

CSS3

- **box-sizing**: permite definir certos elementos (como un div) para que se axusten a un determinado espazo. Os valores que pode tomar son os seguintes:
 - **content-box**, valor por defecto, comportamento como CSS 2.1.
 - **border-box**, axusta o elemento e os seus bordes ó tamaño definido. Por exemplo, un div cun width de 400px e con bordes definidos de 10px ocupará 400px, fronte ós 420px que ocuparía co comportamento content-box.
- **border-radius**: permite definir bordes redondeados para elementos coma div, img,... Os valores que pode ter son os seguintes:
 - **10px**, por exemplo, aplicará a cada esquina do elemento un borde redondeado de radio 10 px.
 - **10px 20px**, aplicará un borde redondeado á esquina superior esquerda e a esquina inferior dereita de 10 px. Ás outras dúas esquinas aplicaralle un radio de 20px.
 - **10px 20px 30px 40px**, aplica a cada esquina (comezando pola superior esquerda e seguindo o sentido da agullas do reloxo) o radio indicado.
 - **10px / 20px**, aplicará un radio horizontal de 10 px e vertical de 20px, buscando un efecto de elipse
- **box-shadow**: aplica unha sombra ó elemento que lle indiquemos. Necesita, alomenos, 3 valores:
 - **cor da sombra**, por exemplo rgb(150,150,150).
 - **desprazamento no eixe X**. Valores positivos colocan a sombra á dereita do elemento. Negativos á esquerda.
 - **desprazamento no eixe Y**. Valores positivos colocan a sombra debaixo do elemento. Negativos enriba.
 - **degradado (opcional)**, canto maior sexa o valor maior é o degradado/difuminado que lle aplica.
 - **inset**, crea a sombra interna ó obxecto, en vez de externa. Neste caso os valores se invirten, é dicir, valores positivos colocaran a sombra arriba e á esquerda e negativos ó contrario.
- **text-shadow**: aplica unha sombra a un texto. Funciona exactamente igual que box-shadow salvo polo inset, que obviamente non o implementa.
- **@font-face**: permite importar un arquivo cunha fonte para ser aplicada no noso site. Os pasos a seguir son os seguintes:
 - **Declarar a fonte**: do seguinte xeito:

```
@font-face {  
    font-family: 'NovaFonte';  
    src: url('arquivo.ttf') format ('truetype'), url ('arquivo.woff') format ('woff'),...;  
}
```
 - **Utilizala**: do seguinte xeito:

```
#elemento {  
    font: bold 36px NovaFonte, verdana,...  
}
```

- **linear-gradient**: gradiente lineal que se pode aplicar a un fondo ou a unha imaxe de fondo. Os valores que hai que introducir son:
 - **posición de inicio**, onde se comezará a aplicar o gradiente (top, bottom, left, right). Pódense combinar (top left, bottom right) e, tamén, dita posición pode ser remplazada por un ángulo, por exemplo 30deg.
 - **cor inicial**. Pódense combinar con transparencias usando rgba (255,0,0,1) inicialmente e (255,0,0,0) na cor final. Verémolo máis adiante.
 - **cor final**.
Opcionalmente, pódense engadir máis cores polo medio, por exemplo, como se fose un arco iris.
- **radial-gradient**: gradiente radial cun funcionamento casi idéntico ó anterior. Xeneralmente utilízase a posición inicial “center”. Ademáis, despois da posición, pódese indicar o seguinte parámetro:
 - **circle/ellipse**,ponse despois da posición inicial.
- **rgba**: transparencias. Ten catro atributos:
 - **Os tres primeiros** son a combinación rgb do cor.
 - **O último** indicará o grao de transparencia que queremos aplicar. Irá do 0 (totalmente transparente) ó 1 (totalmente opaco).

Obviamente, pódese combinar co resto de propiedades.

