



Support de cours jQuery

A large blue diagonal graphic that starts from the top right corner and extends towards the bottom left, creating a split background of white and blue.

1.

Introduction

Introduction

Présentation

- JQuery est une bibliothèque JavaScript.
- JQuery simplifie grandement la programmation JavaScript.
- JQuery est très utilisé

Introduction

Présentation

JQuery est une bibliothèque JavaScript, son slogan : "écrire moins, faire plus".

Le but de jQuery est de rendre beaucoup plus facile l'utilisation de JavaScript.

JQuery permet de faciliter beaucoup de tâches courantes qui nécessitent de nombreuses lignes de code JavaScript à accomplir, et les enveloppent dans des méthodes que vous pouvez appeler avec une seule ligne de code.

JQuery simplifie aussi beaucoup de choses compliquées à partir de JavaScript, comme les appels AJAX et la manipulation DOM.

Introduction

Possibilités

La bibliothèque jQuery contient les fonctionnalités suivantes:

- Manipulation HTML / DOM
- Manipulation CSS
- Méthodes d'événements HTML
- Effets et animations
- AJAX
- Utilitaires

Astuce: En outre, jQuery a des plugins pour presque n'importe quelle tâche.

Introduction

Utilisation reconnue

Il ya beaucoup d'autres frameworks JavaScript, mais jQuery semble être le plus populaire, et aussi le plus extensible.

Beaucoup des plus grandes entreprises sur le Web utilisent jQuery, tels que:

- Google
- Microsoft
- IBM
- Netflix

Introduction

Intégration locale

Il existe deux versions de jQuery disponibles pour le téléchargement:

- **Version de production** : ceci est pour votre site Web en ligne parce qu'il a été minifié.
- **Version de développement** : ceci est pour le test et le développement (code non compressé et lisible).

Les deux versions peuvent être téléchargées à partir de [jQuery.com](https://jquery.com).

La bibliothèque jQuery est un fichier JavaScript unique et vous pouvez la référencer avec la balise HTML `<script>` (notez que la balise `<script>` doit être à l'intérieur de la section `<head>`):

```
<head>  
  <script src="jquery-3.1.1.min.js"></script>  
</head>
```

Introduction

Intégration CDN

Si vous ne souhaitez pas télécharger et héberger jQuery vous-même, vous pouvez l'inclure à partir d'un CDN (Content Delivery Network).

Google et Jquery.com hébergent jQuery.

Pour utiliser jQuery depuis Google ou Jquery.com, utilisez l'une des méthodes suivantes:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
```


Introduction

Intégration CDN - Avantages

Beaucoup d'utilisateurs ont déjà téléchargé jQuery de Google ou Microsoft lors de la visite d'un autre site.

- En conséquence, il sera chargé à partir de cache quand ils visitent votre site, ce qui conduit à un temps de chargement plus rapide.
- En outre, la plupart des CDN veilleront à ce qu'une fois qu'un utilisateur lui demande un fichier, il sera servi par le serveur le plus proche, ce qui conduit également à un temps de chargement plus rapide.

Introduction

Syntaxe générale

La syntaxe jQuery est faite sur mesure pour sélectionner des éléments HTML et effectuer une action sur les éléments.

La syntaxe de base est: **\$ (sélecteur). Action ();**

\$: pour définir / accéder à jQuery
(sélecteur)	: pour "interroger (ou trouver)" les éléments HTML
Action ()	: une action jQuery () à effectuer sur le sélecteur

Introduction

Exemple

Actions JQuery :

show() : affiche un élément
hide() : cache un élément

Exemple : Cacher et afficher un élément

```
// Cacher  
$("#test").hide() ;  
// Afficher  
$("#test").show() ;
```

Introduction

Syntaxe générale

Il est aussi possible d'introduire un contexte pour le sélecteur que nous sélectionnons pour limiter sa portée. Ce contexte peut-être un sélecteur CSS ou un objet provenant du DOM

La syntaxe de base est: **\$ (sélecteur , contexte). Action ();**

Exemple : Cacher et afficher un élément

```
// Cacher  
var maSection = $('section:nth-of-type(1)');  
$("article", maSection ).hide() ;
```

Introduction - Les sélecteurs

Prérequis

Lorsque nous faisons des manipulations de DOM, il faut s'assurer que ces éléments soient chargés. Nous prendrons donc l'habitude d'entourer nos appels JQuery par :

