

# CSS – EXPLICACIÓN Y PRÁCTICA

## 1) Transparencias → Se realiza mediante la propiedad opacity.

Crea tres div en html, cada uno con id “d1”, “d2” y “d3”. Mediante la propiedad background-color les das colores blue, red y green, y aplica la propiedad opacity con valor 0.5 a los tres div. Los tres tendrán un ancho y alto de 100px. Para posicionarlos utiliza las siguientes propiedades:

Primer div

position → absolute, top→ 100px, left→ 100px

Segundo div

position → absolute, top→ 115px, left→ 115px

Tercer div

position → absolute, top→ 130px, left→ 130px

El resultado debe ser el siguiente:



Cambia el valor de la propiedad opacity. ¿Qué pasa si le pones valor 1? ¿Y con valor 0.2?

## 2) Múltiples imágenes de background.

Se puede hacer de dos formas, utilizas la propiedad **background** y en cada fila, separada por comas defines las propiedades para cada imagen

```
background:
<image> <position> <size> <repeat> <attachment> <box>,
<image> <position> <size> <repeat> <attachment> <box>,
<image> <position> <size> <repeat> <attachment> <box>,
<image> <position> <size> <repeat> <attachment> <box> <color>;
```

o bien defines con la propiedad **background-image** todas las imágenes que quieres utilizar, **background-repeat** que especifica si la imagen se muestra sólo una vez, **background-position** para conocer la posición de cada imagen y los valores pueden ser los siguientes

Value	Description	Play it
left top left center left bottom right top right center right bottom center top center center center bottom	If you only specify one keyword, the other value will be "center"	
x% y%	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. . Default value is: 0% 0%	
xpos ypos	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other <a href="#">CSS units</a> . If you only specify one value, the other value will be 50%. You can mix % and positions	
initial	Sets this property to its default value. <a href="#">Read about initial</a>	
inherit	Inherits this property from its parent element. <a href="#">Read about inherit</a>	

La propiedad **background-attachment** indica si la imagen debiera estar fija o se permita hacer scroll y para definir distintos tamaños de las imágenes, se puede utilizar la propiedad **background-size** – **background-size: 25px, 30px, 125px; /\*Fíjate que el tamaño de cada imagen está separado por comas\*/**

```
background-image: <image>, <image>, <image>, <image>;
background-repeat: <repeat>, <repeat>, <repeat>, <repeat>;
background-position: <position>, <position>, <position>, <position>;
```

**chuletila para todos los valores:**

**background-color:** /\* transparent | color en formato válido \*/

**background-image:** url(''); /\*(imágenes separadas por coma)\*/

**background-origin:** ; /\* padding-box|border-box|content-box \*/

**background-position:** /\* left top | left bottom | left center | center | x% y% | ... \*/

**background-size:** ; /\* cover | contain \*/

**background-repeat:** /\* repeat|repeat-x|repeat-y|no-repeat \*/

`background-attachment: ; /* scroll | fixed */`

`background: ; /* podemos escribirlas todas en una declaración */`

Por ejemplo si queremos utilizar el siguiente fondo:



### Código en el HTML →

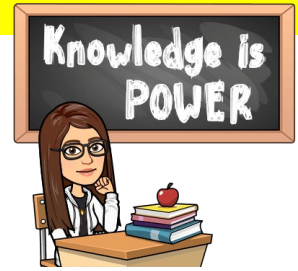
```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <link rel="stylesheet" type="text/css" href="../css/ejercicio02.css" />
6 <title>Practicar múltiples imágenes</title>
7 </head>
8 <body>
9   <div id="contenedor"></div>
10 </body>
11 </html>
```

### Código en el CSS →

```
1 #contenedor{
2   width: 1024px;
3   height: 200px;
4   background: url(..img/sheep.png) 70% 80% no-repeat,
5               url(..img/sheep.png) 60% 78% no-repeat,
6               url(..img/sheep.png) 10% 78% no-repeat,
7               url(..img/sheep.png) 20% 90% no-repeat,
8               url(..img/sheep.png) 40% 78% no-repeat,
9               url(..img/nubes.png) 10% 30% no-repeat,
10              url(..img/nubes.png) 40% 60% no-repeat,
11              url(..img/nubes.png) 60% 10% no-repeat,
12              url(..img/nubes.png) 88% 28% no-repeat,
13              url(..img/sol.png) 80% 20% no-repeat,
14              url(..img/grass.png) left top repeat-x;
15   background-size: 6%, 4%, 4%, 7%, 3%, 12%, 9%, 8%, 15%, 10%, contain;
16 }
```

Ahora crea tu un escenario de background con varias imágenes (intenta ser creativo, por ejemplo un anime que te guste, o algún video juego).

