

# **CSS3: Nuevas Reglas**

Antonio Espín Herranz

# Contenidos

- Introducción.
- Border-radius.
- Box-shadow.
- Text-shadow.
- @font-face.
- Gradiente lineal.
- Gradiente radial.
- RGBA.
- HSLA.
- Outline.
- Border-image.
- Transform / Transition.

# Introducción

- ~~Algunas propiedades puede que no estén disponibles dependiendo el navegador o pueden estar en fase de experimentación.~~ Ya deberían estar en todos los navegadores, pero seguían esta normal de prefijos:
- Se pueden indicar con los siguientes prefijos:
  - Prefijos:
    - moz**: Firefox.
    - webkit**: Safari y Chrome.
    - o**: Opera.
    - khtml**: Konqueror.
    - ms**: IE.
    - chrome**: Específico de Chrome.

# Ejemplo

- Partimos de:

```
<header id="principal">  
  <span id="titulo">  
    Estilos CSS Web 2.0  
  </span>  
</header>
```

- CSS inicial:

```
body {  
  text-align: center;  
}  
  
#principal {  
  display: block;  
  width: 500px;  
  margin: 50px auto;  
  padding: 15px;  
  text-align: center;  
  border: 1px solid #999999;  
  background: #DDDDDD;  
}  
  
#titulo {  
  font: bold 36px verdana, san-serif;  
}
```

# Esquinas redondeadas: border-radius

- CSS3 agregar esquinas redondeadas a partir de la propiedad: **border-radius**.
- Indicamos el radio el pixel y se puede asignar un radio distinto para cada esquina de la misma forma que lo haríamos con margin.

```
#titulo {  
    border-radius: 20px;  
}
```

- También es válido: border-radius: 2px 5px 2px 3px.
  - **Orden de aplicación: esq. Sup iz y sentido horario.**
  - También se puede aplicar solo dos valores.
- Si las propiedades están experimentales se pueden indicar con los prefijos: moz, webkit
  - moz-border-radius: 20px;**
  - webkit-border-radius: 20px;**
  - border-radius: 20px;

# Esquinas redondeadas: border-radius

- A las esquinas también se les puede redondear en forma de eclipse.
- Para ello indicamos un radio horizontal y otro vertical:
- border-radius: 20px / 10px
  - 20px: radio horizontal.
  - 10px: radio vertical.

# Sombras: box-shadow

- Podemos aplicar sombras a cajas.
- 6 propiedades:
  - { offset-x offset-y blur-radius spread-radius color } → sin inset
  - { **inset** offset-x offset-y blur-radius spread-radius color }
- Ejemplo:
  - box-shadow: rgb(150, 150, 150) 5px 5px;
    - Color.
    - Desplazamiento vertical de la sombra.
    - Desplazamiento horizontal de la sombra.
    - Los desplazamientos pueden ser positivos o negativos.
      - Positivo: la sombra se desplaza por la derecha y la parte inferior.
      - Negativo: Por la izq y la parte superior.
    - Para propiedades experimentales:
      - Prefijos: -moz y -webkit.

# Sombras: box-shadow

- Se puede añadir un **tercer parámetro**: número en pixels.
  - **Difumina** la sombra → **blur-radius** (desenfoque)
  - **Extensión** → **spread-radius** (extensión)
- **inset** → **inserción**
  - Convierte la sombra en interna.
  - Efecto de profundidad.



# Ejemplos

. box-shadow { **offset-x** **offset-y** **blur-radius** **spread-radius** **color** }



0px 0px darkred;



5px 10px darkred;



0px 0px 20px darkred;



5px 10px 20px darkred;



5px 10px 40px darkred;



0px 0px 0px 20px darkred;



5px 10px 0px 20px darkred;



5px 10px 0px 40px darkred;

*Si aumenta el valor de desenfoque, la sombra se ve más borrosa. Por otro lado, si aumenta el valor de dispersión, el tamaño de la sombra es mayor que el valor.*

# Sombras en Texto: Text-shadow

- `text-shadow: rgb(0, 0, 150) 3px 3px 5px;`
  - Tiene los mismos parámetros que `box-shadow`:
    - Distancia horizontal.
    - Vertical.
    - Y radio de difuminación.

# @font-face

- CSS3 permite agregar nuestros **propios estilos de fuente**.
- Podemos agregar nuestros propios ficheros de fuente de texto.
- Tenemos que declarar el estilo **@font-face** donde indicamos el **nombre de la fuente** y el **fichero asociado**, después la podemos utilizar dentro de nuestro documento.