```
$(document).ready(function(){  
  
    // méthodes jquery  
  
});
```

Introduction - Les sélecteurs

Cibler une balise

La première façon de sélectionner une balise est de cibler le nom d'une balise :

Exemple :

```
header  
main  
p  
a
```

Introduction - Les sélecteurs

Notion de classe

Exemple :

Côté HTML :

```
<p class="error"> Ceci est une erreur </p>  
<p class="warning"> Ceci est un avertissement</p>  
<p class="error"> Ceci est une autre erreur </p>  
<p> Ceci est un paragraphe sans classe </p>
```

Sélecteur :

```
.error  
.warning
```

Introduction - Les sélecteurs

Notion d' id

Côté HTML :

```
<section id="presentation"> ... </section>  
<section id="commentaires"> ... </section>  
<section> ... </section>
```

Sélecteur :

```
#presentation  
#commentaires
```


Introduction - Les sélecteurs

Balise + non-sémantique

Les sélecteurs de balise et de classe/id sont cumulables. Je peux cibler une certaine balise ayant une certaine classe ou un certain id pour gagner en précision :

Exemple :

section#presentation
section.commentaire
div.error
span.info

Prenons le dernier exemple : “span.warning”. Si j’applique une classe “warning” sur une balise autre (<div class=“info”>Texte</div>), le sélecteur ne s’appliquera pas.

Introduction - Les sélecteurs

Les pseudo-classes

Les pseudos-classes sont notamment utilisées avec les liens :

Pseudo classe	Description
:link	Cible un élément qui est la source d'un lien non encore visité.
:hover	Utilisé pour les imprimantes
:focus	Lorsque le visiteur passe la souris sur le lien
:active	Lorsque le lien a un focus actif
:visited	Cible un élément qui est la source d'un lien ayant été visité.

Introduction - Les sélecteurs

Les pseudo-classes

D'autres pseudos classes :

Pseudo classe	Description
p:first-child	Sélectionne tous les éléments <p> qui sont le premier enfant de son parent
p:first-of-type	Sélectionne chaque <p> élément qui est le premier <p> élément de son parent
p:last-child	Sélectionne tous les éléments <p> qui sont le dernier enfant de son parent
p:last-of-type	Sélectionne chaque <p> élément qui est le dernier <p> élément de son parent

Introduction - Les sélecteurs

Les pseudo-classes

D'autres pseudos classes :

Pseudo classe	Description
:not(p)	Sélectionne chaque élément qui n'est pas un élément <p>
p:nth-child(2)	Sélectionne chaque <p> élément qui le second élément de son parent
p:nth-of-type(2)	Sélectionne chaque <p> élément qui est le second <p> élément de son parent

Introduction - Les sélecteurs

Les pseudo-classes

D'autres pseudos classes :

Pseudo classe	Description
input:checked	Sélectionne chaque élément <entrée> coché
input:disabled	Sélectionne chaque élément <input> désactivé
input:enabled	Sélectionne chaque élément <input> activé
input:focus	Sélectionne l'élément <input> qui a le focus
input:invalid	Sélectionne tous les éléments <input> avec une valeur non valide

Introduction - Les sélecteurs

Les pseudo-classes

D'autres pseudos classes :

Pseudo classe	Description
input:optional	Sélectionne les éléments <input> sans attributs "required"
input:read-only	Sélectionne les éléments <input> avec un attribut "readonly" spécifié
input:read-write	Sélectionne les éléments <input> sans attribut "readonly"
input:required	Sélectionne les éléments <input> avec un attribut "required" spécifié
input:valid	Sélectionne tous les éléments <input> avec une valeur valide

Introduction - Les sélecteurs

Les pseudo-éléments

D'autres pseudos classes :

Pseudo élément	Description
p::first-letter	Sélectionne la première lettre de chaque élément <p>
p::first-line	Sélectionne la première ligne de chaque élément <p>
p::selection	Sélectionne la partie d'un élément qui est sélectionnée par un utilisateur

Introduction - Les sélecteurs

Sélection multiple

CSS offre des sélecteurs permettant de sélectionner divers balises à travers le DOM (hiérarchie, sélection multiple, ...).

Sélecteur	Description
*	Permet de sélectionner tous les éléments
E, F	Sélection des éléments E et F
E F	Sélection des éléments F présent dans les éléments E
E > F	Sélection des éléments F directement enfant des éléments E
E ~ F	Sélection des éléments F étant des frères de E (même niveau dans le DOM)

2.

Le DOM avec jQuery

Le DOM avec jQuery

Récupérer le contenu

Une partie très importante de jQuery est la possibilité de manipuler le DOM. JQuery est livré avec un tas de méthodes DOM connexes qui facilitent l'accès et la manipulation des éléments et des attributs.

Trois méthodes simples, mais utiles, de jQuery pour la manipulation de DOM sont:

- **text()** : Définit ou retourne le contenu textuel des éléments sélectionnés
- **html()** : Définit ou retourne le contenu des éléments sélectionnés (y compris le balisage HTML)
- **val()** : Définit ou renvoie la valeur des champs de formulaire

Le DOM avec jQuery

Récupérer le contenu

Exemple :

HTML :

