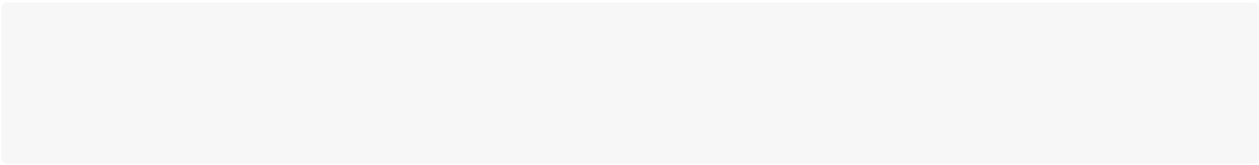


jQuery est un framework JavaScript, il est donc essentiel de comprendre un minimum ce langage pour progresser plus facilement.

Vous pouvez afficher votre 1ère variable dans la console de votre navigateur.

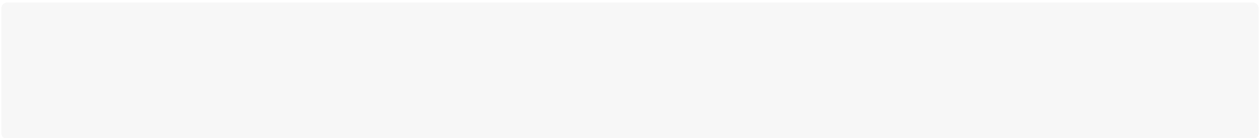
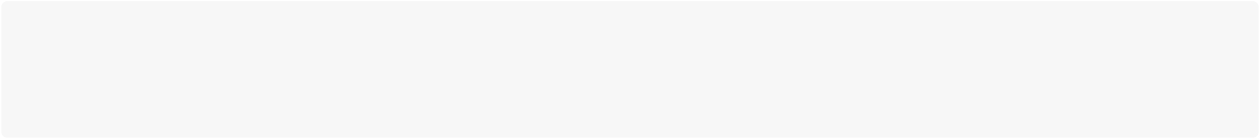
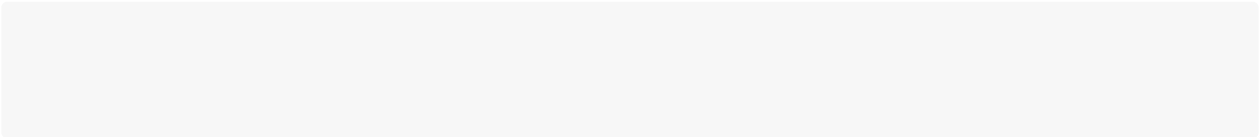
```
var maVariable = "J'aime JavaScript";  
console.log(maVariable);
```


L opérateur arithmétique +

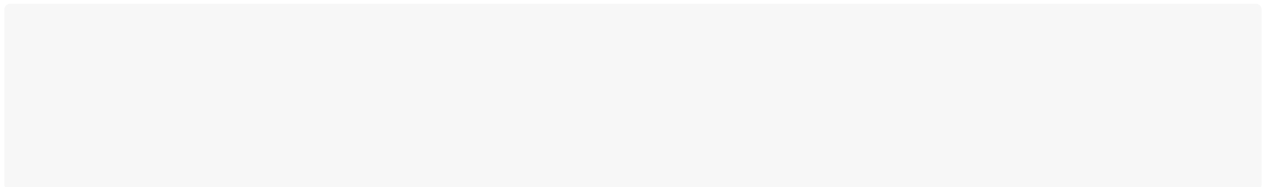


Les boucles sont des structures qui permettent d'exécuter plusieurs fois la même série d'instructions jusqu'à ce qu'une condition ne soit plus réalisée.