Recuerda que la superposición va de la imagen que quieras que esté primero hacia abajo, es decir, si quiero que lo que se vea es la nube y que detrás vaya el sol, primero tendré que poner la nube y luego el sol.



### 3) Efectos en textos → text-shadow

Como puedes ver la propiedad text-shadow agrega sombra al texto.

text-shadow: (x-offset) (y-offset) (blur-radius) (color)

X-Offset: To position the shadow along the x-axis.

Y-Offset: To position the shadow along the y-axis.

Blur-radius: To set the amount of blur.

Color: To set the color of the shadow.

**Ejemplo:** ANIMAL CROSSING

#### Código en el HTML →

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <link rel="stylesheet" type="text/css" href="../../css/ejercicio03.css" />
6 <title>Practicar sombras</title>
7 </head>
8 <body>
9 <div><span id="idprueba">ANIMAL CROSSING</span></div>
10 </body>
11 </html>
```

#### Código en el CSS →

```
#idprueba{
    text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;
}
```

Como puedes ver tienes dos sombras la primera con X-Offset = 0, Y-Offset=0, Blur-radius=3px y el color de la sombra el #FF0000 y la segunda sombra es X-Offset = 0, Y-Offset=0, Blur-radius=5px y el color de la sombra el #0000FF

**Ahora te toca a tí.**

**Realiza las siguientes transformaciones en el texto para conseguir el resultado mostrado.**

- 1.- Crea una nueva página web que tenga como título "Practicar sombras".
- 2.- Ahora vas a crear 5 **div** y dentro de cada uno un **span** con el siguiente texto:
  - div01 | span id = sp1  
contenido del span "Colorful Style"
  - div02 | span id = sp2

contenido del span “Double Font Style”

- div03 | span id = sp3

contenido del span “Neon Glow Style”

- div04 | span id = sp4

contenido del span “The Inset Effect”

- div05 | span id = sp5

contenido del span “The Outset Style”

**Ahora crea un CSS en el que vamos a ir recogiendo todas estas propiedades:**

- ✓ Aplicar la propiedad font-size = 3.5 em a todos los div.
- ✓ **Colorful Style** → En este caso creamos tres sombras separadas por coma con los siguientes valores
  - primera sombra: 0 6px 4px #85c226
  - segunda sombra: -3px -5px 4px #fe2192
  - tercera sombra: 3px -5px 4px #f7c200
- ✓ **Double Font Style** → creamos una sola sombra con valores
  - sombra: 0 -15px 0 #fe2192
- ✓ **Neon Glow Style** → creamos una sola sombra con valores
  - sombra: 0 0 20px #1e90ff
  - color del texto #1e90ff
- ✓ **the Inset Effect** → creamos dos sombras con valores
  - primera sombra: -1px -1px 3px #666
  - segunda sombra: 1px 1px 3px #fff
  - propiedad **background** y propiedad **color** a #ccc
- ✓ **The Outset Style** → creamos dos sombras con valores
  - primera sombra: 1px 1px 3px #666
  - segunda sombra: -1px -1px 3px #fff
  - propiedad **background** y propiedad **color** a #ccc

#### 4) Efecto con sombras→ box-shadow



To create Outer Shadow:

`box-shadow:X-Offset Y-Offset Blur Blur-Offset Color;`  
Ex: `box-shadow: 0px 0px 20px 10px #00CED1;`



To create Inner Shadow:

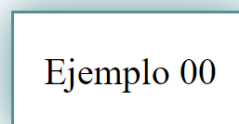
`box-shadow:inset X-Offset Y-Offset Blur Blur-Offset Color;`  
Ex: `box-shadow:inset 0px 0px 20px 10px #00CED1;`

Crear 8 div con su id correspondiente y una clase en común denominada propiedades, y aplicar los siguientes estilos para que el resultado sea como el que se muestra en las imágenes de cada enunciado.

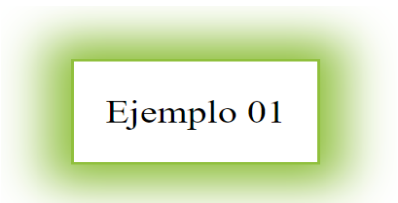
```
.propiedades{  
  font-size: 1em;  
  text-align: center;  
  width: 100px;  
  height: 50px;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

