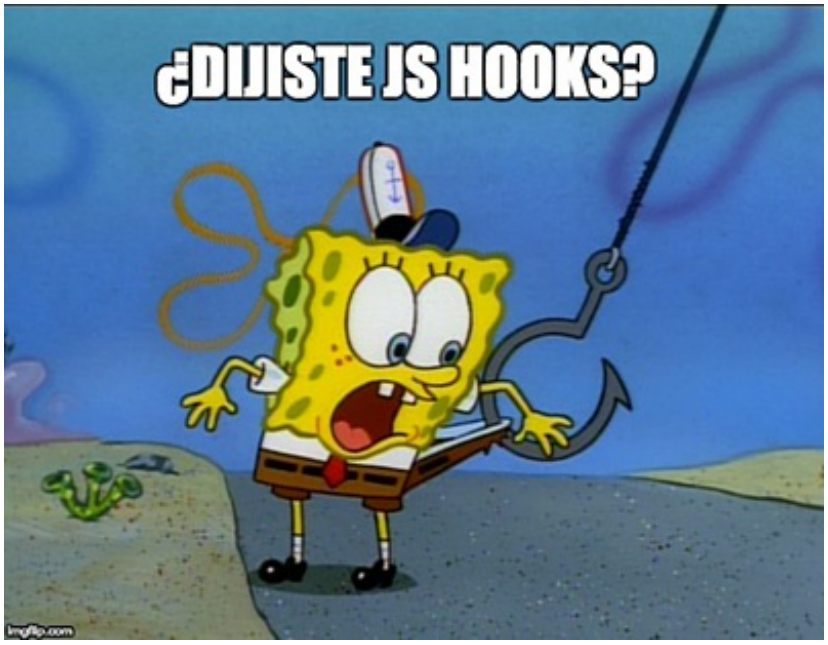
Hooks de Javascript

Es importante saber que a la hora de personalizar una plantilla de **Tienda Nube**, hay componentes de la maqueta que son críticos para que las funcionalidades funcionen correctamente por ese motivo identificamos estos componentes con algo llamado **Js Hooks**.



Realmente no es nada de otro mundo, simplemente agregamos el prefijo **js-** para asociar una clase con un componente que va a tener algún tipo de comportamiento con Javascript para separar "cómo se ve" un elemento de "cómo actúa" el mismo.

El **Js hook** es usado cuando necesitamos asociar algo funcional a un componente que se repite en la página. De lo contrario si este es único y sabemos que no va a repetirse, es recomendable usar el ID .

Para agregar un **Js hook** lo hacemos de la misma manera que cuando lo asociamos para estilarlo:

```

```

De esta forma, si no nos gusta alguna clase la borramos sin problemas

porque sabemos que no tiene ninguna funcionalidad asociada y podemos evitar casos como el siguiente:

Quizás decidamos borrar la clase **"a-really-confusing-class"** porque no encontramos ningún CSS asociado a la misma (de hecho nos molesta ese **span)**, pero **"a-really-confusing-class"** era usada para hacer un cálculo del total en el carrito de compras y al eliminarla causaría problemas.

Es por esto que los **JS hooks** nos ayudan a estar atentos y poder diferenciar qué componentes usan **Javascript** y cuáles no lo hacen.

Por otro lado, no vamos a estilar los elementos a través de las clases J**s hooks** ya que están prohibidas en el CSS. En ese caso estaríamos atando el "cómo se ve" ese componente a "cómo actúa" y, si a futuro queremos llevarnos ese comportamiento a otros componentes, tener J**s hooks** con estilos nos puede traer problemas, por lo que es bueno evitar casos como el siguiente:

```
.js-hook{
    your CSS properties...
}
```

Como mencionamos anteriormente, otra alternativa para asociar componentes a **Javascript** es el viejo y querido **ID**.

Recuerden intentar siempre mantener el comportamiento por un lado y los estilos por el otro. Eso los va a ayudar a moverse con más confianza a la hora de modificar el código, no solo a ustedes sino a colegas que con los que pueden estar trabajando =)