```
<body>
  <div class="wrapper">
    <div class="frame">
      <h2>
        320 &times; 480
        <small>(mobile)</small>
      </h2>
      <iframe src="index.html"
        sandbox="allow-same-origin allow-forms"
        seamless
        width="320" height="480"></iframe>
    </div>
```

.  
.

# EMULADORES DISTINTOS TERMINALES

- Tester

```
.  
  
    <div class="frame">  
        <h2>  
            480 &times; 640  
            <small>(small tablet)</small>  
        </h2>  
        <iframe src="index.html"  
            sandbox="allow-same-origin allow-forms"  
            seamless  
            width="480" height="640"></iframe>  
    </div>  
</div>  
</body>  
</html>
```