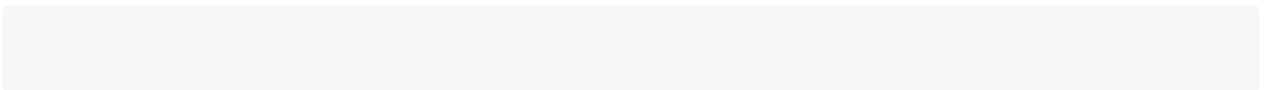
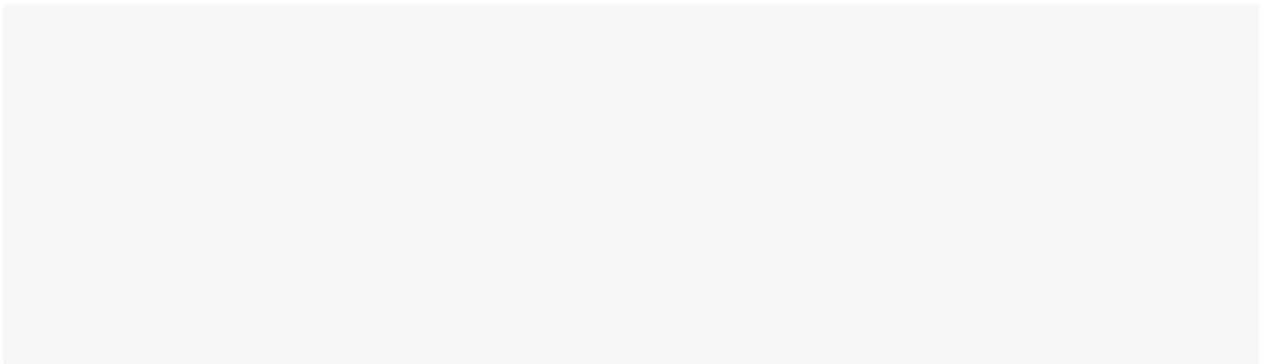
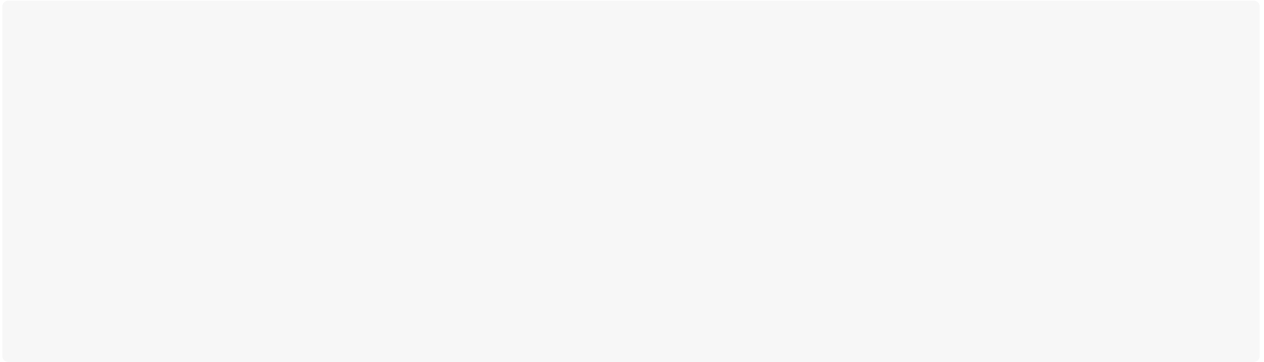
La façon la plus commune de faire une boucle est de créer un compteur (une variable qui s'incrémente, c'est-à-dire qui augmente de 1 à chaque tour de boucle) et de faire arrêter la boucle lorsque le compteur dépasse une



```
while (condition réalisée) {
```



