

Agencia de  
Aprendizaje  
a lo largo  
de la vida

# Desarrollo Fullstack



# Les damos la bienvenida

Vamos a comenzar a grabar la clase

## Clase 06

### CSS Inicial

- ▶ Unidades de medida
- ▶ Modelo de Caja
- ▶ Positions

## Clase 07

### CSS Intermedio

- ▶ Pseudoselectores
- ▶ Pseudoclasas
- ▶ Animaciones
- ▶ Transiciones
- ▶ Transformaciones

## Clase 08

### Flexbox

- ▶ Display Flex
- ▶ Características
- ▶ Propiedades

# CSS

Hasta el infinito y más allá...

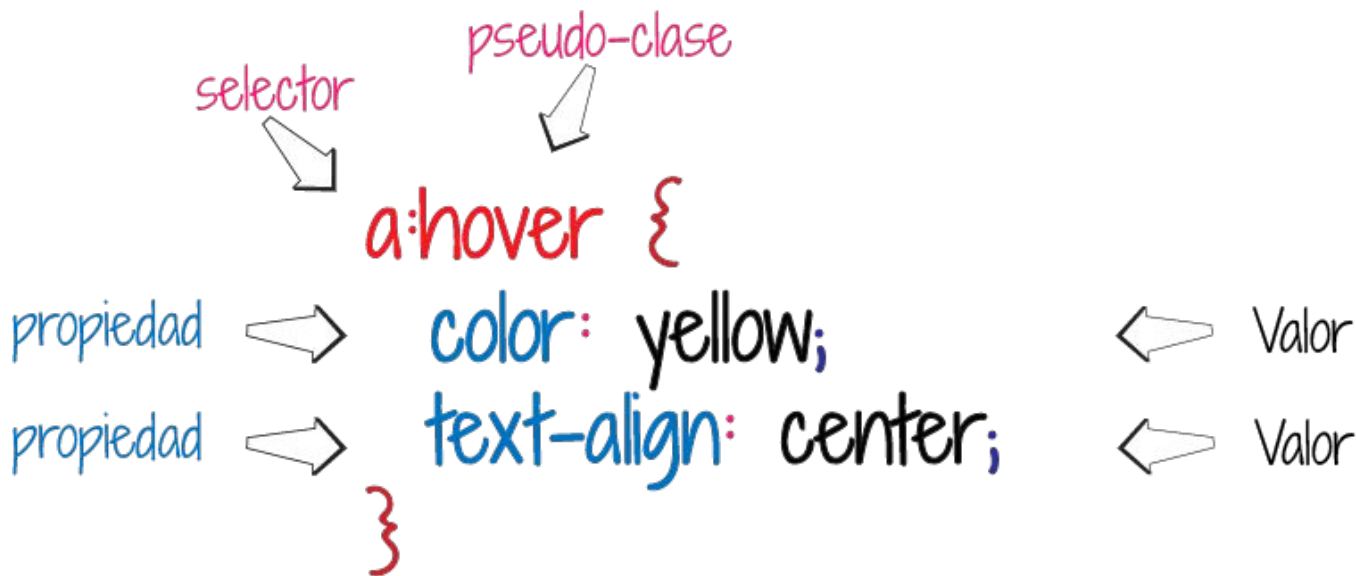


# Pseudoclases

Una **pseudoclase** en CSS es una palabra clave añadida a un selector que **especifica un estado especial del elemento** o elementos seleccionados.

Permiten **aplicar un estilo** a un elemento no sólo en relación con el contenido de la estructura del documento, **sino también en relación a los factores externos.**

# Definición



# pseudoclases de enlaces :link y :visited

:link	Aplica estilos cuando el enlace no ha sido visitado todavía.
:visited	Aplica estilos cuando el enlace ha sido visitado anteriormente.

```
a:link {  
  background-color: lightblue;  
}
```

Este párrafo contiene [un enlace](#) al párrafo siguiente.