```
<p id="paraTest">Ce texte est <b>important</b> malgré tout.</p>
```

JavaScript :

```
alert("Text: " + $("#paraTest").text() +  
      "\nHTML: " + $("#paraTest").html());
```

// ou avec l'usage d'une variable intermédiaire

```
var paraTest = $("#paraTest");  
alert("Text: " + paraTest.text() + "\nHTML: " + paraTest.html());
```

Le DOM avec jQuery

Récupérer le contenu

Dans le cas d'un champ de formulaire :

Exemple :

HTML :

```
<p>Label : <input type="text" value="Valeur du champ de texte"></p>
```

JavaScript :

```
alert("Valeur: " + $('input[type="text"]').val());
```

Le DOM avec jQuery

Récupérer les attributs

La méthode jQuery attr () est utilisée pour obtenir des valeurs d'attribut :

Exemple :

HTML :

```

```

JavaScript :

```
alert($(".img").attr("src"));
```

Le DOM avec jQuery

Manipulation du CSS

Affichage à partir du DOM

- » Créer une balise « p » possédant un « id »
- » Attribuer du contenu textuel à ce paragraphe
- » A l'intérieur du paragraphe, créer une balise « strong » sur une partie du paragraphe

Afficher dans une autre balise :

- » Le contenu html du paragraphe
- » Le contenu textuel du paragraphe

Le DOM avec jQuery

Modifier le contenu

La méthodes `text()`, `html()`, and `val()` peuvent aussi permettre de définir de nouvelles valeurs de contenu à un élément. Pour cela, préciser le contenu voulu dans le premier argument des trois méthodes.

Exemple :

```
$("#test1").text("Hello world!");  
$("#test2").html("<b>Hello world!</b>");  
$('input[type="text"]').val("Dolly Duck");
```

Le DOM avec jQuery

Ajouter des éléments

Pour ajouter des éléments jQuery propose entre autre quatre méthodes qui sont utilisées pour ajouter du nouveau contenu :

- `Append ()` : Insère du contenu à la fin des éléments sélectionnés
- `Prepend ()` : Insère du contenu au début des éléments sélectionnés
- `After ()` : Insère du contenu après les éléments sélectionnés
- `Before ()` : Insère du contenu avant les éléments sélectionnés

Le DOM avec jQuery

Ajouter des éléments

- La méthode jQuery `append()` insère du contenu A LA FIN des éléments HTML sélectionnés.

```
$("#p").append("Some appended text.");
```

- La méthode jQuery `prepend()` insère du contenu AU DÉBUT des éléments HTML sélectionnés.

```
$("#p").prepend("Some prepended text.");
```

Le DOM avec jQuery

Ajouter des éléments

Les méthodes jQuery `append()` et `prepend()` accepte en arguments (illimités) pour chaîner les différents contenus à ajouter :

```
var txt1 = "<p>Text.</p>";           // Création HTML
var txt2 = $("<p></p>").text("Text."); // Création jQuery
var txt3 = document.createElement("p"); // Création JavaScript
txt3.innerHTML = "Text.";

$("body").append(txt1, txt2, txt3); // Ajouter les nouveaux éléments
```

Le DOM avec jQuery

Ajouter des éléments

- La méthode jQuery `after()` insère du contenu APRÈS les éléments HTML sélectionnés.

```
$("img").after("Some text after");
```

- La méthode jQuery `before()` insère du contenu AVANT les éléments HTML sélectionnés.