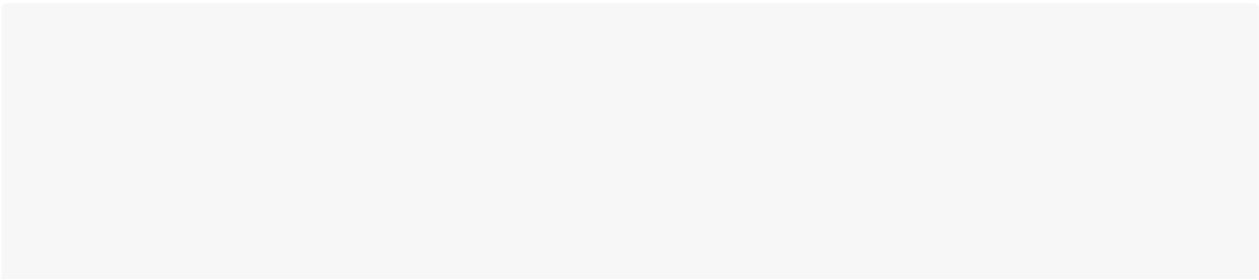
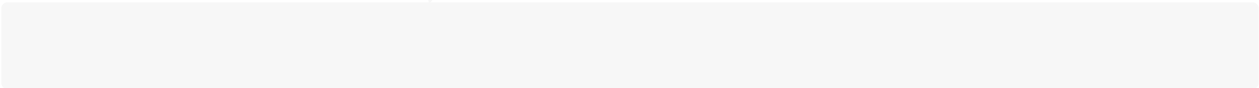
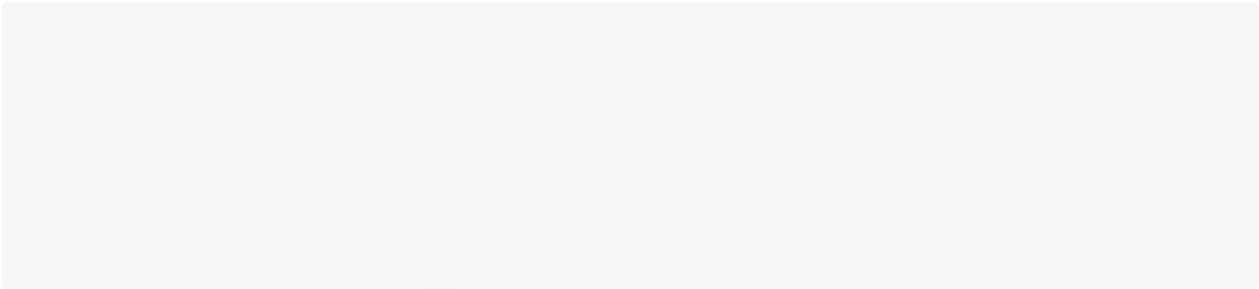
1. un nommé "jquery-2.1.1.js"
2. un autre nommé "jquery-2.1.1.min.js"



1. Pour le développement de votre site, utilisez jQuery en local pour la rapidité d'exécution.
2. Pour la production, passez par un CDN afin d'utiliser le cache de vos utilisateurs.