Este párrafo es el destino del enlace anterior.

```
a:visited {  
  color: red;  
}
```

Este párrafo tiene un [enlace a sí mismo](#). Si el navegador reconoce que el enlace se ha visitado, en vez de en morado, se muestra en rojo.

# pseudoclases de dinámicas :hover y :active

:hover	Aplica estilos cuando pasamos el ratón sobre un elemento.
:active	Aplica estilos cuando estamos clickeando sobre el elemento.

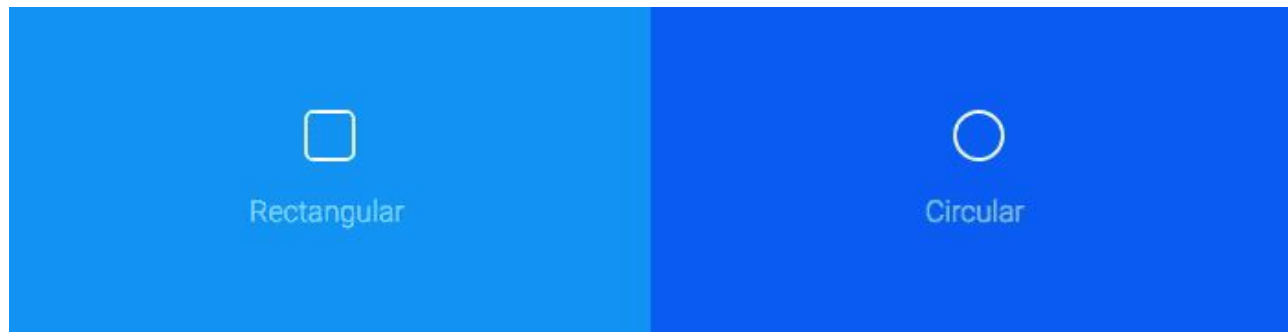




# pseudoclases de formulario :focus y :checked

:focus	Aplica estilos cuando el elemento tiene el foco.
:checked	Aplica estilos cuando la casilla está seleccionada.

Gradient border focus fun



# pseudoclasas a elementos hijos de distinto tipo

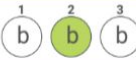
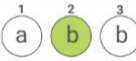
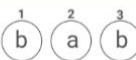
*Existen varias pseudoclasas que permiten hacer referencias a los elementos del documento HTML según su posición y estructura de los elementos hijos.*

`:first-child/:last-child` se utiliza para representar al primer o último elemento entre un grupo de elementos hermanos.

`:nth-child()` coincide con un elemento en función de su posición entre un grupo de hermanos.

Este es el primer párrafo <p> en una división que contiene tres párrafos.  
Este es el segundo párrafo <p> en una división que contiene tres párrafos.  
Este es el tercer párrafo <p> en una división que contiene tres párrafos.






```
div {  
  border: black 5px solid;  
  margin: 10px;  
  padding: 10px;  
}  
  
div p:first-child {  
  color: red;  
}
```

<code>&lt;b /&gt; &lt;b /&gt; &lt;b /&gt;</code>	<code>b:nth-child(2) {}</code>	
<code>&lt;a /&gt; &lt;b /&gt; &lt;b /&gt;</code>		
<code>&lt;b /&gt; &lt;a /&gt; &lt;b /&gt;</code>		 <p>* nth-child(2) is 'a' element, not 'b' element. So there is nothing to be selected.</p>

# pseudoclasas a elementos hijos por mismo tipo

`:first-of-type/:last-of-type` se utiliza para representar al primer o último elemento entre un grupo de elementos hermanos de la misma etiqueta.

`:nth-of-type()` coincide con un elemento en función de su posición entre un grupo de hermanos de la misma etiqueta.