El texto que se encuentra en el div lo hemos justificado utilizando `display: flex` y las propiedades `justify-content` y `align-items`.

- Ejemplo 00 → El div tendrá un borde sólido con color #559098 y para el box shadow tendrá los valores 0px 0px 20px 0px y color #559098.

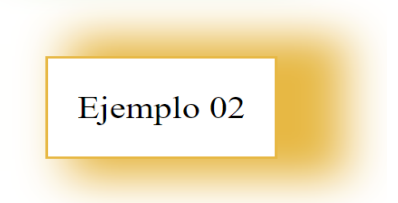


- Ejemplo 01 → El div tendrá un borde sólido con color #8abc37 y para el box shadow tendrá los valores 0px 0px 20px 10px y color #8abc37.



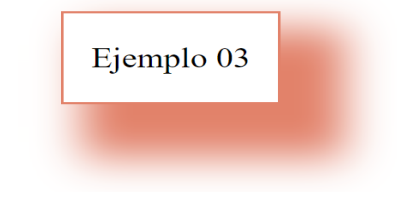
### Ejemplo 01

- Ejemplo 02 → El div tendrá un borde sólido con color #e5b43e y para el box shadow tendrá los valores 20px 0px 20px 10px y color #e5b43e.



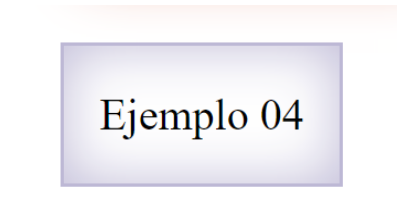
### Ejemplo 02

- Ejemplo 03 → El div tendrá un borde sólido con color #e07c64 y para el box shadow tendrá los valores 20px 20px 20px 10px y color #e07c64.



### Ejemplo 03

- Ejemplo 04 → El div tendrá un borde sólido con color #b9b4d2 y para el box shadow tendrá los valores 0px 0px 20px 0px y color #b9b4d2. (Recuerda que es sombra interior)



### Ejemplo 04

- Ejemplo 05 → El div tendrá un borde sólido con color #c9367d y para el box shadow tendrá los valores 0px 0px 10px 5px y color #c9367d. (Recuerda que es sombra interior)



### Ejemplo 05

- Ejemplo 06 → El div tendrá un borde sólido con color #559098 y para el box shadow tendrá los valores 5px 0px 5px 5px y color #559098. (Recuerda que es sombra interior)

### Ejemplo 06

- Ejemplo 07 → El div tendrá un borde sólido con color #8abc37 y para el box shadow tendrá los valores 5px 5px 5px 5px y color #8abc37. (Recuerda que es sombra interior)

### Ejemplo 07

## 5) Múltiples columnas → column-count

- column-count → especifica el número de columnas deseadas
- column-gap → especifica el ancho del espacio entre columnas
- column-rule → Son bordes entre cada columna. Proporciona el color, estilo y ancho de cada columna.

Por ejemplo, el siguiente texto

La serie comienza con la ejecución de Gol D. Roger, un hombre conocido como Rey de los Piratas (海賊王 Kaizoku-Ō?). Justo antes de su muerte, Roger hace mención de su gran tesoro legendario, el One Piece (ワンピース 『ひとつなぎの大秘宝』 Wan Pīsu?) y que puede ser tomado por quien lo encuentre. Esto marca el inicio de una era conocida como la Gran Era Pirata (大海賊時代 Daikaizokujidai?). Como resultado, un sin número de piratas zarparon hacia el Grand Line con el objetivo de encontrarlo.

