



# Coding Bootcamp

## Sprint 1



# Temario



# Temario

- Sombras.
- Gradientes.
- vh y vw.
- calc.
- Variables.
- Transiciones.
- Transformaciones.
- Animaciones.
- SASS → Lo vamos a dejar para más adelante.



# Sombras

# Sombras

- La propiedad `box-shadow` permite darle una sombra a un elemento.
- Se pueden usar a partir de Internet Explorer 9.
- Más info [aquí](#).

```
#box {  
  box-shadow: 8px 5px 30px #111111CC;  
}
```

offset-x

offset-y

blur-radius

color

Opcionalmente se puede agregar la keyword `inset` para indicar que la sombra es hacia dentro del elemento (en lugar de hacia afuera), dando una sensación de relieve.



# Gradientes



# Gradientes

- `linear-gradient` es una función CSS para crear *backgrounds* con un efecto degradado. Técnicamente los gradientes son imágenes de fondo.
- Se pueden usar a partir de Internet Explorer 10 (parcialmente).
- Crear gradientes no suele ser fácil, por lo que hay herramientas para simplificar la tarea como: <https://cssgradient.io>.
- Más info [aquí](#).

```
#box {  
  background: linear-gradient(#e66465, #9198e5);  
}
```





vh y vw





# vh y vw

- `vh` y `vw` son dos **unidades de medida** en CSS (así como lo son `px`, `rem` y `%`).
- Son relativas al tamaño del **viewport** (tamaño de la ventana).
- $1\text{vh} = 1\%$  del alto de la ventana.
- $1\text{vw} = 1\%$  del ancho de la ventana.
- Se pueden usar a partir de Internet Explorer 9.
- Es útil para generar diseños como [este](#).
- Más info [aquí](#).



calc



# calc

- Es una función (dentro de CSS) que permite asignarle valores a propiedades (que resultado de una operación matemática).
- Los operadores que se pueden usar son: +, -, \* y /.
- Es importante dejar un espacio antes y después de un operador.
- Se puede usar a partir de Internet Explorer 9.
- Más info [aquí](#).

```
#box {  
    width: calc(100% - 30px);  
}
```



# CSS Variables



# CSS Variables (Custom Properties) (1/2)

- Son entidades que permiten **guardar valores (CSS)** que luego se pueden reutilizar en el documento.
- No funciona en ninguna versión de Internet Explorer. Ver [Can I Use](#).
- El nombre de una variable siempre comienza con **dobles guión**: `--`
- Los nombres son **case-sensitive**.
- Las variables se pueden declarar en cualquier selector, pero lo más usual es colocarlas en el selector `:root` para que queden disponibles en todo el documento.
- Luego, para hacer uso de la variable es necesario llamar a la función `var`.
- Más info [aquí](#).



# CSS Variables (Custom Properties) (2/2)

Ejemplo de declaración de una variable:

```
:root {  
  --hack-academy-orange: #ef6c00;  
}
```

Ejemplo de uso de la variable (más abajo, incluso en otro archivo CSS):

```
p {  
  color: var(--hack-academy-orange);  
}
```



# Transiciones



# Transiciones (1/2)

- Una transición CSS es una forma de controlar la **velocidad** con la que **cambia alguna propiedad CSS** de un elemento. Por defecto, cuando se cambia una propiedad (ej: el color de un párrafo), el cambio es instantáneo.
- El cambio de una propiedad se puede dar vía JS o por una psuedo clase.
- Se pueden usar a partir de Internet Explorer 10. Ver [Can I Use](#).
- Más info [aquí](#).

```
#box {  
    transition-duration: 2s;  
}
```





## Transiciones (2/2)

- Se puede establecer que la transición suceda sólo para determinada propiedad (ej: para el `font-size`).
- Se puede especificar cómo deben cambiar los valores de la propiedad durante la transición (ej: para lograr un efecto de aceleración o frenado). *Timing Function*.
- Se puede establecer un *delay* para que la transición arranque más tarde.

```
#box {  
  transition-property: font-size;  
  transition-timing-function: ease-in;  
  transition-delay: 1.2s;  
}
```



# Transformaciones



# Transformaciones

- Una transformación permite trasladar, rotar, escalar o sesgar (skew) un elemento HTML. Ver [ejemplos](#).
- Se puede usar a partir de Internet Explorer 9 (pero parcialmente). Ver [Can I Use](#).
- Más info [aquí](#).

```
#box {  
    transform: translate(100px, 2rem);  
  
    transform: rotate(20deg);  
  
    transform: skew(45deg, 45deg);  
  
    transform: scale(1.5, 2);  
}
```

Con la propiedad `transform-origin` se pueden cambiar los ejes de coordenadas desde donde se realiza la transformación



# Animaciones



# Animaciones (1/3)

- Una animación CSS permite establecer cambios en las propiedades de un elemento HTML, estableciendo una serie de **fotogramas**.
- Primero hay que definir una animación con la palabra `@keyframes`. Luego hay que asignar dicha animación a uno o más elementos.
- Se puede usar a partir de Internet Explorer 10. Ver [Can I Use](#).
- Más info [aquí](#).
- Como no siempre es fácil crear animaciones, existen herramientas que simplifican el proceso como, por ejemplo:
  - <https://animista.net>.
  - <https://daneden.github.io/animate.css>.

# Animaciones (2/3)

Ejemplo de **definición** de una animación:

```
@keyframes unaAnimacion {  
  to {  
    width: 0px;  
    background-color: yellow;  
  }  
  from {  
    width: 100px;  
    background-color: red;  
  }  
}
```

En este caso sólo se definieron dos fotogramas, el inicial (0%) y el final (100%).

También se podrían haber definido fotogramas intermedios. Ej: 60% { width: 200px }



# Animaciones (3/3)

Ejemplo de uso de una animación:

```
#box {  
  animation-name: unaAnimacion;  
  animation-duration: 3s;  
  animation-iteration-count: infinite;  
}
```