First of Type Selector	<code>&lt;a/&gt;</code> <code>&lt;b/&gt;</code> <code>&lt;a/&gt;</code> <code>&lt;b/&gt;</code>	<code>b:first-of-type {}</code>	
Nth of Type Selector	<code>&lt;a/&gt;</code> <code>&lt;b/&gt;</code> <code>&lt;a/&gt;</code> <code>&lt;b/&gt;</code> <code>&lt;a/&gt;</code> <code>&lt;b/&gt;</code> <code>&lt;a/&gt;</code>	<code>a:nth-of-type(2) {}</code>	
		<code>a:nth-of-type(even) {}</code>	
		<code>a:nth-of-type(odd) {}</code>	
		<code>a:nth-of-type(2n+1) {}</code>	

\* 'n' is an every positive integer or zero value.

# Pseudoelementos

Son otra de las características de CSS que permiten hacer referencias a «comportamientos virtuales no tangibles», o lo que es lo mismo, se le puede dar estilo a elementos que no existen realmente en el HTML, y que se pueden generar desde CSS.

# pseudoelementos

`::first-letter` se utiliza para darle estilo a la primera letra de un texto.

`::selection` se utiliza para darle estilo al texto seleccionado.

`::first-line` se utiliza para darle estilo a la primera línea de un párrafo.

`::before` puede utilizarse para agregar algo antes del contenido de un elemento.

```
a::before{ content:" ✨";}
```

`::after` coloca contenido después de un elemento

```
a::after{ content:" 🐵";}
```

How to make the first letter of a paragraph float to the left in a web-page like in magazines and newspapers. So have fun learning the basics of CSS

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Consequatur beatae rerum officia ex molestias, distinctio voluptatibus quod tempore quo eius reiciendis reprehenderit veniam placeat natus laborum illum dolores quae aliquam.

TechOnTheNet.com has been providing helpful references, how-to's and FAQs since 2003. We focus on technologies such as Microsoft Access, Microsoft Excel, Microsoft Word, SQL, Oracle/PLSQL, MySQL, HTML, CSS, and the C Language.

✨ Inicio 🐵 ✨ Contacto 🐵

# Transiciones

Las transiciones permiten manejar cómo cambia el valor de una o más propiedades en un período de tiempo sobre un evento determinado.

Para crear un efecto de transición, hay que **especificar** dos cosas:

- ▶ La propiedad CSS a la que desea agregar un efecto.
- ▶ La duración del efecto.

# Propiedad `transition`

`transition-property:` para definir qué propiedad vamos a alterar. `all | none | <prop>`

`transition-duration:` para definir la duración de la transición. `s | ms`

`transition-delay:` definimos el tiempo a esperar antes de que se ejecute la animación. `s | ms`

`transition-timing-function:` curva de velocidad del efecto de la transición. `linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier() | step-end | steps()`

`transition:` shorthand property de las anteriores.

`transition: width 2s 800ms ease;`



# Veamos algunos ejemplos...

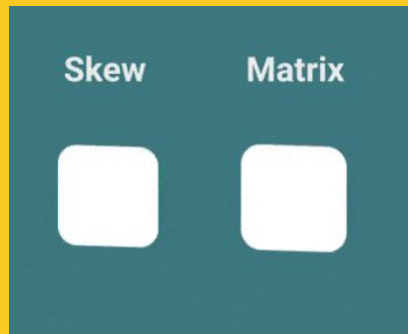
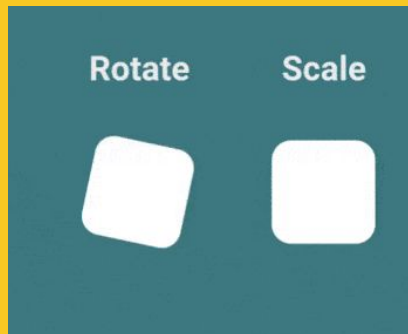




# Transformaciones

Permiten rotar, torcer, escalar o desplazar los elementos de nuestra página web

Son una de las características de CSS más interesantes y potentes que nos ayudan a convertir las hojas de estilo en un sistema capaz de realizar efectos visuales 2D y 3D.