```
$("img").before("Some text before");
```

Le DOM avec jQuery

Ajouter des éléments

Les méthodes jQuery `after()` et `before()` accepte aussi des arguments pour ajouter plusieurs contenus :

```
var txt1 = "<p>Text.</p>";           // Création HTML
var txt2 = $("<p></p>").text("Text."); // Création jQuery
var txt3 = document.createElement("p"); // Création JavaScript
txt3.innerHTML = "Text.";

$("img").after(txt1, txt2, txt3);    // Les éléments sont insérés après l'image
```

Le DOM avec jQuery

Supprimer des éléments

Pour supprimer des éléments et du contenu, il existe principalement deux méthodes jQuery:

→ **Remove ()** : Supprime l'élément sélectionné (et ses éléments enfants)

```
$("#div1").remove();
```

→ **Empty ()** : Supprime les éléments enfants de l'élément sélectionné

```
$("#div1").empty();
```

Le DOM avec jQuery

Supprimer des éléments

La méthode jQuery `remove()` accepte également un paramètre qui permet de filtrer les éléments à supprimer.

Le paramètre peut être l'une des syntaxes du sélecteur jQuery.

Exemple :

```
$("#p").remove(".test");  
$("#p").remove(".test, .demo");
```

Le DOM avec jQuery

Manipulation du CSS

jQuery dispose de plusieurs méthodes de manipulation CSS. Nous examinerons les méthodes suivantes:

- **AddClass ()** : Ajoute une ou plusieurs classes aux éléments sélectionnés
- **RemoveClass ()** : Supprime une ou plusieurs classes des éléments sélectionnés
- **ToggleClass ()** : Ajoute ou supprime (en fonction de l'existence) des classes des éléments sélectionnés
- **Css ()** : Définit ou renvoie l'attribut de style

Le DOM avec jQuery

Manipulation du CSS

→ AddClass ()

```
$("#h1, h2, p").addClass("blue");  
$("#div1").addClass("important blue");
```

→ RemoveClass ()

```
$("#h1, h2, p").removeClass("blue");
```

→ ToggleClass ()

```
$("#h1, h2, p").toggleClass("blue");
```


Le DOM avec jQuery

Manipulation du CSS

Pour retourner la valeur d'une propriété CSS spécifiée, utilisez la syntaxe suivante pour la méthode `css()` :

Syntaxe :

```
css(" propertyname ");
```

Exemple :

```
$("p").css("background-color");
```

Le DOM avec jQuery

Manipulation du CSS

Pour modifier la valeur d'une propriété CSS spécifiée, utilisez la syntaxe suivante pour la méthode `css()` :

Syntaxe :

```
css(" propertyname ", " value", );
```

Exemple :

```
$("p").css("background-color", "chocolate");
```

Le DOM avec jQuery

Manipulation du CSS

Pour définir plusieurs propriétés CSS, utilisez la syntaxe suivante avec la méthode `css()` :

Syntaxe :

```
css({"propertyname ":" value ","propertyname ":" value ", ... });
```

Exemple :

```
$("p").css({"background-color": "navy", "font-size": "200%"});
```

Le DOM avec jQuery

Manipulation du CSS

Style

1. Créer une div
Définir son modèle de boîte en CSS pour pouvoir le visualiser (dimensions, bordures, marges)
2. Changer le style de cet élément grâce à jQuery.
 - » Dimensions
 - » Couleur de fond
 - » Bordures

A large blue diagonal graphic element that starts from the top right corner and extends towards the bottom left, creating a split background of white and blue.

3.

Les événements

Les événements

Présentation

Toutes les actions du visiteur différentes auxquelles une page Web peut répondre sont appelées événements.

Un événement représente le moment précis où quelque chose se passe.

Exemples:

- Déplacer une souris sur un élément
- Sélection d'un bouton radio
- Cliquer sur un élément

Les événements

Les événements souris

événement	Description
click	Clic simple
dblclick	Double clic
mousedown	Le bouton gauche de la souris est enfoncé
mouseenter	Le pointeur de la souris entre au dessus de l'élément sélectionné
mouseleave	Le pointeur de la souris quitte l'élément sélectionné.
mousemove	Le pointeur de la souris se déplace dans l'élément sélectionné.
mouseup	Le bouton gauche de la souris est libéré sur l'élément sélectionné.

Les événements

Les événements clavier

événement	Description
keydown	La touche est sur son chemin vers le bas
keypress	La touche est enfoncée
keyup	La touche est libérée

Les événements

D'autres événements

événement	Description
focus	Lorsque l'élément est sélectionné par un clic de souris ou par "navigation par onglets" (avec la touche "tab")
blur	Lorsqu'un élément perd le focus
resize	Lorsque l'élément est redimensionné
scroll	L'utilisateur fait défiler l'élément spécifié.
submit	Lorsqu'un formulaire est soumis.

Les événements

D'autres événements

Dans jQuery, la plupart des événements DOM ont une méthode jQuery équivalente.

Exemple :