Más de veinte años después de la muerte de Roger, el One Piece sigue sin ser encontrado. Un joven llamado Monkey D. Luffy, inspirado por la admiración que desde su infancia le tiene al poderoso pirata Shanks el Pelirrojo, comienza su aventura desde su hogar en el mar East Blue para encontrar el One Piece y autoproclamarse a él mismo como el nuevo Rey de los Piratas. Con el fin de crear y convertirse en el capitán de una tripulación propia, el muchacho funda los Piratas de Sombrero de Paja (麦わら海賊団篇 Mugiwara Kaizoku-dan?), siendo el espadachín Roronoa Zoro el primer nakamaNota que Luffy rescata y alista, lo que permite que ambos continúen emprendiendo su viaje hacia el tesoro. Poco después, Nami, una navegante ladrona, Usopp, un francotirador mentiroso, y Sanji, un cocinero mujeriego se les unen en sus travesías, además de conseguir un barco llamado el Going Merry. En sus viajes, la tripulación se enfrenta a varios enemigos poderosos como Buggy el Payaso, el Capitán Kuro y Don Krieg. Después, el protagonista combate contra el gyojin Arlong, antiguo tripulante de los Piratas del Sol, quien piensa que los humanos son inferiores a los individuos de su propia especie. El capitán de los de Sombrero de Paja logra derrotarlo y Nami es aceptada oficialmente como uno de sus nakamas, además de que la Marina pone precio a la cabeza de este. El grupo logra arribar a Loguetown, un pueblo porteño, y Luffy conoce a un oficial de la Marina llamado Smoker, el cual posee la habilidad de transformarse en humo, y logra capturarlo, pero es detenido por Monkey D. Dragon, el padre de Luffy, aunque él no lo reconoce. Después de llegar finalmente a la Grand Line, el grupo conoce a Nefeltari Vivi, una

princesa que desea salvar su país, el Reino de Arabasta, de manos de una peligrosa organización criminal llamada Baroque Works. Durante su trayecto hacia Arabasta, los piratas encabezan una revuelta la isla de Drum, donde los de Sombrero de Paja invitan a un reno antropomórfico médico llamado Tony Tony Chopper a unirseles al grupo, y éste acepta.

El **código HTML** quedaría de la siguiente forma:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <link rel="stylesheet" type="text/css" href="../../css/ejercicio05p.css" />
6 <title>Practicar múltiples columnas</title>
7 </head>
8 <body>
9   <div id="prueba">
10     La serie comienza con la ejecución de Gol D. Roger, un hombre conocido como Rey de los Piratas (海賊王 Kaizoku-Ō?). Justo antes de su
11     Más de veinte años después de la muerte de Roger, el One Piece sigue sin ser encontrado. Un joven llamado Monkey D. Luffy, inspirado por la ad
12   </div>
13 </body>
14 </html>
```

El **código CSS** quedaría

```
#prueba{
  column-count: 3;
  column-gap: 5em;
  column-rule: 1px solid #000;
```

Bien, utilizando el mismo texto que se utilizó en el ejemplo, ahora vas a aplicar las siguientes reglas:

Cuatro columnas, espacio entre columnas 13em y estilo de las líneas de separación ancho 2px solid y color red.

## 6) Pseudo-classes

Se usan para definir un estado especial de un elemento. Por ejemplo:

- Dar estilo a un elemento cuando un usuario mueve el ratón sobre él
- Dar estilo a un link que ha sido o no visitado
- Dar estilo a un elemento cuando coge el foco

The syntax of pseudo-classes:

```
selector:pseudo-class {  
    property:value;  
}
```


























### All CSS Pseudo Classes

Selector	Example	Example description
<b>:active</b>	a:active	Selects the active link
<b>:checked</b>	input:checked	Selects every checked <input> element
<b>:disabled</b>	input:disabled	Selects every disabled <input> element
<b>:empty</b>	p:empty	Selects every <p> element that has no children
<b>:enabled</b>	input:enabled	Selects every enabled <input> element
<b>:first-child</b>	p:first-child	Selects every <p> elements that is the first child of its parent
<b>:first-of-type</b>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<b>:focus</b>	input:focus	Selects the <input> element that has focus
<b>:hover</b>	a:hover	Selects links on mouse over
<b>:in-range</b>	input:in-range	Selects <input> elements with a value within a specified range
<b>:invalid</b>	input:invalid	Selects all <input> elements with an invalid value
<b>:lang(<i>language</i>)</b>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<b>:last-child</b>	p:last-child	Selects every <p> elements that is the last child of its parent
<b>:last-of-type</b>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<b>:link</b>	a:link	Selects all unvisited links
<b>:not(selector)</b>	:not(p)	Selects every element that is not a <p> element
<b>:nth-child(n)</b>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<b>:nth-last-child(n)</b>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<b>:nth-last-of-type(n)</b>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<b>:nth-of-type(n)</b>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<b>:only-of-type</b>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<b>:only-child</b>	p:only-child	Selects every <p> element that is the only child of its parent
<b>:optional</b>	input:optional	Selects <input> elements with no "required" attribute
<b>:out-of-range</b>	input:out-of-range	Selects <input> elements with a value outside a specified range
<b>:read-only</b>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<b>:read-write</b>	input:read-write	Selects <input> elements with no "readonly" attribute
<b>:required</b>	input:required	Selects <input> elements with a "required" attribute specified
<b>:root</b>	root	Selects the document's root element
<b>:target</b>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<b>:valid</b>	input:valid	Selects all <input> elements with a valid value
<b>:visited</b>	a:visited	Selects all visited links

### All CSS Pseudo Elements

Selector	Example	Example description
<b>::after</b>	p::after	Insert content after every <p> element
<b>::before</b>	p::before	Insert content before every <p> element
<b>::first-letter</b>	p::first-letter	Selects the first letter of every <p> element
<b>::first-line</b>	p::first-line	Selects the first line of every <p> element
<b>::selection</b>	p::selection	Selects the portion of an element that is selected by a user

Los valores preestablecidos para el puntero se muestran a continuación:

Puntero	Navegadores que lo soportan
 <code>cursor: default</code>	Todos
 <code>cursor: crosshair</code>	Todos
 <code>cursor: hand</code>	Solo Internet Explorer
 <code>cursor: pointer</code>	Todos salvo Internet Explorer
 <code>cursor: pointer; cursor: hand</code>	Todos
 <code>cursor: move</code>	Todos
 <code>cursor: text</code>	Todos
 <code>cursor: wait</code>	Todos
 <code>cursor: help</code>	Todos
 <code>cursor: n-resize</code>	Todos
 <code>cursor: ne-resize</code>	Todos
 <code>cursor: e-resize</code>	Todos
 <code>cursor: se-resize</code>	Todos
 <code>cursor: s-resize</code>	Todos
 <code>cursor: sw-resize</code>	Todos
 <code>cursor: w-resize</code>	Todos
 <code>cursor: nw-resize</code>	Todos
 <code>cursor: progress</code>	Solo Internet Explorer
 <code>cursor: not-allowed</code>	Solo Internet Explorer
 <code>cursor: no-drop</code>	Solo Internet Explorer
 <code>cursor: vertical-text</code>	Solo Internet Explorer
 <code>cursor: all-scroll</code>	Solo Internet Explorer
 <code>cursor: col-resize</code>	Solo Internet Explorer
 <code>cursor: row-resize</code>	Solo Internet Explorer
 <code>cursor: url(...)</code>	Solo Internet Explorer

- a) Crear un documento html que contenga un div con el texto “Hover over me to change my style.”, y crear una regla en css para que cambie su color de fondo a negro, el color de letra a blanco y el puntero sea de tipo pointer. El resultado debe ser el siguiente:

---

**a) Hover over me to change my style.**

**a) Hover over me to change my style.**

- b) Crear un documento html que contenga un div con el texto “Hover over me to show the element”, y dentro de él un párrafo oculto que contenga el siguiente texto “Here I am.” crear una regla en css para que muestre el párrafo cuando se pase sobre el div. Utilizar el selector de pseudoclase **hover**. El resultado debe ser el siguiente:

**Hover over me to show the element**

**Here I am**

## 7) Transformaciones y animaciones

Con las transformaciones 2D de CSS3 podemos mover ( `translate` ) redimensionar ( `scale` ) girar ( `rotate` ) y sesgar ( `skew` ) los elementos HTML.

### SINTAXIS