# Propiedad `transform`

Las dos propiedades que nos sirven para **definir las transformaciones** son:

`transform`: propiedad **base** para cualquier transformación. **El punto de origen se ubica en el centro del elemento.**

`transform-origin`: nos permite **cambiar el punto de origen** de nuestra transformación.

**En esta clase nos centraremos en ver las transformaciones 2D más comunes.**

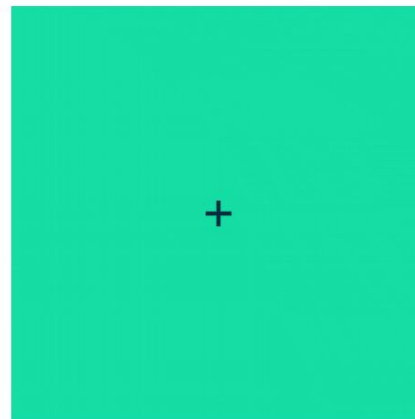


# scale()

**Modifica el tamaño de los elementos.** Esta función se establece con uno o dos valores, que representan la cantidad de escala que se aplica **en cada dirección**: `scale(x)` o `scale(x,y)`.

*Valores entre 0 y 1 achican y mayores a 1 agrandan.*

```
div {  
  /* 2,5 - X,Y */  
  /* 2 - X/Y */  
  transform: scale(2,2);  
}
```



# translate()

**Cambia la posición del elemento hacia la izquierda, derecha, arriba o abajo.** Esta función se establece con uno o dos valores: `translate(x)` o `translate(x,y)`.

```
div {  
  /*  
  Valores positivos mueven a la derecha/abajo.  
  Valores negativos mueven a la izquierda/arriba.  
  */  
  transform: translate(10, 30);  
}
```



# rotate()

Gira o rota los elementos en grados: `rotate(deg)`.

La unidad de medida es **deg** y puede tomar **valores positivos** (gira hacia la derecha) o **negativos** (gira hacia la izquierda) de 0 a 360°.

```
div {  
  transform: rotate(230deg);  
}
```



Rotate

# skew()

**Distorsiona** los elementos según el **ángulo en grados**. Esta función se establece con uno o dos valores: `skew(x)` o `skew(x,y)`.

La unidad de medida utilizada también es deg al igual que en rotate().

```
div {  
  transform: skew(20deg,10deg);  
}
```



# Animaciones

Las **animaciones** permiten animar la transición entre un estilo CSS y otro. A diferencia de los **transitions**, estos no se disparan frente a un evento determinado, si no que **comienzan desde que el sitio es cargado**.

Para utilizar animaciones, es necesario crearlas o definirlas previamente mediante la regla css **@keyframes**.

# Propiedades de Animaciones

Constan de dos componentes:

- Propiedades CSS de las animaciones, que definen el comportamiento de la misma.
- Un conjunto de fotogramas que indican su estado inicial y final, así como posibles puntos intermedios en la misma.

Shorthand property

```
animation: mi-animacion 5s linear 0.2s  
infinite normal;
```

```
animation-name: <nombre de la  
animación>;
```

```
animation-duration: <duración>s | ms;
```

```
animation-delay: <retardo>s | ms;
```

```
animation-timing-function: efecto de  
transición. linear | ease | ease-in |  
ease-out | ease-in-out |  
cubic-bezier() | step-end | steps();
```

```
animation-direction: normal | reverse  
| alternate;
```

```
animation-iteration-count: <veces> |  
infinite;
```



# @keyframes

Regla CSS donde vamos a **crear una animación** para **utilizarla después** todas las veces que lo necesitemos.

```
@keyframes agrandar {  
  from {  
    width: 100px;  
    height: 100px;  
    background-color: orange;  
  }  
  
  to {  
    width: 500px;  
    height: 500px;  
    background-color: crimson;  
  }  
}
```

```
@keyframes nombre-animation {  
  time-selector {  
    propiedad : valor ;  
    propiedad : valor  
  }  
}
```

# No te olvides de dar el presente

## **Recordá:**

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

**Todo en el Aula Virtual.**

# Gracias