Curso de Preprocesadores CSS

# Conceptos básicos de CSS

CSS significa **Cascading Style Sheets** o su traducción al español Hojas de Estilo en Cascada.

Asignar reglas CSS a un documento HTML se puede hacer de varias formas:

* Colocando las reglas en un documento \*.css y relacionarlo al HTML usando la etiqueta
* Colocando en el atributo style de cada elemento HTML las reglas para cada etiqueta
* Colocando los estilos dentro de una etiqueta style dentro del head del documento.

Como su nombre lo dice siempre lee en cascada de arriba hacia abajo sobrescribiendo valores excepto los dados por el atributo style.

**Un estilo CSS está formado por: selector, declaración, propiedad y valor.**

# Selectores de CSS

Los selectores nos sirven para seleccionar los diferentes elementos en una página web y aplicar estilos

Existen los siguientes tipos de selectores:

* **Selector universal**\* Sirve para agregar estilos a todos los elementos de la página. Normalmente se utiliza para hacer “reset” de estilos.
* **Selector etiqueta**. Aplica estilos a todos los elementos de ese tipo (p, h1, header etc.)
* **Selector id**. Aplica estilos a un elemento único con ese Id, se recomienda hacer buen uso de este selector.
* **Selector clase**. Aplica estilos a todos los elementos con esa clase (el que más vas a usar).
* **Selector anidado**. Aplica estilos a un elementos descendientes de otros elementos (no necesario que sea hijo directo).
* **Selector hijo >**. Aplica estilos a los elementos que sean hijos directos de otros.
* **Selector adyacente +**. Aplica estilos al elemento adyacente.
* **Selector de atributo input[type=“number”]**. Aplica estilos al elemento con el atributo especificado.

La prioridad de un selector se determina por la suma de su contenido:

* ID = 100
* Clase = 10
* Etiqueta = 1

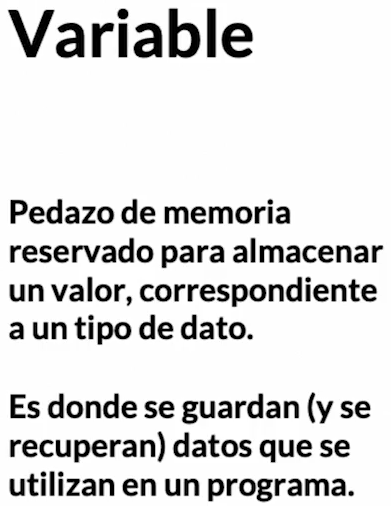
A mayor la suma, mayor prioridad. Esto esta por sobre la cascada

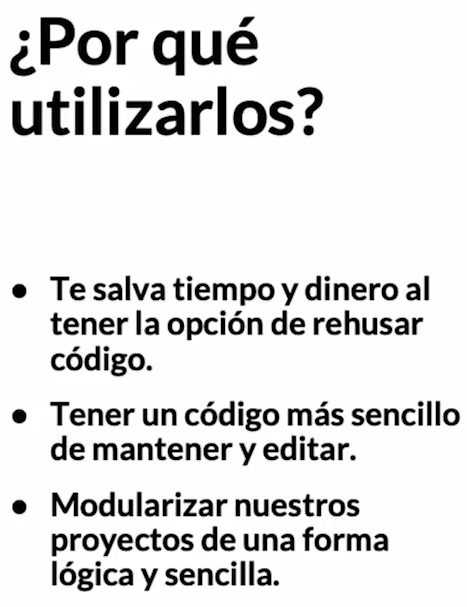
**!important** es un valor especial tiene un valor de un millón, nunca lo uses a menos que sea tu única opción como cuando no tienes acceso al código fuente.

# Introducción a los Preprocesadores

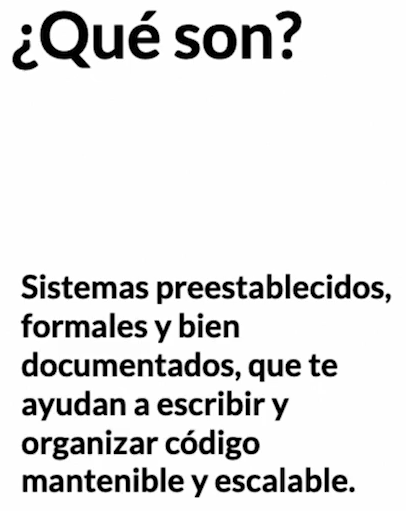
Un preprocesador es una herramienta que nos permite escribir pseudocódigo de forma modular, más fácil de rehusar, leer, y mantener. pseudocódigo que después será convertido a CSS o HTML estándar que el navegador entiende.