# @font-face

- **Declaración:**

```
@font-face {  
    font-family: 'MiFuente';  
    src: url('font.ttf');  
}
```

- **Uso de la fuente:**

```
#titulo {  
    font: bold 36px MiFuente, verdana, sans-serif;  
}
```

# Ejemplo

- Se pueden declarar múltiples variaciones de una familia de fuentes cambiando las propiedades `font-weight` y `src` de la regla `@font-face`.

```
/* Regular Weight */
@font-face {
  font-family: Montserrat;
  src: url("../fonts/Montserrat-Regular.ttf");
}

/* SemiBold (600) Weight */
@font-face {
  font-family: Montserrat;
  src: url("../fonts/Montserrat-SemiBold.ttf");
  font-weight: 600;
}

/* Bold Weight */
@font-face {
  font-family: Montserrat;
  src: url("../fonts/Montserrat-Bold.ttf");
  font-weight: bold;
}
```

```
/* Regular */
font-family: Montserrat;

/* Semi Bold */
font-family: Montserrat;
font-weight: 600;

/* Bold */
font-family: Montserrat;
font-weight: bold;
```

# Gradiente lineal

- CSS3 permite generar degradados como estilos de fondo.
- Se configuran con las propiedades background o background-image.
- Sintaxis:
  - linear-gradient(inicio, color inicial, color final)
  - Utilizar prefijos:  
background: -moz-linear-gradient(top, #FFFFFF, #006699);  
background: -webkit-linear-gradient(top, #FFFFFF, #006699);

# Gradiente lineal

- Los posibles valores del inicio, pueden ser:
  - top, left, right, bottom.
  - Pixels o porcentaje.
  - O se puede indicar un ángulo:
    - `linear-gradient(30deg, #FFFFFF, #006699);`
  - Se puede indicar los puntos de terminación de cada color:
  - *`linear-gradient(top, #FFFFFF 50%, #006699 90%);`*

**Estilos CSS Web 2.0**

# Gradiente radial

- Comienzo se indica en %, px o con las palabras claves: center, top, bottom, left, right.
- La forma puede ser ellipse / circle.
- Color de inicio y terminación.
  - Al igual que el linear se pueden indicar los puntos de inicio y terminación del color.

background: *-moz-radial-gradient(center, circle, #FFFFFF , #006699);*

background: *-webkit-radial-gradient(center, circle, #FFFFFF, #006699);*



# RGBA

- La función `rgba()` es una ampliación de la función `rgb()`.
- Añade un parámetro que es la opacidad.
- Los valores `r`, `g` y `b` se indican en el rango de 0 a 255.
- La opacidad en el rango de 0..1.
  - 0 → Transparente.
  - 1 → Totalmente opaco.
- Se aplica a propiedades que tengan que ver con el color.

# HSLA

- Al igual que ocurre con `rgb()` y `rgba()`, la función `hsl()` se complementa con `hsla()`.
- Ejemplo:
  - `color: hsla(120, 100%, 50%, 0.5);`
- **`hsla(tono, saturación, luminosidad, opacidad)`**
  - Tono: valores de 0 a 360°.
    - Rojos: cerca de 0 y 360.
    - Verdes: 120
    - Azules: 240.
  - Saturación: porcentaje.
    - 0% (escala de grises) a 100% (completamente saturado).
  - Luminosidad: porcentaje.
    - 0% (completamente oscuro) a 100% (completamente iluminado).
  - Opacidad:
    - De 0 a 1.

# Outline

- CSS3 añade **un valor de desplazamiento al borde** que muestra.

outline: mismos parámetros que border.

Ejemplo:

outline: 2px dashed red;

outline-offset: 15px;

(indica el desplazamiento del borde que añade)

# Border-image

- Nos ofrece la posibilidad de crear un borde con una imagen propia.
- Toma la imagen indicada mediante url y la toma como patrón para rellenar un borde alrededor.
- Tenemos que indicar 3 atributos:
  - La imagen de muestra.
  - El tamaño.
  - Y una palabra clave para indica como tenemos que rellenar el borde.

# Border-image

- `border-image: url("mi_imagen.png") 25 stretch`
  - 25 indica el tamaño en pixels.
  - Palabras clave para el 3er. Parámetro:
    - **repeat**: repite la imagen las veces que sea necesario. Puede cortar una pieza.
    - **round**: Ajusta el tamaño de la pieza para no cortarla.
    - **stretch**: Estira sólo una pieza para cubrir el lado completo.
- Utilizar la propiedad con los prefijos.