Les fonctions de rappel ou **callback**

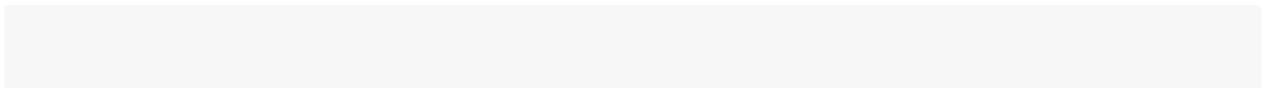
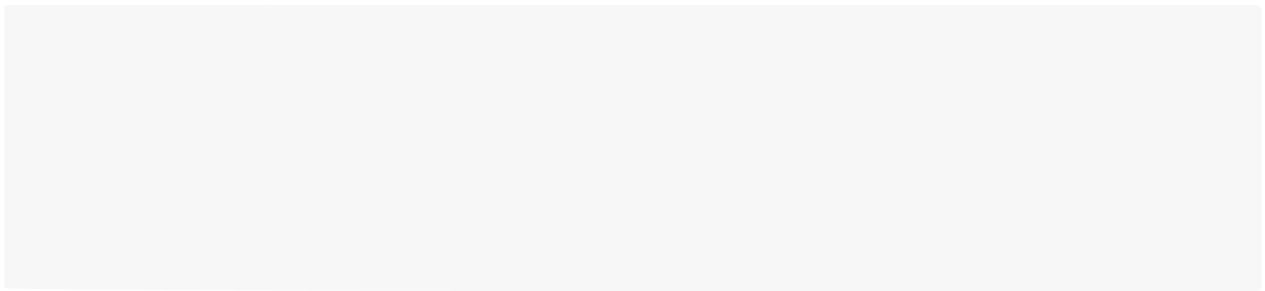
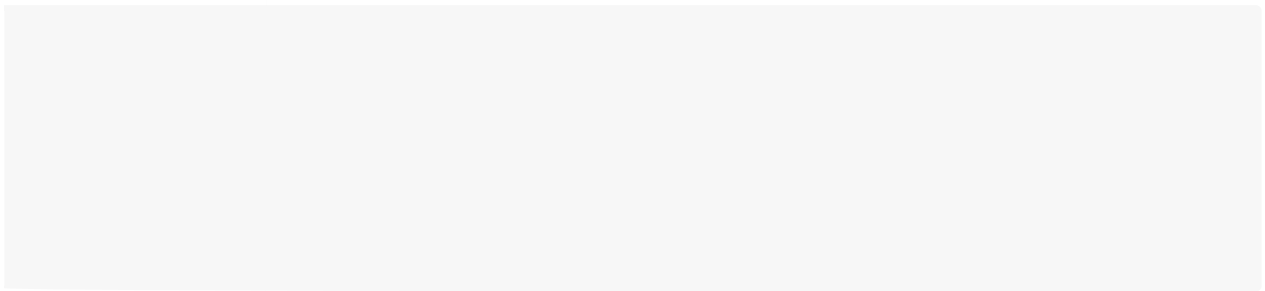
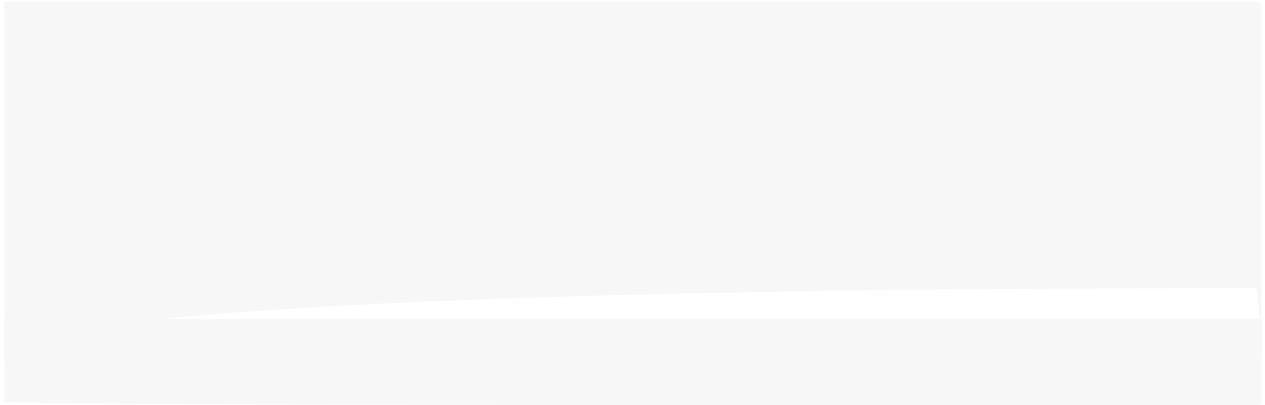
Dans cette partie, np ttl cm i

