Anteriormente

- Etiquetas semánticas
- Sombras
- Bordes redondeados
- Uso de webfonts
- Uso de webicons
- Responsive web design
- Etiqueta viewport
- Pseudo selectores
- Uso de @mediaqueries
- Colocar <video>
- Contenido menos relevante <aside>
- SEO
- Organización del código html y css
- Recursos y herramientas
- Uso de Github

Animaciones con CSS

Animaciones con CSS

Si las usamos correctamente, las animaciones y transiciones le dan vida e interacción a internet.

Podemos hacer:

- ANIMACIONES
- TRANSICIONES

(ejemplo 1)

Animaciones con CSS

Tomemos el logo de nuestro diseño.
Las transformaciones pueden ser de rotación, escalabilidad, deformación, etc.

```
header #logo
{
    transform: rotate(90deg);
}
```

Animaciones con CSS

Las animaciones forman parte avanzada de CSS3. Cada navegador las interpreta de manera diferente, es por eso que si queremos que se vea bien en los browsers, tenemos que usar prefijos para que los tome.

Prefijos dependiendo de cada motor de renderización:

Chrome > Blink > prefijo: webkit
Firefox > Gekko > prefijo: moz
Opera > Prefijo: webkit (Antes era sólo o)
IE > Prefijo: ms

```
header #logo
{
    -webkit-transform: rotate(90deg);
}
```

Animaciones con CSS

Para hacer una animación tenemos que aplicarle un evento.

Al logo le vamos a aplicar un evento cuando hagamos **hover** sobre él.

```
header #logo
{
    border-right: 1px solid white;
    margin-right: 0.5em;
}

header #logo img
{
    margin-right: 0.5em;
    padding-right: 0.5em;
    width: 60px;
    -webkit-transform: rotate(90deg);
}
```

Animaciones con CSS

:**hover** sobre el logo.

```
header #logo img
{
    margin-right: 0.5em;
    padding-right: 0.5em;
    width: 60px;
}

header #logo img:hover
{
    -webkit-transform: rotate(90deg);
}
```

Veamos qué hace en nuestra web.

Transición

Para agregar más instancias y crear una animación, necesitamos una transición.

Podemos aplicarle parámetros como duración, velocidad, etc.

Utilizamos otro prefijo: **transition**

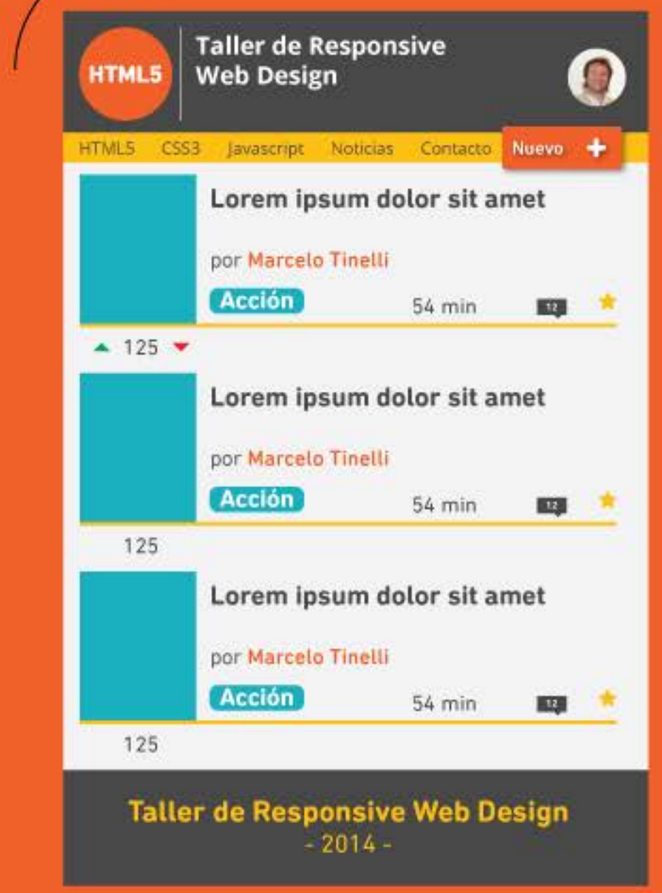
```
header #logo img
{
    margin-right: 0.5em;
    padding-right: 0.5em;
    -webkit-transition: 1s
    width: 60px;
}
```

Veamos qué hace en nuestra web.

