

CSS



Aplicaciones Web

CSS 3

Conceptos Básicos

Oscar Fernando Aristizábal Cardona
Ingeniero de Sistemas y Telecomunicaciones
ofaczero@gmail.com

Introducción CSS 3

- ❖ CSS3 es el último estándar de CSS.
- ❖ CSS3 es compatible completamente con versiones anteriores de CSS.
- ❖ La especificación CSS3 está todavía en desarrollo por la W3C.
- ❖ Muchas de las nuevas propiedades de CSS3 se han implementado en los navegadores modernos.

Módulos CSS 3

Algunos de los módulos de CSS3 más importantes son:

- ❖ Selectores
- ❖ Modelo de caja
- ❖ Fondos y Bordes
- ❖ Valores de imagen y reemplazo de contenidos
- ❖ Efectos de texto
- ❖ 2D/3D Transformaciones
- ❖ Animaciones
- ❖ Diseño de columna múltiple
- ❖ Interfaz de usuario

1. Nuevos Selectores CSS 3

Selector	Ejemplo	Descripción
element1~element2	p~ul	Selecciona todos los elementos precedidas por un elemento <p>
[attribute^=value]	a[src^="https"]	Selecciona todos los elems <a> cuyo valor del atributo src comienza con "https"
[attribute\$=value]	a[src\$=".pdf"]	Selecciona todos los elems <a> cuyo valor del atributo src termina con ". Pdf"
[attribute*=value]	a[src*="google"]	Selecciona todos los elems <a> cuyo valor del atributo src contiene la cadena "google"
:first-of-type	p:first-of-type	Selecciona el primer elemento <p> de su padre
:last-of-type	p:last-of-type	Selecciona el último elemento <p> de su padre
:only-of-type	p:only-of-type	Selecciona el único elemento <p> de su padre
:only-child	p:only-child	Selecciona el único elemento hijo <p> de su padre
:nth-child(n)	p:nth-child(2)	Selecciona el segundo elemento hijo <p> de su padre
:nth-last-child(n)	p:nth-last-child(2)	Selecciona el segundo elem hijo <p> , contando desde el último elem hijo
:nth-of-type(n)	p:nth-of-type(2)	Selecciona el segundo elemento hijo <p> de su padre
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selecciona el segundo elem hijo <p> , contando desde el último elem hijo

2. Nuevos Selectores CSS 3

Selector	Ejemplo	Descripción
:last-child	p:last-child	Selecciona el último elemento hijo <p> de su padre
:root	:root	Selecciona el elemento raíz del documento
:empty	p:empty	Selecciona los elementos <p> que no tiene hijos (incluidos los nodos de texto)
:target	#noticias:target	Selecciona el elemento #noticias actualmente activo (Se hace click en el enlace que contiene ese nombre de ancla)
:enabled	input:enabled	Selecciona todos los elementos <input> habilitados
:disabled	input:disabled	Selecciona todos los elementos <input> inhabilitados
:checked	input:checked	Selecciona todos los elementos <input> marcados
:not(selector)	:not(p)	Selecciona cada elemento que no es un elemento <p>
::selection	::selection	Selecciona la porción de un elemento que es seleccionado por un usuario

1. Bordos CSS 3

Con CSS3, puede crear bordes redondeados, añadir sombra a las cajas y utilizar una imagen como borde sin necesidad de utilizar un programa de edición de imágenes como Fireworks y Photoshop.

Propiedades de borde:

- border-radius
- box-shadow
- border-image

2. Bordos CSS 3

border-radius:

Sintaxis: *border-radius: 1-4 length/% / 1-4 length/%;*

Los cuatro valores para cada uno de los radios se dan en el orden top-left, top-right, bottom-right, bottom-left.

Ejemplo:

```
section {  
    border: 2px solid;  
    border-radius: 20px; /*border-top-left-radius: 10px*/  
}
```

3. Bordos CSS 3

box-shadow:

Sintaxis: *box-shadow: h-shadow v-shadow blur spread color inset;*

- h-shadow:** “Valor Requerido”, La posición de la sombra horizontal. Valores negativos permitidos.
- v-shadow:** “Valor Requerido”, La posición de la sombra vertical. Valores negativos permitidos.
- blur:** “Valor Opcional”, Distancia de desenfoque.
- spread:** “Valor Opcional”, Tamaño de la sombra.
- color:** “Valor Opcional”, Color de la sombra.
- inset:** “Valor Opcional”, Cambia de sombra exterior a sombra interior.