```
$("p").click(function(){  
    // action effectuées lors de l'événement  
});
```

Les événements

Exemples

Exemples :

```
$("#p1").mouseenter(function(){ alert("You entered p1!"); });
$("#p1").mouseleave(function(){ alert("Bye! You now leave p1!"); });
$("#p1").mousedown(function(){ alert("Mouse down over p1!"); });
$("#p1").mouseup(function(){ alert("Mouse up over p1!"); });
$("#p1").hover(function(){ alert("You entered p1!"); },
$("#input").focus(function(){ $(this).css("background-color", "#cccccc"); });
$("#input").blur(function(){ $(this).css("background-color", "#ffffff"); });
```

Les événements

Ancienne méthodes de gestion d'événements

Jusqu'à jQuery 1.7 on faisait la distinction entre la méthode “bind()”, “live()” et “delegate()” afin de créer des écouteurs d'événements.

Depuis jQuery 1.7, toutes ces méthodes ont été remplacées par l'unique méthode “on()”.

Les principes de performance sont les mêmes, seule la syntaxe change.

Les événements

Exemples

La méthode “on()” attache un ou plusieurs gestionnaires d'événements pour les éléments sélectionnés.

Exemple :

```
$("#p").on("click", function(){  
    $(this).hide();  
});
```

Les événements

Plusieurs événements dans un gestionnaire

Il est possible d'attacher plusieurs événements à un seul sélecteur :

```
$("#p").on({  
  mouseenter: function(){  
    $(this).css("background-color", "lightgray");  
  },  
  mouseleave: function(){  
    $(this).css("background-color", "lightblue");  
  },  
  click: function(){  
    $(this).css("background-color", "yellow");  
  }  
});
```

Les événements

Plusieurs événements avec une même action

Il est aussi possible d'attacher plusieurs événements avec la même action :

```
$( "#cart" ).on( "mouseenter mouseleave", function( event ) {  
    // Action  
});
```

Les événements

Utiliser \$(this)

Pour récupérer l'élément sélectionné avec accéder à ses propriétés et/ou méthodes, utiliser "this" ;

```
$( "p" ).on( "click", function() {  
    alert( $( this ).text() );  
});
```


Les événements

Empêcher la soumission d'un formulaire

Lorsque vous faites de la validation de formulaires côté client ou que vous voulez empêcher l'activation d'un lien, il existe 2 méthodes :

```
$( "form" ).on( "submit", false );
```

Ou

```
$( "form" ).on( "submit", function( event ) {  
    event.preventDefault();  
});
```

Les événements

Détacher un événement

Il est possible de détacher un événement attaché avec “on” grâce à “off”

Exemple :

```
$( "p" ).on( "click", function() {  
    alert( $( this ).text() );  
});
```

```
$( "p" ).off();
```

Les événements

Détacher un événement

Si un événement est lié à plusieurs fonctions il est possible d'en détacher une sans détacher les autres.

Exemple :

HTML :

```
<button id="reactOnClick">Cliquer</button>  
<button id="reactOffClick">Supprimer la réaction au clic</button>
```

JavaScript:

```
function log1() { console.log('log1'); }  
function log2() { console.log('log2'); }  
  
function clearLog1() {console.log('clear log1'); $("#reactOnClick").off('click', log1); }  
  