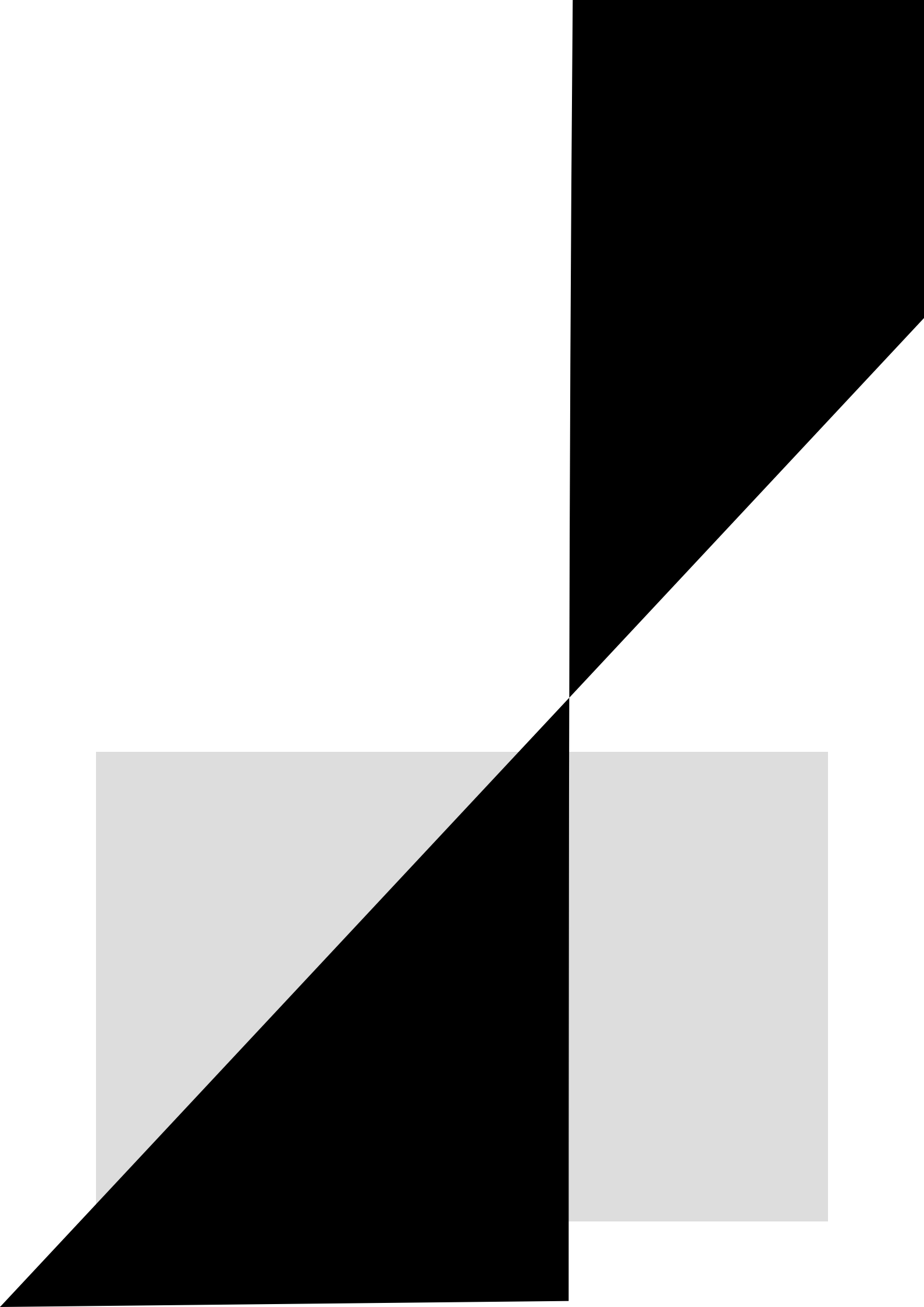


<http://api.jquery.com/prevAll/>

`.prevAll()` retourne tous les frères prédédents de l'élément sélectionné.