Ejemplo:

```
section {  
    box-shadow: 10px 10px 5px #666;  
}
```

4. Bordes CSS 3

border-image:

Sintaxis: *border-image: source slice width outset repeat;*

- source:** La ruta a la imagen que se utilizará como imagen de borde.
- slice:** El desplazamiento hacia el interior.
- width:** El ancho de la imagen del borde.
- outset:** Cantidad de la imagen que se extiende más allá del borde de la caja.
- repeat:** Si la imagen se debe repetir, **round** o **stretch**.

Ejemplo:

```
section {  
    border-image:url(imgs/borde.png) 40 40 round;  
}
```

1. Fondos CSS 3

CSS3 contiene nuevas propiedades para los fondos, que permiten un mayor control del elemento **background**.

Propiedades de fondo:

- background-size
- background-origin
- background-clip

2. Fondos CSS 3

background-size:

La propiedad background-size especifica el tamaño de la imagen de fondo.

Antes de CSS 3, el tamaño de la imagen de fondo se determinaba por el tamaño real de la imagen.

Se puede especificar el tamaño en píxeles o en porcentajes. Si se especifica el tamaño en porcentaje, el tamaño es relativo al ancho y al alto del elemento padre.

3. Fondos CSS 3

background-size:

Sintaxis: *background-size: value;*

- auto:** Valor por defecto.
- length:** Valor en pixeles, el primer valor especifica el ancho y el segundo el alto.
- percentage:** Valor en porcentaje, el primer valor especifica el ancho y el segundo el alto.
- cover:** Escala la imagen de fondo al tamaño total del ancho.
- contain:** Escala la imagen de fondo al tamaño exacto del elemento contenedor (ancho y alto).

Ejemplo:

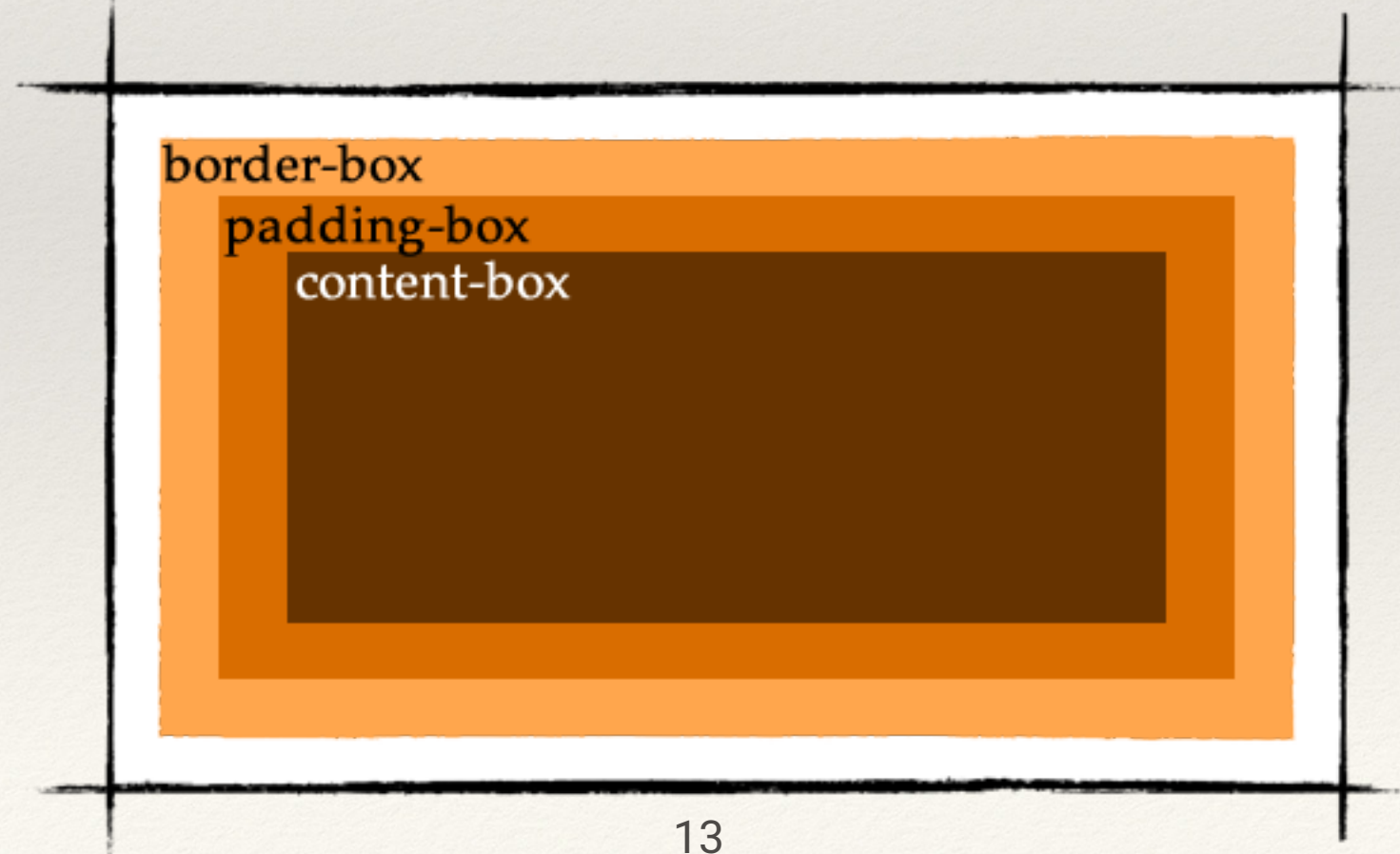
```
section {  
  background: url(fondo.png);  
  background-size: 100px 80px;  
  background-repeat: no-repeat;  
}
```


