

HINTS

AJAX

Ajax es una tecnología que permite hacer consultas por HTTP de manera asíncrona contra el servidor. Esta tecnología permite acceder a datos existentes en el servidor sin necesidad de recargar la página completamente.

Para conocer más de AJAX podemos ingresar a [Introducción a AJAX](#).

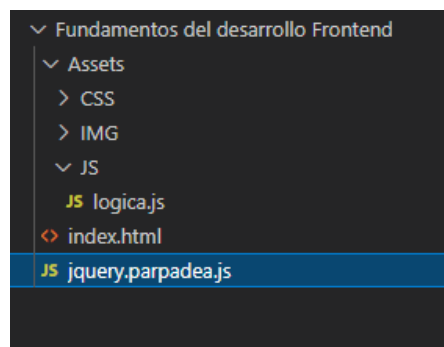
API

API o *Application Programming Interfaces* o Interfaz de programación de aplicaciones, es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

CREAR PLUGIN

Además de agregar plugin a nuestros proyectos, podemos crear nuestros plugin personalizados. Para eso debemos tener en cuenta las siguientes consideraciones:

- Debemos crear un archivo en el cual se escribirá todo el código del plugin y que llevará por nombre `jquery.{nombre del plugin}.js`, por ejemplo, si creamos un plugin de nombre “parpadea” el archivo llevará por nombre `jquery.parpadea.js`.



- Las funciones se agregarán por medio de la propiedad `fn` del objeto `jQuery`.

```
1 jQuery.fn.parpadea = function ()
```

- Se usará la palabra reservada `"this"` para hacer referencia al objeto `jQuery` que revive el método.
- Colocar punto y coma al final del método creado como plugin para que el código fuente se comprima y siga funcionando correctamente.
- El método plugin debe retornar el objeto `jQuery` con la indicación `return this;`
- Usaremos `this.each` para iterar sobre cada uno de los elementos seleccionados.
- Para agregar el plugin al proyecto lo incluiremos de la misma forma que un plugin externo en el archivo principal.

```
1 <script type="text/javascript" src="jquery.parpadea.js"></script>
```

- Para hacer uso de este plugin lo haremos en el archivo `JS`.

```
1 $(document).ready(function() {  
2     $("p").parpadea();  
3 });
```

- El plugin "parpadea" tiene el siguiente código:

```
1 jQuery.fn.parpadea = function () {  
2     this.each(function () {  
3         elem = $(this);  
4         elem.fadeOut(250, function () {  
5             $(this).fadeIn(250);  
6         });  
7     });  
8     return this;};
```

Además, es necesario saber que podemos realizar, para este ejemplo simple, modificaciones en la función que se lleva a cabo a gusto y necesidad nuestra.