Gracias a los preprocesadores podemos extender las características de CSS y HTML al nivel de otros lenguajes de programación, permitiéndonos usar características como variables, funciones y mixins.





# Metodologías para estructurar código

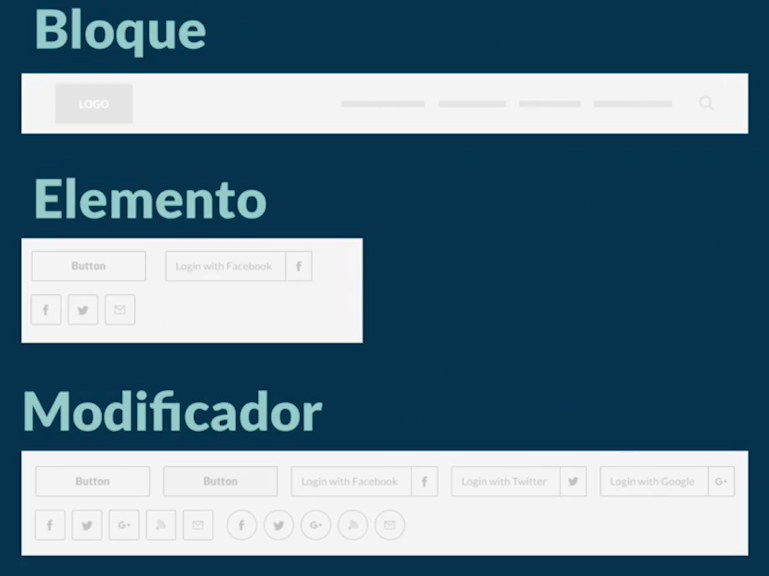
 

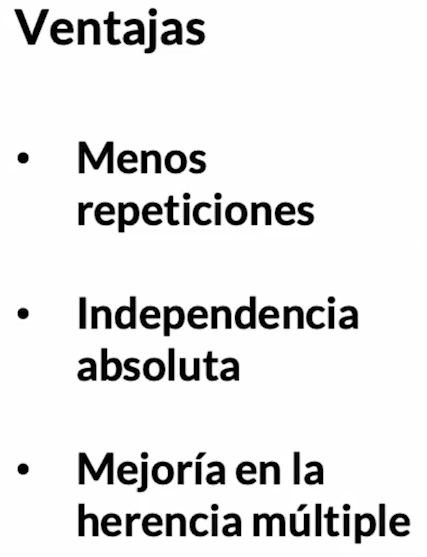
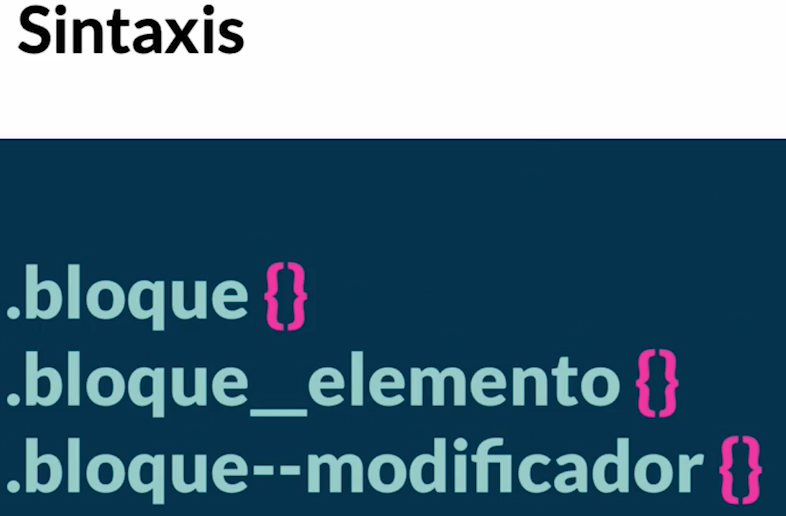
# Introducción a BEM