4. Fondos CSS 3

background-origin:

Especifica el área de posicionamiento de las imágenes de fondo.

La imagen de fondo se puede colocar dentro del contenido de la caja, el relleno de la caja o zona de la frontera de la caja.



5. Fondos CSS 3

background-origin:

Sintaxis: *background-origin: padding-box/border-box/content-box;*

- padding-box:** Valor por defecto, la imagen de fondo se coloca respecto a la caja de relleno.
- border-box:** La imagen de fondo se coloca en relación con el cuadro de frontera.
- content-box:** La imagen de fondo se posiciona respecto a la caja de contenido.

Ejemplo:

```
section {  
    background: url(fondo.png);  
    background-repeat: no-repeat;  
    background-origin: content-box;  
}
```

6. Fondos CSS 3

background-clip:

Especifica el área de la pintura del fondo

Sintaxis: *background-clip: border-box/padding-box/content-box;*

- border-box:** El fondo se recorta a la caja de la frontera.
- padding-box:** El fondo se recorta a la caja de relleno.
- content-box:** El fondo se recorta a la caja de contenido.

Ejemplo:

```
section {  
    padding: 30px;  
    background-color: #f00;  
    background-clip: padding-box;  
}
```

7. Fondos CSS 3

CSS 3 permite utilizar multiples imágenes de fondo.

Sintaxis: *background: url('bg1.png'), url('bg2.png'), url('bgn.png');*

Ejemplo:

```
section {  
    background: url(fondo1.png),url(fondo2.png);  
    background-size:100% 100%;  
    background-repeat:no-repeat;  
}
```

1. Degradados CSS 3

Los degradados CSS3 permiten mostrar transiciones suaves entre dos o más colores especificados.

Anteriormente, se tenía que utilizar imágenes para estos efectos. Mediante el uso de gradientes CSS3 se puede reducir el tiempo de descarga.

CSS3 define dos tipos de gradientes:

- linear-gradient
- radial-gradient

2. Degradados CSS 3

linear-gradient:

Para crear un degradado lineal debe definir al menos dos paradas de color. También puede establecer un punto de partida y una dirección (o ángulo), junto con el efecto de degradado.

Sintaxis: *background: linear-gradient(direction, color-stop1, color-stop2, ...);*

Ejemplo:

```
section#degradadolineal {  
    background: -webkit-linear-gradient(red, blue); /* Safari */  
    background: -moz-linear-gradient(red, blue); /* Firefox */  
    background: -o-linear-gradient(red, blue); /* Opera */  
    background: linear-gradient(red, blue);  
}
```

3. Degradados CSS 3

- Degradado lineal - De arriba a abajo (esto es por defecto)

background: linear-gradient(<top><bottom>, color1, color2);

- Degradado lineal - de izquierda a derecha

background: linear-gradient(<right><left>, color1, color2);

- Degradado lineal - Diagonal

background: linear-gradient(<left top><right bottom>, color1, color2);