Transición

Definimos:

- A qué elemento animar
- En qué evento ocurrirá la animación
- Qué tipo de animación
- Con qué transición se hará



Transición

Hagamos un poco más compleja nuestra animación.

```
header #logo img
{
    margin-right: 0.5em;
    padding-right: 0.5em;
    -webkit-transition: .5s;
    width: 60px;
}
header #logo img:hover
{
    -webkit-transform: rotate(360deg);
}
```

Transición

Nuestro menú animado.

```
nav ul li a:hover
{
    background: white;
}
```

Le modificamos el padding para que quede mejor centrado visualmente.

```
nav ul li a
{
    color: #474747;
    display: block;
    padding: 0.5em 0;
    text-decoration: none;
}
```

Transición

Le añadimos una transición a nuestro elemento.

```
nav ul li a
{
    color: #474747;
    display: block;
    padding: 0.5em 0;
    text-decoration: none;
    -webkit-transition: .3s;
}
```

¿POR QUÉ .3s?

Transición

Tomemos ahora "Nuevo +"
Esto afecta a toda la caja.

```
nav ul #nuevo_nav a:hover
{
    background: red;
}
```

Seleccionemos sólo el after.

```
nav ul #nuevo_nav a:hover:after
{
    color: red;
}
```

Escalamos.

```
nav ul #nuevo_nav a:hover:after
{
    font-size: 1.2em;
}
```

(Si fuera un elemento, podríamos usar:
-webkit-transform: scale(1.2);)

Transición

Para acomodar la caja contenedora y que sólo se escale el + debemos hacer algunos cambios.

```
nav ul #nuevo_nav a:hover:after
{
    font-size: 1.2em;
    position: absolute;
    left: 0;
    top: 0;
}
```

Nuestro + se pondrá relativo a su contenedor más cercano. Con left y top en cero veremos sobre qué caja.

Transición

Seguimos acomodando un poco mejor todo.

```
nav ul #nuevo_nav a
{
    background: #F16028;
    border-radius: 0.25em;
    box-shadow: rgba(0,0,0,0.5) 2px 2px 5px;
    color: white;
    padding: 0.8em 2em 0.8em 0.5em;
    text-decoration: none;
    text-shadow: rgba(0,0,0,0.5) 2px 2px 5px;
}
```

Esto nos dará aire para reubicar el +.

Transición

Dejamos el **hover** sólo el aumento de tipografía y al **a** le generamos un absolute.

```
nav ul #nuevo_nav a:after
{
    position: absolute;
    right: 0;
    top: 0;
}
nav ul #nuevo_nav a:hover:after
{
    font-size: 1.2em;
}
```

De esta forma, nos queda acomodado el "Nuevo" y tenemos móvil al +.

Transición

Posicionamos finalmente el +

```
nav ul #nuevo_nav a:after
{
    position: absolute;
    right: .5em;
    top: .9em;
}
```

Transición

Posicionamos finalmente el +

```
nav ul #nuevo_nav a:after
{
    position: absolute;
    right: .5em;
    top: .9em;
}
```

Padding

```
nav ul #nuevo_nav a
{
    background: #F16028;
    border-radius: 0.25em;
    box-shadow: rgba(0,0,0,0.5) 2px 2px 5px;
    color: white;
    padding: 0.8em 2.5em 0.8em 0.5em;
    text-decoration: none;
    text-shadow: rgba(0,0,0,0.5) 2px 2px 5px;
}
```

Resumen

Las animaciones con CSS3 son la mejor manera de darle vida a sus proyectos web.