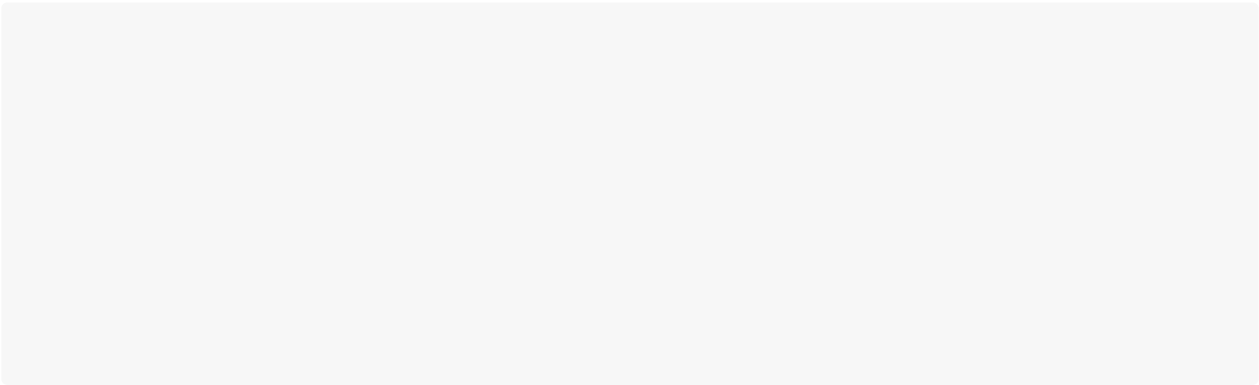
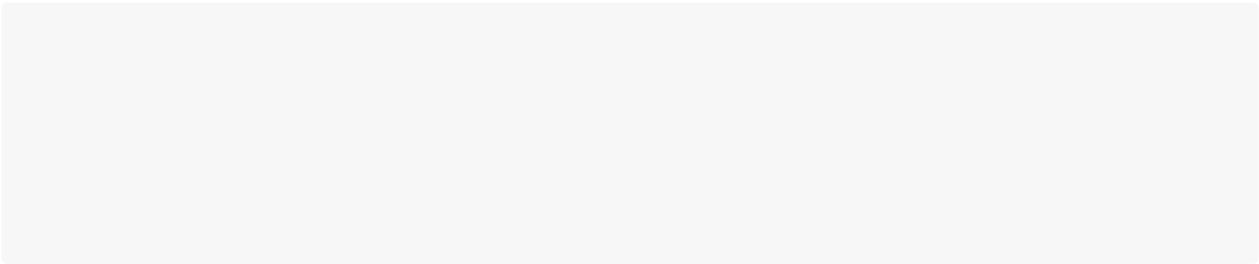
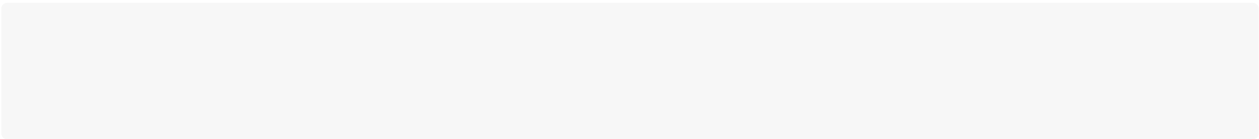
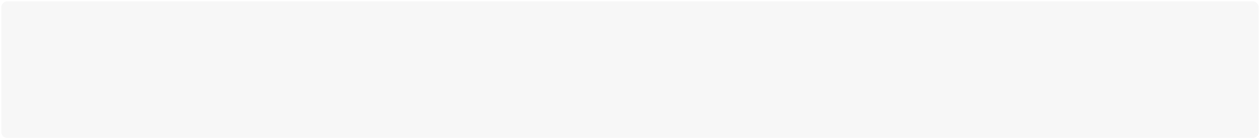
L'exemple suivant ajoute `class="selected"` aux frères prédédents de





Dans la précédente partie, nous avons vu comment effectuer des tâches lors d'un événement sur le navigateur.