- Usando Ángulos

background: linear-gradient(<angle>, color1, color2);

- Usando varios colores

background: linear-gradient(color1, color2, color3...colorn);

- Usando transparencias

background: linear-gradient(rgba1, rgba1);

4. Degradados CSS 3

- Degradado lineal - Repetición

background: repeating-linear-gradient(color1, color2 10%, color3 20%);

- Degradado Radial - Paradas de Color uniformes

background: radial-gradient(color1, color2, color3);

- Degradado Radial - Paradas de Color con diferente espaciado

background: radial-gradient(color1 5%, color2 15%, color3 60%);

- Degradado Radial - Formas

background: radial-gradient(circle, color1, color2, color3);

- Degradado Radial - Palabras Claves

background: radial-gradient(60% 55%, closest-side,color1,color2,color3,color4);

(closest-side, farthest-side, closest-corner, farthest-corner)

- Degradado Radial - Repetición

background: repeating-radial-gradient(color1, color2 10%, color3 15%);

1. Efectos Texto CSS 3

CSS3 define dos propiedades de texto:

- text-shadow
- word-wrap

2. Efectos Texto CSS 3

text-shadow:

En CSS 3, la propiedad **text-shadow** aplica sombra al texto.

Sintaxis: *text-shadow: h-shadow v-shadow blur color;*

Ejemplo:

```
h1 {  
  text-shadow: 4px 4px 8px #333;  
}
```

3. Efectos Texto CSS 3

word-wrap:

Si una palabra es muy larga para caber dentro de una caja, se expande fuera de la caja, en CSS3, la propiedad **word-wrap** permite forzar el texto para que se ajuste.

Sintaxis: ***word-wrap: normal/break-word;***

Ejemplo:

```
p {  
    word-wrap: break-word;  
}
```

1. Web Fonts

@font-face:

Web Fonts permite a los diseñadores web utilizar fuentes que no están instaladas en el ordenador (PC) del usuario.

Sintaxis: *@font-face { font-family: nombreFuente; src: url(fuente.ttf); }*

Ejemplo:

```
@font-face {  
    font-family: miFuente;  
    src: url(Roboto.woff);  
}
```

```
h1 {  
    font-family: miFuente;  
}
```

2. Web Fonts

Distintos formatos de fuentes

Fuentes TrueType (TTF)

TrueType es el formato de la fuente más común tanto para el Mac OS y sistemas operativos de Microsoft Windows.

Fuentes OpenType (OTF)

OpenType es un formato para las fuentes informáticas escalables y es una marca comercial registrada de Microsoft.

Fuentes Web Open Font Format (WOFF)

WOFF es un formato de fuente para su uso en páginas web, ahora es una Recomendación del W3C.

Fuentes SVG / formas

SVG permite el uso de glifos (vectoriales) para mostrar mostrar el texto.

Fuentes OpenType incrustadas (EOT)

Fuentes EOT son una forma compacta de fuentes OpenType diseñado por Microsoft para su uso como fuentes incrustadas en las páginas web.

1. Transformaciones 2D

transform:

Con **transform** en CSS3, podemos mover, escalar, girar y rotar.

Una transformación es un efecto que permite a un elemento cambiar la forma, el tamaño y la posición.

- **translate()**
- **rotate()**
- **scale()**
- **skew()**
- **matrix()**

2. Transformaciones 2D

translate:

Con el método **translate()**, el elemento se mueve desde su posición original, en función a los nuevos parámetros de posición en el (eje X) y (eje Y).

Sintaxis: *transform: translate(x,y);*

transform: translateX(medida);

transform: translateY(medida);

Ejemplo:

```
section {  
    transform: translate(80px,160px);  
}
```

3. Transformaciones 2D

rotate:

Con el método **rotate()**, el elemento gira en sentido horario en función a un valor (Ángulo) en grados. Los valores negativos son permitidos, el elemento gira al contrario del sentido horario.

Sintaxis: *transform: rotate(ángulo);*

Ejemplo:

```
section {  
    transform: rotate(45deg);  
}
```

4. Transformaciones 2D

scale:

Con el método de **scale()**, el elemento aumenta o disminuye el tamaño dependiendo de los parámetros dados para el ancho (eje X) y la altura (eje Y).