- **hsla**: casi idéntica a rgba pero utiliza outro formato. Consta de catro propiedades:
 - **tono**, utilízase unha roda imaxinaria na que os valores cercanos a 0 e 360 serían vermellos, os cercanos a 120 verdes e os cercanos a 240 azuis.
 - **saturación**, irá de 0% (escala de grises) a 100% (totalmente saturado).
 - **luminosidade**, irá do 0% (totalmente escuro) ó 100% (totalmente iluminado)
 - **opacidade**, ou transparencia, igual que no caso anterior.
- **outline**: propiedade de CSS 2.1 que o que fai é aplicar un borde externo a un elemento de tipo caixa. Se comporta igual que border (tamaño, cor e tipo de borde) pero permite unha salvedade:
 - **outline-offset**, combinada con outline indica o desprazamento hacia fora de dito segundo borde.
- **border-image**: utiliza unha imaxe como patrón para realizar un borde (é necesario indicar o tamaño do border previamente). Necesitamos especificar tres atributos:
 - **url(“arquivo”)**, ubicación da imaxe.
 - **tamaño do patrón**, é dicir, tamaño que queremos que teñan as pezas individualmente.
 - **Distribución**, palabra chave que nos indicará como se distribuirán as pezas ó redor do elemento. Poden ser:
 - **repeat**, a partires do espacio dispoñible (lonxitude do borde) e tendo en conta o tamaño declarado da imaxe, porá tantas imaxes como quepan. No caso de que non quepan xustas porá anacos das mesmas.
 - **round**, similar ó anterior salvo que “redondea” ó espazo dispoñible ás imaxes facéndoo algo máis grandes para que non aparezan trozos soltos, é como se fose un párrafo xustifico.

- stretch, utiliza unha soa imaxe e a axusta/estira/encolle ó tamaño de que dispón.

Esta propiedade ten varias consideracións:

- Pódese combinar as distribucións, é dicir, se facemos round stretch conseguiremos un efecto round nas liñas horizontais e un efecto stretch nas liñas verticais.
- De momento non funciona con Internet Explorer.
- Se utilizamos Firefox debemos indicar na propiedade border, ademáis do tamaño, o estilo do borde e a cor (transparente neste caso).
- A imaxe deberá ser deste estilo:

Esquina recheo **Esquina**
recheo recheo
Esquina recheo **Esquina**

- text-overflow: especifica qué debe pasar cando o texto sobrepasa o tamaño do contenedor. Posibles valores:
 - **clip**, valor por defecto, simplemente oculta o que non cabe no texto.
 - **ellipsis**, combinado con white-space:nowrap e con overflow:hidden (para que non se force un crecemento do contenedor) substitúe o texto que non cabe por puntos suspensivos.
- opacity: permite realizar un degradado de, por exemplo, unha imaxe, podendo deixala case que transparente. Os posibles valores van de 1 (totalmente opaca) a 0 (totalmente transparente).

TRANSFORM

- **scale**: permite escalar un elemento. Posibles combinacións:
 - **scale(2)** - escala o elemento utilizando un multiplicador, neste caso 2.
 - **scaleX(2)** - escala o elemento lonxitudinalmente utilizando un multiplicador, neste caso 2.
 - **scaleY(2)** - escala o elemento verticalmente utilizando un multiplicador, neste caso 2.
 - **scale(2,2)** - escala o elemento lonxitudinal e verticalmente utilizando un multiplicador, neste caso 2.

Pódense utilizar negativos.

- **rotate**: rota o elemento en sentido das agullas do reloxo. O valor debe de ser indicado en grados, por exemplo, 30deg. Tamén se pode rotar tendo en conta cada un dos eixos de coordenadas. Por exemplo:
 - **rotateX(ángulo)**, rota en función do eixo X.
 - **rotateY(ángulo)**, rota en función do eixo Y.
 - **rotateZ(ángulo)**, funciona igual que rotate a secas.

Tamén se poden utilizar valores negativos.