API : <http://api.jquery.com/category/events/>



--	--

API :


```
});
```

Un autre exemple :

See the Pen [rfkcl](#) by Sutterlity ([@sutterlity](#)) on [CodePen](#).

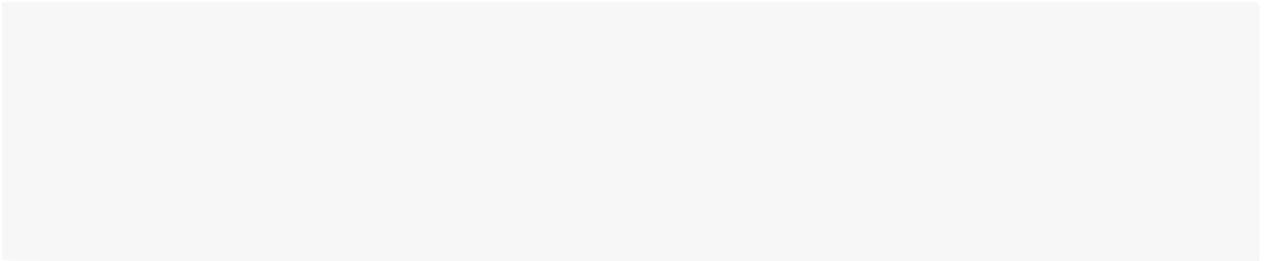
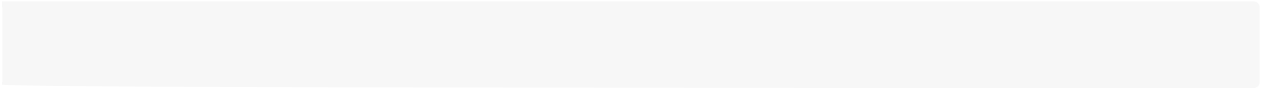
API : <http://api.jquery.com/toggle/>

L 3

Dans cette partie nous verrons les méthodes

<http://api.jquery.com/slideToggle/>

```
.slideToggle()
```



API :

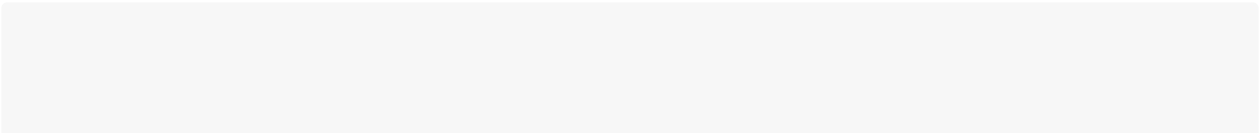
API : <http://api.jquery.com/jQuery.post/>

La méthode `.post()` permet d'envoyer des données. Par exemple, vous pouvez l'utiliser pour envoyer des données saisies dans un formulaire.

La méthode consiste à envoyer des données vers un fichier **Php** qui se chargera de les transmettre au serveur. Cette méthode peut également retourner des informations en **callback** dans la page.

```
$.post( 'send.php',
  {
    name: 'Jean-Michel',
    email: 'jeanmich@caramail.com'
  }, function(data) {
    alert(data);
  });
```

```
$.get( url, data, function(data,status,xhr), dataType );
```


```
success: function (data) {  
    /-c      an
```

Dans cette partie, on va voir comment on peut avoir une

représentation d'un objet mathématique sous forme d'un tableau.

On va voir comment on peut avoir une représentation d'un objet

mathématique sous forme d'un tableau.

On va voir comment on peut avoir une représentation d'un objet

```
(function($) {
```



Comme la bibliothèque jQuery, jQuery UI n'est pas installé par défaut dans votre navigateur, il va