Sintaxis: *transform: scale(x,y);*

transform: scaleX(n);

transform: scaleY(n);

Ejemplo:

```
section {  
    transform: scale(2,2);  
}
```

5. Transformaciones 2D

skew:

Con el método de **skew()**, el elemento gira en un ángulo dado, en función de los parámetros dados para el (eje X) horizontal y el (eje Y) vertical.

Sintaxis: *transform: skew(ánguloX,ánguloY);*

transform: skewX(ángulo);

transform: skewY(ángulo);

Ejemplo:

```
section {  
    transform: skew(30deg,60deg);  
}
```

6. Transformaciones 2D

matrix:

El método **matrix()** combina la totalidad de métodos de transformación 2D.

El método toma seis parámetros, que contiene funciones matemáticas, que le permiten: rotar, escalar, mover (trasladar) y sesgar (inclinarse) elementos.

Sintaxis: *transform: matrix(n,n,n,n,n,n);*

Ejemplo:

```
section {  
    transform: rotate(15deg) translateX(230px) scale(1.5, 2.6) skew(220deg, -150deg) translateX(230px)  
    /* scale(x,y), skew(x,y), translate(x,y) */  
    transform: matrix(1.06, 1.84, 0.54, 2.8, 466px, 482px);  
}
```

Transformaciones 3D

transform:

CSS3 permite cambiar los elementos usando transformaciones 3D.

- **rotateX()**

El elemento gira alrededor de su eje X en un valor dado en grados.

Sintaxis: ***transform: rotateX(20deg);***

- **rotateY()**

El elemento gira alrededor de su eje Y en un valor dado en grados.

Sintaxis: ***transform: rotateY(60deg);***

1. Transiciones

transition:

Las transiciones CSS3 son efectos que permiten un elemento cambiar gradualmente de un estilo a otro.

Requisitos mínimos para una transición:

Sintaxis: *transition: propiedad tiempo;*

```
section {  
    /*  
    transition-property: width;  
    transition-duration: 2s;  
    transition-timing-function: linear;  
    transition-delay: 2s;  
    */  
    transition: width 2s;  
}  
section:hover {  
    width: 600px;  
}
```

2. Transiciones

Multiples transiciones:

Para agregar efectos de transición para más de una propiedad CSS, separe las propiedades con una coma:

Sintaxis: *transition: propiedad1 tiempo1, propiedad2 tiempo2 ;*

```
section {  
    transition: width 2s, height 2s, transform 2s;  
}  
section:hover {  
    height: 400px;  
    transform: rotate(120deg);  
    width: 600px;  
}
```

1. Animaciones

@keyframes

La regla **@keyframes** es donde se crea la animación.

Se debe especificar un estilo CSS dentro de la regla **@keyframes** y la animación cambiará gradualmente del estilo actual al nuevo estilo.

Sintaxis: **@keyframes nombreAnimacion {**
 0% { propiedad: valor; }
 100% { propiedad: nuevovalue; }
}

```
@keyframes miAnimacion {  
  from { background-color: #ff0; }  
  to   { background-color: #f00; }  
}
```

1. Animaciones

animation

Una animación permite a un elemento cambiar gradualmente de un estilo a otro.

Sintaxis: *animation: nombreAnimacion duración ;*

```
section {  
    /*  
        animation-name: miAnimacion;  
        animation-duration: 2s;  
        animation-timing-function: linear;  
        animation-delay: 1s;  
        animation-iteration-count: infinite;  
        animation-direction: alternate;  
        animation-play-state: running;  
    */  
    /* animation: miAnimacion 2s linear 1s infinite alternate; */  
    animation: miAnimacion 2s;  
}
```

Multiples Columnas

- **column-count**
- **column-gap**
- **column-rule**

Con CSS3, puede crear varias columnas para el texto, como en los periódicos.

```
section {  
    column-count: 3;  
    column-gap: 20px;  
    column-rule: 4px outset #f00;  
}
```

Media Queries

@media screen and () { ... }

La regla **@media** nos permite especificar a ciertas reglas CSS que solo se deben aplicar a cierto tipo de dispositivo.

Así las definiciones dentro del bloque de la regla **@media screen { ... }** solo serían interpretadas por dispositivos conectados a monitores de ordenador.

Sintaxis: *@media screen and (condición) {*

selector {

propiedad: valor;

}

}

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

```
@media (min-width: 800px) {
```

```
  #menu {
```

```
    display: none;
```

```
  }
```

```
}
```