

Metodologías para la organización y el preprocesador Sass

Metodología SMACSS

***Implementar una interfaz
de usuario web utilizando
buenas prácticas
en el manejo de estilos
para brindar un aspecto
visual e interacciones
acordes a lo requerido***

- Unidad 1:
Metodologías para la organización
y el preprocesador Sass
- Unidad 2:
El modelo de cajas y el Layout
- Unidad 3:
Utilizando Bootstrap como
Framework CSS



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Describe las características principales de la metodología SMACSS para la organización y modularización de estilos.*

¿A qué nos referimos
cuando hablamos de
“modularizar” nuestros
estilos?

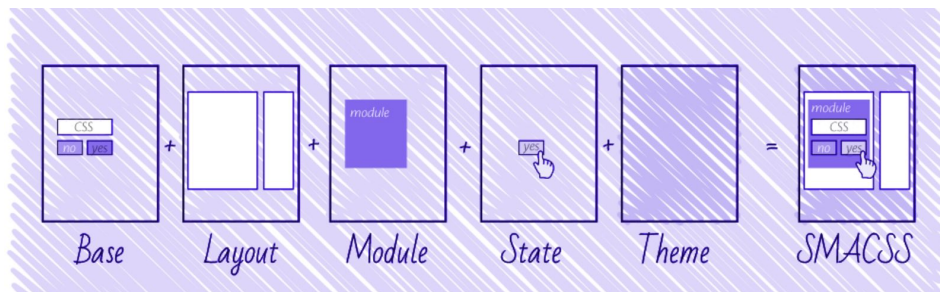


/* Metodología SMACSS */

En qué consiste esta metodología

Podemos definir SMACSS como una guía de estilo o una serie de pautas útiles que nos enseñan cómo escribir CSS escalable y entendible.

Para lograr esto, SMACSS propone cinco reglas de categorización:



Fuente: [Medium](#).

- **Base**: Son los estilos por defectos de los elementos.
- **Layout**: Es la estructura general del sitio web.
- **Module**: Los elementos reutilizables de la web.
- **State**: Es el estado de cada módulo.
- **Theme**: Los colores y el estilo de los módulos.

Categorías de elementos SMACSS

Base

Los estilos “**base**” definen el aspecto visual que debería tener algún elemento independiente su ubicación dentro de la página, es decir, estilos predeterminados.

En este estilo base, se recomienda solo incluir **selectores de elementos**, es decir, no utilizar selectores de clase o id a menos que sea necesario.



```
1  html {  
2      margin: 0;  
3      font-family: 'Courier New', Courier, monospace;  
4  }  
5  
6  a {  
7      color: #729E2E;  
8  }  
9  
10 button {  
11     color: #2B94D5;  
12     border: 1px solid #ccc;  
13 }
```

Categorías de elementos SMACSS

Layout



```
19 #header {  
20     background: #fcfcfc;  
21 }  
22  
23 #header .l-right {  
24     float: right;  
25 }  
26  
27 #header .l-align-center {  
28     text-align: center;  
29 }
```

Los estilos “**layout**” definen la estructura del sitio web, es decir, dividirán nuestra página web en diferentes secciones, como header, sidebar, footer, etc.

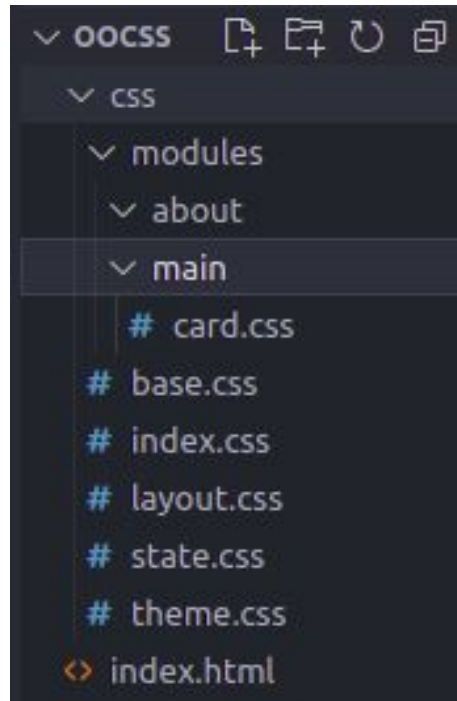
SMACSS recomienda añadirle un prefijo con la letra “l” para distinguirlos de los estilos de módulos.

Categorías de elementos SMACSS

Module

Module o módulos, son fragmentos o bloques de código CSS más pequeños que los elementos de layout, y se pueden reutilizar en la página.

Una **buena práctica** es tener una carpeta independiente para los módulos, ya que a medida que crezca el proyecto nos será más fácil organizarlo, aunque esto no es de carácter obligatorio



Categorías de elementos SMACSS

Module

Dado el ejemplo anterior, teníamos una card que puede ser un módulo por sí solo.

Para estructurarlo en CSS deberíamos tener una clase `.card` que pueda tener elementos secundarios como imagen, título y texto.

Para mantenerlos en el mismo módulo necesitamos prefijar los elementos secundarios (similar a los modificadores de BEM).

```
1  .card {
2      width: 250px;
3      height: 400px;
4      background: #ccc;
5      /* ..... */
6  }
7
8  .card--image {
9      width: 100%;
10     height: 100%;
11     /* .... */
12 }
13
14 .card--title {
15     font-size: 24px;
16     color: #35393B;
17 }
18
19 .card--text {
20     font-size: 16px;
21     color: #555A5D;
22 }
```

Categorías de elementos SMACSS

State

Una vez definido los estilos del layout y de cada módulo, debemos crear los estilos del estado.