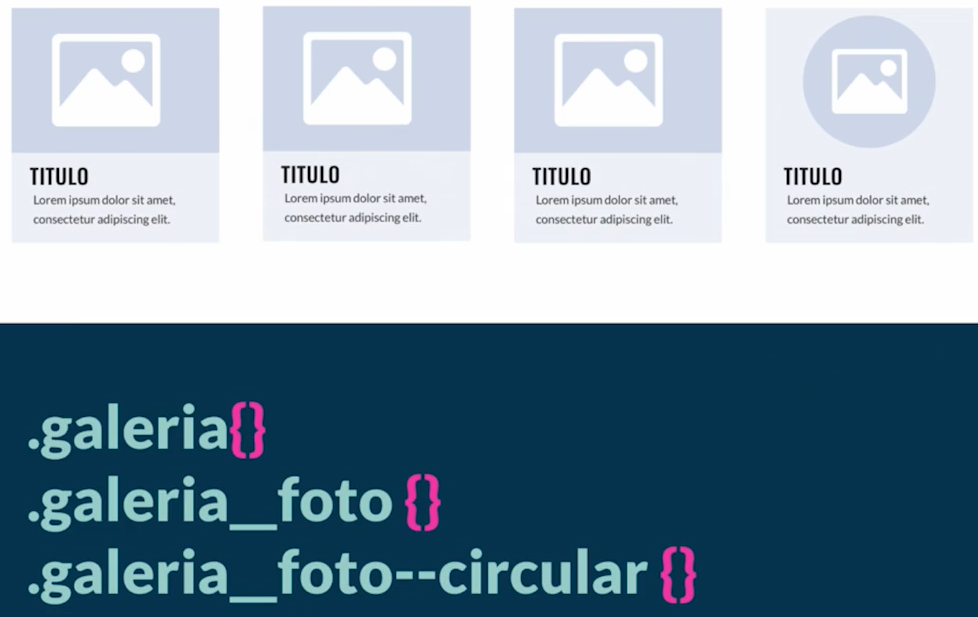
BEM es la metodología que vamos a usar a lo largo del curso. El objetivo de BEM es dividir lógicamente las piezas de las que se compone una web.

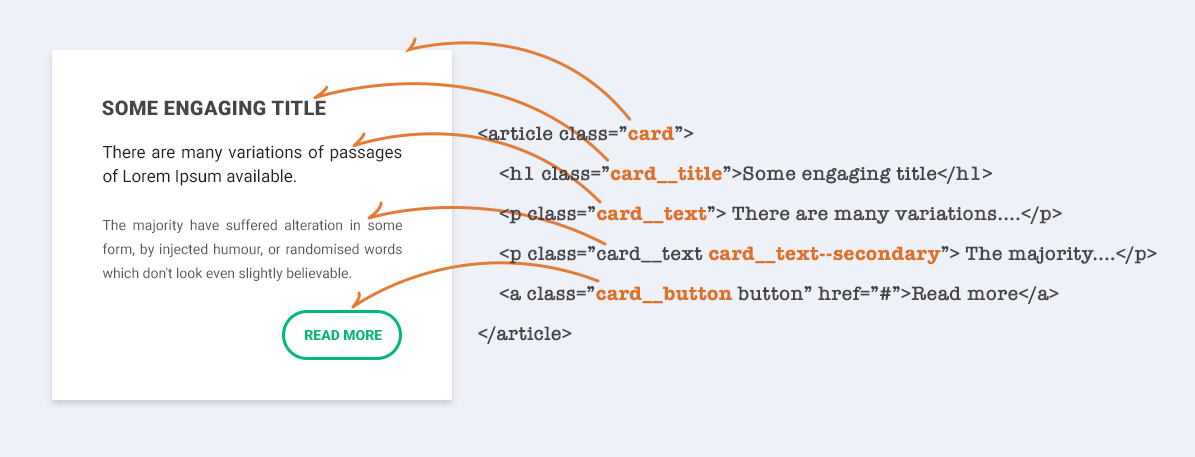
BEM establece que debemos usar clases para nuestros selectores, clases que se dividen en:

* **Bloques**. Los bloques son nuestros contenedores más grandes que a su vez contienen elementos u otros bloques.
* **Elementos**. Los elementos siempre forman parte de un bloque, normalmente son los botones, textos, imágenes etc.
* **Modificadores**. Los modificadores se usan para establecer estilos diferentes a un mismo bloque o elemento.