- **skew**: inclina/distorsiona/estira o elemento tendo en conta o ángulo escollido.
 - **skewX(ángulo)**, despraza a arista superior do elemento á esquerda e a inferior do elemento á dereita según o ángulo escollido ata 90°, onde se solapan.
 - **skewY(ángulo)**, o mesmo pero no eixo Y.
 - **skew(ánguloX,ánguloY)**, combinación dos dous.Tamén se poden utilizar valores negativos.
- **translate**: despraza o elemento.
 - **translate(x,y)**, despraza x unidades á dereita e y unidades hacia abaixo. Se usamos numeración negativa será ó revés.
 - **translateX(x)**.
 - **translateY(y)**.

TRANSITION

- Permite realizar efectos como transicións dun estado a outro. Admite ata catro valores:
 - **propiedade (transition-property)**, por exemplo transform, width,...
 - **duración (transition-duration)**, tempo que lle leva ó proceso.
 - **velocidade da curva de transición (transition-timing-function)**, por exemplo linear (velocidade de execución lineal), ease-in (lento ó principio e despois se acelera), ease-out (lento só ó final), ease-in-out (lento ó principio, se acelera e se frena ó final tamén) e ease (como ease-in-out pero máis brusco).
 - **demora (transition-delay)**, pódese indicar que o efecto non comece xusto no segundo 0.

Pódese executar en short-hand ou propiedade a propiedade. Unhas páxinas de exemplo de como e para qué usar transicións poden ser:

<http://tympanus.net/codrops/2012/01/09/filter-functionality-with-css3/>

<http://tympanus.net/codrops/2011/12/26/css3-lightbox/>

ANIMATION

- Permite crear animacións en CSS3. Baséase no uso da regra `@keyframes`. Podemos escoller entre as seguintes propiedades:
 - `animation-name`, o nome que terá a regra `@keyframes` que creemos.
 - `animation-duration`, cantos segundos (s) ou milisegundos (ms) durará a animación.
 - `animation-timing-function`, será a curva de velocidade da animación. Pode ter valores coma `linear` (misma velocidade de entrada e saída), `ease` (comeza lento, se acelera e termina lento), `ease-in` (comeza lento), `ease-out` (acaba lento), `ease-in-out` (moi parecido a `ease`) e `cubic-bezier` (defines ti mesmo os valores da animación).
 - `animation-delay`, retardo no inicio da animación en segundos (s) ou milisegundos (ms).
 - `animation-iteration-count`, especifica o número de veces que se realiza a animación. Por defecto só se fai 1 vez. Pódese por `infinite` para indicar que se repite indefinidamente.
 - `animation-direction`, podemos indicar que a dirección de animación pode ser normal (por defecto), ó revés (`reverse`), que alterne un sentido co inverso (`alternate`), o mesmo pero comezando ó revés (`alternate-reverse`)...
 - `animation-play-state`, podemos pausar a animación (`paused`).
- Coa regra `@keyframes` é co que realmente creamos a animación. A sintaxe difire un pouco do que estamos acostumbrados, xa que engloba diferentes regras ou comportamentos dentro dela. Vexamos un exemplo:

```
@keyframes nosa_regra {
  0% {
    top: 0px;
  }
  100% {
    top: 200px;
  }
}
```

0% é equivalente á palabra reservada `from` e 100% equivale á palabra reservada `to`. Neste caso quere dicir que ó inicio da animación (0% - `from`) terase un comportamento e ó final (100% - `to`) terase outro. Cos porcentaxes podemos xogar con estados intermedios pero coas palabras reservadas non.

```
@keyframes nosa_regra {
  0% {
    top: 0px;
  }
  25% {
    top: 200px;
  }
  75% {
    top: 50px;
  }
  100% {
```

```
        top: 100px;
    }
}
```

Esta regra faría un efecto vaivén. Ó inicio da transición atoparase na posición 0, baixará á posición 200 no primeiro cuarto da animación, subirá á posición 50 no segundo cuarto da animación para baixar á posición 100 cara o final.

Podemos aplicarlle estilos relacionados coa posición (top, left,...), coa cor (background-color,...), etc.