Una regla de estado se aplica cuando un elemento **tiene varios estados**. Por ejemplo, una alerta de un formulario que indica si el mensaje fue enviado correctamente ☒ o si ocurrió un error inesperado ☐.

Para ambos casos, debemos representar diferentes estilos para cada estado.




Categorías de elementos SMACSS

State

Al igual que añadimos prefijos para los estilos de layout, para el estado también debemos usar prefijos, esto nos ayudará a distinguirlos y darle la prioridad en la cascada de CSS.

En este caso el prefijo es “is”, por ejemplo:



```
15 <header id="header">
16   <ul class="nav">
17     <li class="nav--item">Nosotros</li>
18     <li class="nav--item is-selected">Contacto</li>
19   </ul>
20 </header>
```




```
24 .nav--item.is-selected {
25   color: #fff;
26 }
```

Categorías de elementos SMACSS


Theme

Esta categoría se usa para definir los estilos de los colores principales, sombras, formas, bordes y más.

La mayoría de estos elementos se repetirán en el sitio web y no queremos redefinirlos cada vez que los creamos, sino que se crearán clases únicas que se agregarán al elemento que queramos.



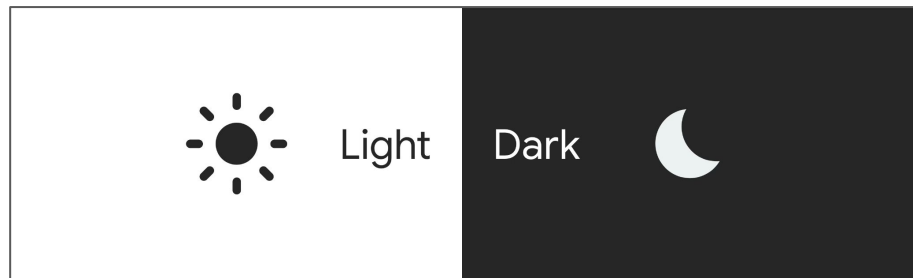
```
22 <button class="button-large">Like</button>
```



```
28 .button-large {  
29     width: 60px;  
30     height: 60px;  
31 }
```

Categorías de elementos SMACSS

Theme



Fuente: web.dev.

Es sumamente importante **no confundir estas reglas con la categoría base**, ya que estas últimas hacen referencia a los estilos que restablecen los estilos predeterminados del navegador.

Mientras que las reglas de tema se aplican al estilo final, un ejemplo claro de esto son los botones que permiten cambiar el tema de color desde light a dark

Escribe en el chat:

Un ejemplo de una regla de
la categoría state aplicada
a un elemento HTML



Ejercicio guiado - Aplicando SMACSS

{desafío}
latam_



Ejercicio guiado

Aplicando SMACSS

¿Recuerdan al cliente que nos pidió reorganizar el código en el ejercicio anterior?

Quedó muy feliz con su trabajo y ahora quiere que organicemos el código de la **página de posts de su blog**, para eso debemos

- Organizar el código CSS aplicando la metodología SMACSS.
- En un único archivo .css, estructurar los estilos según las **5 categorías** de la metodología.

{desafío}
latam_



Debes apoyarte del archivo de apoyo “Metodología SMACSS”

Blog FrontEnd

Posts



Semántica HTML

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Harum accusamus laboriosam vitae iste nemo voluptate facilis repellat deserunt voluptatibus repellendus.

[#Html #FrontEnd](#)



Flexbox

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Harum accusamus laboriosam vitae iste nemo voluptate facilis repellat deserunt voluptatibus repellendus.

[#CSS #FrontEnd](#)



JS

Herencia

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Harum accusamus laboriosam vitae iste nemo voluptate facilis repellat deserunt voluptatibus repellendus.

[#JavaScript #FrontEnd](#)




Formularios HTML


Lorem ipsum dolor sit, amet consectetur adipiscing elit. Harum accusamus laboriosam vitae iste nemo voluptate facilis repellat deserunt voluptatibus repellendus.

[#Html #FrontEnd](#)

Categorías

 **HTML**

 **CSS**

 **JavaScript**

Blog FrontEnd

Todos los derechos reservados - 2022

Si haces memoria... ¿Has
aplicado alguna de estas
reglas en los proyectos del
módulo anterior?



Resumen

A

Podemos definir **SMACSS** como una guía de estilo o una serie de pautas útiles que nos enseñan cómo escribir CSS escalable y entendible.

B

Los estilos “layout” **definen la estructura del sitio web**, es decir, dividirán nuestra página web en diferentes secciones, como header, sidebar, footer, etc.

C

Una buena práctica es tener una **carpeta independiente para los módulos**, ya que a medida que crezca el proyecto nos será más fácil organizarlo

D

No confundir las reglas de theme con la categoría base, ya que estas últimas hacen referencia a los estilos que restablecen los estilos predeterminados del navegador.

Mientras que las reglas de tema se aplican al estilo final



Próxima sesión...

- *Desafío evaluado*

{desafío}
latam_

*Academia de
talentos digitales*