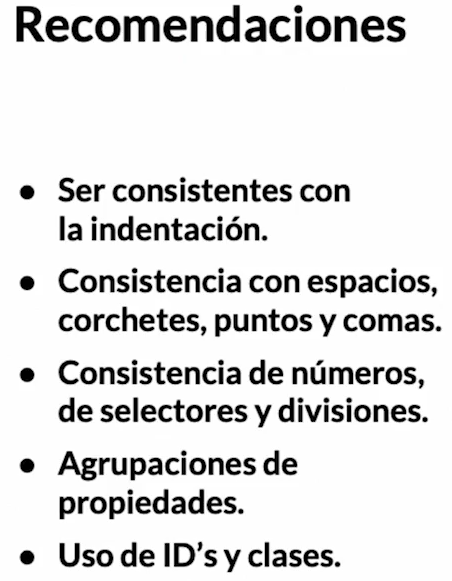


# Guías para creación y mantenimiento de código

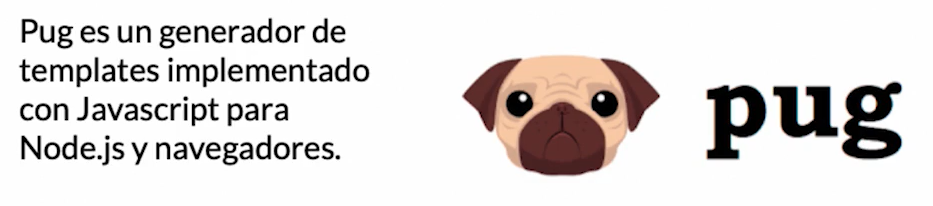
La meta de tener una guía de código es hacer que luzca como si una sola persona lo haya escrito para que se entendible por todo el equipo.

Para nuestro proyecto **PlatziGames** vamos a tener una guía en la que definimos:

* Ser consistentes con la indentación.
* Consistencia con espacios, corchetes, puntos y comas.
* Consistencia de números, de selectores y divisiones.
* Agrupaciones de propiedades.
* Uso de ID’s y clases.



# Introducción a Pug



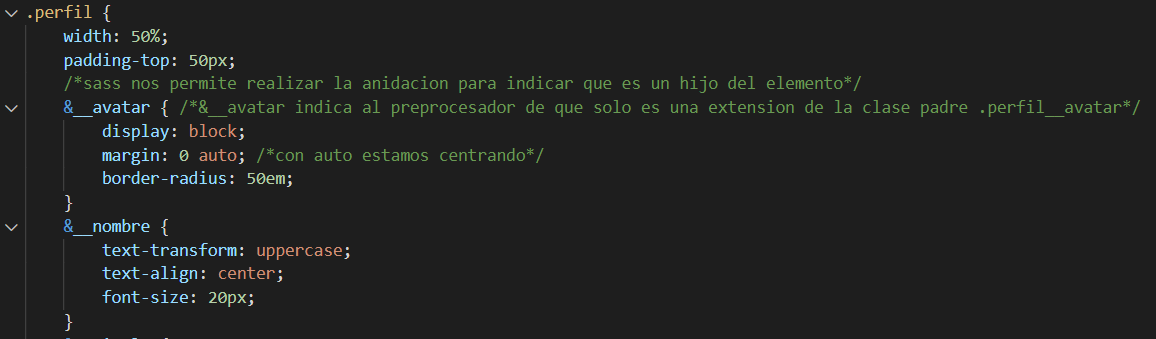
# Introducción a Sass



Sass nos permite usar variables, reglas anidadas, mixins y funciones.

La razón de que en SASS usemos la extensión ‘.scss’ es porque esta nos permite usar una sintaxis muy parecida a css.