# Transform / Transition

- CSS3 se completa con la generación de transformaciones y transiciones que antes sólo se podían realizar con javascript o librerías como jquery.
- **Transform:** proporciona 4 tipos de transformaciones:
  - **scale** (escalar), **rotate** (rotar) , **skew** (inclinarse) y **translate** (mover).

# Transform: scale

- Indicamos los valores en x e y, si especificamos un solo valor hacemos referencia a los dos ejes:
  - moz-transform: scale(2)
  - webkit-transform: scale(2)
- Valores  $> 1$  amplían,  $< 1$  disminuyen y 1 se quedan igual.
- Indicando el valor 1,-1:

Estilos CSS MEP 5.0

# Transform: rotate

- Podemos rotar indicando los grados: “deg”.
  - moz-transform: rotate(30deg);
  - webkit-transform: rotate(30deg);
- Si indicamos un número negativo cambia la dirección en que será rotado.



# Transform: skew

- **Cambia la simetría del elemento en grados y en ambas direcciones.**
  - moz-transform: skew(20deg);
  - webkit-transform: skew(20deg);
  - En este ejemplo sólo cambia la horizontal.
- Si sólo indicamos un parámetro sólo afecta a una de las dimensiones.
- Podemos utilizar las funciones: **skewX** y **skewY**.

# Transform: translate

- Mueve o desplaza el objeto a una nueva posición.
  - moz-transform: translate(100px);
  - ms-transform: translate(100px);
- Cuando se indica un único parámetro hace referencia a la posición horizontal.
- ***El origen se centra en el 0,0 esquina sup. Izq. Valores negativos desplazan a la izq y hacia arriba. Valores positivos a la derecha y hacia abajo.***
- Podemos indicar los dos, con: **translateX,**  
**translateY**

# Transformando todo / dinamismo

- Las transformaciones se pueden aplicar a la vez:
  - moz-transform:translateY(100px) rotate(45deg) scaleX(0.3);
  - webkit-transform:translateY(100px) rotate(45deg) scaleX(0.3);
- Transformaciones dinámicas:
  - Se pueden combinar las transformaciones con pseudoclases para aplicar dinamismo.

```
#principal:hover {  
    -moz-transform: rotate(5deg);  
    -webkit-transform: rotate(5deg);  
}
```

# Transiciones

- Crea una transición de un estado inicial a uno final.
- Suaviza los cambios crea los pasos intermedios que se encuentran implícitos en el movimiento.
- transition: recibe 4 parámetros.
  - 1) Indica la transformación.
  - 2) Indica el tiempo que dura la transformación.
  - 3) Alguna palabra clave: ease, linear, ease-in, ease-out, ease-in-out: hacen referencia a transiciones basadas en curvas de Bézier.
  - 4) Indica el retardo, cuanto tiempo tarda en empezar.

# Transiciones: ejemplo

```
#principal {  
  display: block;  
  width: 500px;  
  margin: 50px auto;  
  padding: 15px;  
  text-align: center;  
  border: 1px solid #999999;  
  background: #DDDDDD;  
  
  -moz-transition: -moz-transform 1s ease-in-out 0.5s;  
  -webkit-transition: -webkit-transform 1s ease-in-out 0.5s;
```

```
#principal:hover {  
  -moz-transform: rotate(5deg);  
  -webkit-transform: rotate(5deg);  
}
```

# Animaciones

- Ejemplo: **Fade In** de una capa.

```
@keyframes fadein {  
  from {  
    opacity:0;  
  }  
  to {  
    opacity:1;  
  }  
}
```

```
.diapositiva {  
  border: 2px solid #AAAAAA;  
  margin: 30px;  
  border-radius: 10px;  
  background-color: #EEEEEE;  
  animation: fadein 4s;  
}
```

# Enlaces

- **Generador CSS3:**

- <http://css3generator.com/>
- <http://www.colorzilla.com/gradient-editor/>

- **Recursos y Herramientas:**

- <http://www.cosassencillas.com/articulos/herramientas-recursos-guias-html5-css3>

- **CSS3 en IE9:**

- <http://msdn.microsoft.com/en-us/ie/ff468705.aspx>

# Enlaces

- Ejemplos:
- <http://www.freshdesignweb.com/html5-css3-3d-examples-demo.html>