```
elemento { transform: none | funciones de transformación | initial | inherit; }
```

Algunas funciones de transformación:

Transformaciones	Descripción
<code>translate(x,y)</code>	Traslada o mueve un elemento a lo largo de los ejes horizontal ( x ) y vertical ( y )
<code>translateX(x)</code>	Traslada o mueve un elemento solo a lo largo del eje horizontal ( x )
<code>translateY(y)</code>	Traslada o mueve un elemento solo a lo largo del eje vertical ( y )
<code>scale(x,y)</code>	Redimensiona un elemento 2D
<code>scaleX(x)</code>	Redimensiona un elemento solo a lo largo de su eje horizontal ( x )
<code>scaleY(y)</code>	Redimensiona un elemento solo a lo largo de su eje vertical ( y )
<code>skew(ax,ay)</code>	Sesga un elemento a lo largo de sus dos ejes
<code>skewX(ax)</code>	Sesga un elemento a lo largo del eje X (horizontal), donde <code>ax</code> es el ángulo de deformación.
<code>skewY(ay)</code>	Sesga un elemento a lo largo del eje Y (vertical), donde <code>ay</code> es el ángulo de deformación.
<code>rotate(a)</code>	Define una rotation 2D, donde <code>a</code> es el ángulo ( en grados hexagesimales ), donde <code>a</code> es el ángulo de rotación.
<code>matrix(n,n,n,n,n,n)</code>	Define una transformación 2D, utilizando una matriz de 6 valores.
<code>matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)</code>	Define una transformación 3D, utilizando una matriz de 4x4 = 16 valores.

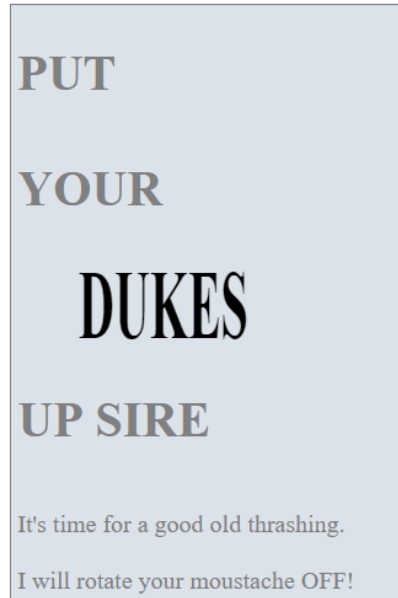
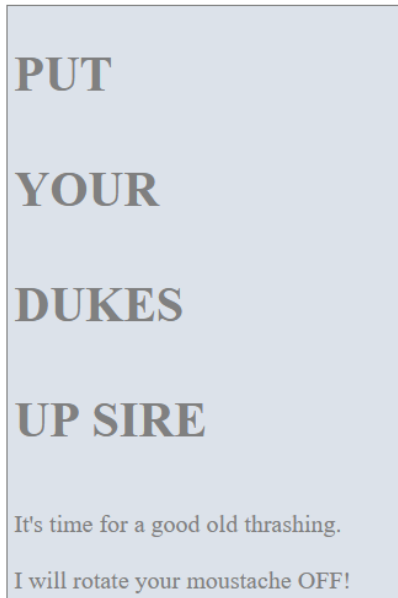
- `Rotate(10deg);`

Ejemplo:

```
div{  
    transform: translateX(20px) scaleY(4px);  
}
```

- Crear un div con id “recuadro” que tenga dentro 6 div con id's div01 hasta div06 cada uno con el siguiente texto **PUT, YOUR, DUKES, UP SIRE, It's time for a good old thrashing., I will rotate your moustache OFF!**.
- El div “recuadro” tendrá de ancho 250px, borde sólido de 1px, color de fondo #dce2ea y un padding de 4px.

- Todos los div tendrán la letra de color gris.
- El div01 tendrá en su propiedad font-size 0.5em y el resto se irá incrementando en 0.2 (es decir, el segundo div tendrá font-size 0.7em, etc).
- Al pasar por encima por el tercer div (propiedad hover) el color a aplicar será #000, se desplazará en el eje X 40px, se realizará un cambio de altura de valor 2. El resultado es el siguiente:





## 8) Animaciones.

**transition-property**, **transition-duration**, **transition-timing-function** y **transition-delay**.

Lista de propiedades sobre las que se pueden hacer transiciones

- background-color y background-position
- border-color, border-spacing y border-width
- bottom, top, left y right
- clip ■ color ■ crop
- font-size y font-weight ■ height and width
- letter-spacing ■ line-height ■ margin ■ max-height, max-width, min-height y min-width
- opacity ■ outline-color, outline-offset y outline-width ■ padding ■ text-indent ■ text-shadow
- vertical-align ■ visibility ■ word-spacing ■ z-index

- Transition-timing-function: Especifica la curva de velocidad de la transición, es decir, si queremos que mantenga una velocidad constante, o un comienzo o final rápido, lento, etc. En concreto podemos utilizar los siguientes:
  - **Linear**: Mantiene la misma velocidad de principio a fin.
  - **Ease**: Comienzo lento, luego rápido y termina lento.
  - **Ease-in**: Comienza lento, y después mantiene velocidad.
  - **Ease-out**: Mantiene velocidad con un final lento.
  - **Ease-in-out**: Comienzo y fin lentos, muy similar a ease sólo que este último empieza más rápido de lo que termina.
  - **Cubic-bezier**: Este es un poco más complejo, no voy a entrar en detalle, pero podemos configurar la curva de velocidad a nuestro gusto utilizando la función cúbica de Bezier. Acepta cuatro valores entre 0 y 1.
  - **Transition-delay**: Especifica el tiempo de retardo en segundos o milisegundos antes del comienzo de la transición.

Si se utiliza hover y se ponen allí las propiedades, entonces en la transición es en la que definimos como se van a comportar esas propiedades.

Por ejemplo defino un div con id "prueba" que cuando se pase el ratón sobre él cambie su tamaño a 200px y cambie el tipo de cursor.

**Código en el HTML →**

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <link rel="stylesheet" type="text/css" href="../css/ejercicio08p.css" />
6 <title>Practicar transiciones</title>
7 </head>
8 <body>
9 | <div id="prueba"></div>
10 </body>
11 </html>
```

### Código en el CSS →

```
#prueba{
  width: 100px;
  height:100px;
  background-color: blue;
}
#prueba:hover{
  width:200px;
  cursor: pointer;
}
```

/\*Visualizalo en el navegador a ver que pasa\*/

Ahora añadimos una transición separada por coma con sus diferentes valores

### Código en el CSS después de añadir la transición →

```
1  #prueba{
2    width: 100px;
3    height:100px;
4    background-color: blue;
5    transition: width 3s linear;
6  }
7  #prueba:hover{
8    width:200px;
9    cursor: pointer;
10 }
```

Vamos a practicar las transiciones. Para ello deberás crear un HTML que contenga tres div. Uno principal y dentro de él otros dos contenedores hijos. Las propiedades de cada uno son:

- Contenedor principal →
  - id = contenedor
  - Propiedades CSS
    - ancho 600px
    - margen 0 auto;
    - alineación del texto centrada
- Contenedor hijo 1 →
  - id = sintransicion
- Contenedor hijo 2 →
  - id = contransicion
- Una clase 'cuadrado' que tendrán los dos contenedores hijos del contenedor principal con las siguientes características CSS
  - ancho 150px
  - alto 150px

- color de fondo #2980b9
- margen 20px
- color #fff
- propiedad line-height 150px
- Cuando el ratón pase por encima de cualquier de esos dos div hijos (hover) las propiedades serán
  - color de fondo #c0392b
  - ancho 200px
  - box-shadow 0px 0px 6px #000
- El efecto de transición que se le aplicará al contenedor hijo 2 'contransicion'
  - para el color de fondo, duración de la transición 2s, manteniendo la velocidad con un final lento.
  - Para el ancho, duración de la transición 1s, comienzo lento, luego rápido y termina lento.
  - Para la sombra de la caja, duración de la transición 1s, mantiene la velocidad de principio a fin.

Uno de los div será sin transición y el otro con transición. Poner una etiqueta a cada uno de ellos para saber cual es de la transición y cuál es el que no tiene transición. Para añadir la etiqueta puedes utilizar una etiqueta span o párrafo. En la carpeta Recursos encontrarás un vídeo de cómo tiene que quedar el ejercicio.