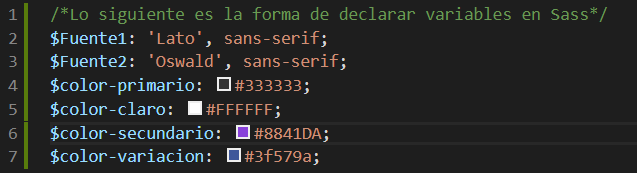
La otra opción es usar SASS con la extensión ‘.sass’ la única diferencia es que con esta extensión podremos omitir las llaves ‘{}’ y los punto y coma ‘;’ después de cada valor, esta sintaxis interpretará los atributos y valores por medio de la identación.



# Variables

En las variables almacenamos datos que se puede reutilizar en todas nuestras hojas de estilos. Así evitamos tener que escribir lo mismo una y otra vez cuando se realiza algún cambio, ya que sólo vamos a modificar el valor de la variable y se aplicará a todos los lugares donde sea usada.

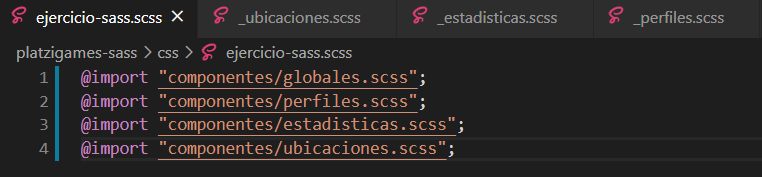
Comúnmente almacenamos en variables las guías de estilo de nuestro sitio, como pueden ser los colores y fuentes.

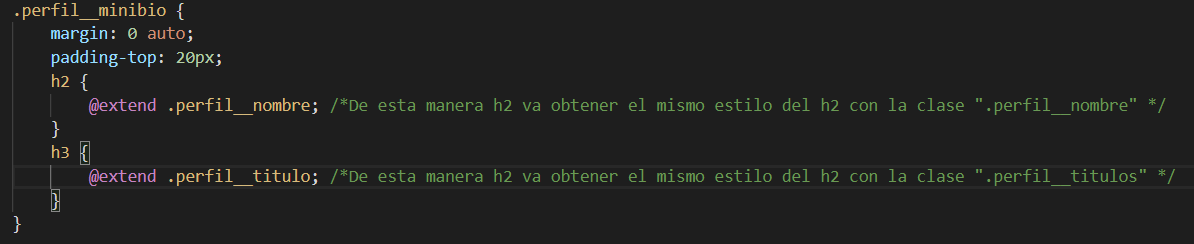


# Imports y Extends

**Import** nos permite escribir código modular separando en diferentes archivos para después importarlos todos en uno solo y tener una base código mucho más ordenada.

**Extends** sirve para insertar los estilos de un selector en otro.

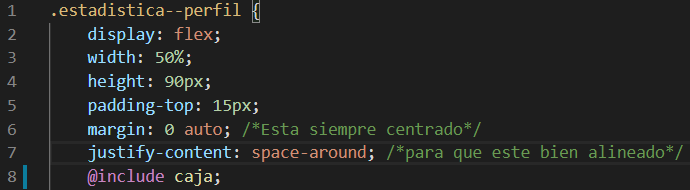
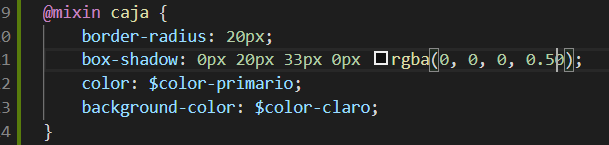




# Mixins

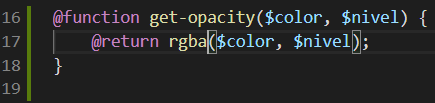
Su finalidad es ofrecer una funcionalidad que pueda ser reutilizada en otras clases pero que no está pensada para usarse de forma autónoma. Nos permite crear bloques reusables de código que cambian su resultado dependiendo del parámetro que enviemos.

Con los mixin logramos escribir menos código, produciendo un código más claro, más expresivo y sobre todo más fácil de mantener.



# Funciones

La diferencia entre mixins y funciones es que las funciones por general hacen cálculos y regresan un resultado que es usado como valor de alguna propiedad.



# **Condicionales y Loops**

Un **condicional** nos permite evaluar cierta condición y bifurcar entre dos caminos dependiendo de si se cumple o no.

Un **loop** es un fragmento de código que va a ejecutar de forma repetitiva hasta que cumpla una condición.