Básicamente lo que hacemos es cambiar una propiedad CSS durante un tiempo específico determinado.

Sin la ayuda de **Javascript**, lo único que podemos aplicar por ahora las animaciones es a los estados **hover** y **active**.

Estamos limitados con eso, pero con un nivel básico de CSS podemos implementar fácilmente y hacer un sitio más cool. Con muy poco, podemos lograr ese impacto inicial al usuario.

Animaciones / Transiciones
Las transiciones nos permiten ir de un punto inicial A, hacia el B sin nada intermedio.
Las animaciones pueden tener múltiples keyframes.

Uso recomendado
- Links con fading al hacer :hover

- Fading en el color de fondo de botones al hacer :hover

- Mover y escalar elementos al hacer :hover

Nota blog

CSS3: Animations vs. Transitions

http://www.kirupa.com/html5/css3_animations_vs_transitions.htm

Stylus

Para instalar **Stylus** primero tenemos que instalar **Node.js**.

http://nodejs.org/

Node.js es un sistema de javascript del lado del servidor.
Lo vamos a usar para correr Stylus.
Stylus está programado con Node.js.

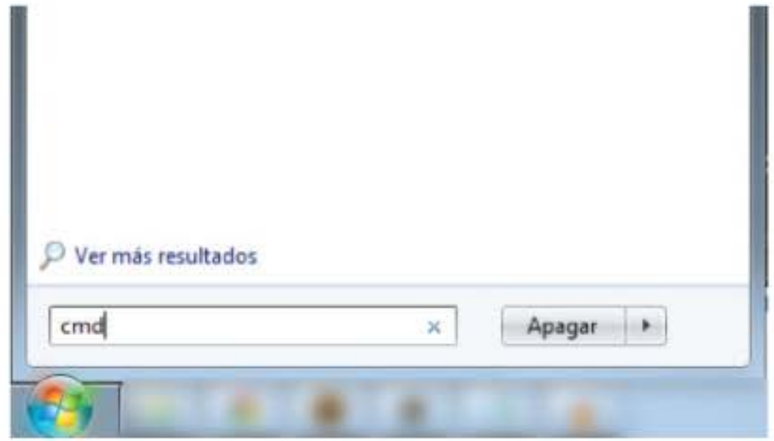
INSTALL

Stylus

Una vez descargado e instalado **Node.js**,
tenemos que abrir una terminal del SO.

En Windows es CMD.
En Mac, es Terminal.

En Linux... los que saben Linux, saben.



Stylus

¿Por qué tenemos que usar Stylus?

Es la mejor forma de escribir código CSS.
Ya vimos que al escribir CSS de forma tradicional, tiene una sintaxis muy engorrosa con comas, punto y comas, corchetes, paréntesis. No permite el uso de variables.

Stylus simplifica todo y lo transforma compatible con todos los navegadores.

Es una aplicación escrita en Javascript, por eso es que para correr aplicaciones Javascript, tenemos que tener instalado Node.js

Stylus

Para verificar la versión de Node.js en la terminal tipear:
node -v

Luego para instalar el
Node Package Manager tipear:
npm install stylus -g

(cuando hagan esto les va a salir un error)

En **Mac** hay que anteponer permisos de administrador:
sudo npm install stylus -g
(Va a pedir el password de su máquina si es que está con clave)
Reiniciar la terminal.

Para verificar la versión de Stylus en la terminal tipear:
stylus -V

En **Windows**:
npm install stylus -g
Reiniciar la terminal.

Para verificar la versión de Stylus en la terminal tipear:
stylus -V

Stylus

En Sublime creamos un nuevo archivo suelto junto con los demás y lo nombramos:
stylus.styl

Dentro del archivo .styl ya guardado tipeamos respetando el indentado:

```
body
    background red
```

Salvamos.

En la terminal, compilamos el archivo .styl

```
stylus stylus.styl -c -w
```

Veamos qué ocurre... jeje