$("#reactOnClick").on('click', log1);  
$("#reactOnClick").on('click', log2);  
$("#reactOffClick").on('click', clearLog1);
```

Les événements

Exercice

Faire un formulaire d'inscription, avec les champs suivants :

Nom*, prénom, email*, profession ,mot de passe*, confirmation mot de passe*

Checkbox avec la mention "Ces données seront exploitées seulement par notre entreprise et nos partenaires"*.

Checkbox avec la mention "Voulez-vous nous soutenir et recevoir des mails de nos partenaires"

Lors du clique sur la deuxième checkbox nous afficherons en dessous de la checkbox une div avec un fond vert. La div portera la mention "Merci beaucoup pour votre soutiens. Si l'utilisateur décoche la case, la div devra disparaître.

Les événements

Exercice

Faire un bouton dark theme qui mets un fond noir à tout le document et l'écriture en blanc.

Faire un bouton light theme qui mets un fond blanc à tout le document et l'écriture en noir.

Bonus

Les événements se déclencheront lors du clique sur une div avec les bordures arrondies de 10px. La div dark theme aura un fond noir et le texte en blanc. Le light theme aura le fond blanc avec une écriture en noir.

Lors du survol sur les div les bordures de la div deviendront rouge.

4.

**Le DOM : une grande
famille**

Le DOM : une grande famille

Les parents

Il est possible de récupérer les ancêtres directes ou indirectes depuis un élément jQuery.

La méthode `parent()` renvoie l'élément parent direct de l'élément sélectionné.

Cette méthode ne traverse qu'un seul niveau vers le haut de l'arbre DOM.

Exemple :

```
console.log($(".li:nth-child(1)").parent());
```

Le DOM : une grande famille

Les parents

La méthode `parents()` renvoie tous les éléments ancêtres de l'élément sélectionné, jusqu'à l'élément racine du document (`<html>`).

Exemple :

```
console.log($(".li:nth-child(1)").parents());
```

Vous pouvez également utiliser un paramètre optionnel pour filtrer la recherche des ancêtres.

Exemple :

```
console.log($(".li:nth-child(1)").parents('section'));
```


Le DOM : une grande famille

Les parents

La méthode `parentsUntil ()` renvoie tous les éléments ancêtres jusqu'à un certain élément :

Exemple :

```
console.log($(".li:nth-child(1)").parentsUntil('body'));
```

Remarque : l'élément précisé en paramètre (ici "body" n'est pas compris dans la sélection.

Le DOM : une grande famille

Les enfants

La méthode `children()` renvoie tous les enfants directs de l'élément sélectionné.

Cette méthode ne traverse qu'un seul niveau dans l'arbre DOM.

Exemple :

```
console.log($(".ul").children());
```

Vous pouvez également utiliser un paramètre optionnel pour filtrer la recherche des enfants.

Exemple :

```
console.log($(".ul").children('li.classStylised'));
```

Le DOM : une grande famille

Les enfants

La méthode `find()` renvoie les éléments descendants de l'élément sélectionné, jusqu'à la dernière descendance.

Exemple :

```
console.log($(".section").find("li"));
```

Le DOM : une grande famille

Les frères et soeurs

La méthode des `siblings()` renvoie tous les éléments de fratrie de l'élément sélectionné.

Exemple :

```
console.log($(".li.classStylised").siblings());
```

Vous pouvez également utiliser un paramètre optionnel pour filtrer la recherche de frères et sœurs.

Exemple :

```
console.log($(".section").siblings('section'));
```

Le DOM : une grande famille

Les frères et soeurs

La méthode `next()` renvoie l'élément de fratrie suivant l'élément sélectionné.

Exemple :

```
console.log($(".li.classStylised").next());
```

La méthode `prev()` renvoie l'élément de fratrie précédant l'élément sélectionné.

Exemple :

```
console.log($(".li.classStylised").prev());
```

Le DOM : une grande famille

Premier et dernier élément

La méthode `first()` renvoie le premier élément des éléments sélectionnés.

Exemple :

```
console.log($(".li").first());
```

La méthode `last()` renvoie le dernier élément des éléments sélectionnés.

Exemple :

```
console.log($(".li").last());
```

A large, solid green diagonal shape that starts from the top right and extends towards the bottom left, covering the right half of the slide.

5.

Les animations

Les animations

Effet simple

Avant de faire des animations, passons en revue les différents effets qu'il est possible de faire facilement avec jQuery.

Avec jQuery, vous pouvez masquer et afficher des éléments HTML avec les méthodes `hide ()` et `show ()`:

Exemple :

```
$("#hide").click(function(){  
    $("p").hide();  
});
```

```
$("#show").click(function(){  
    $("p").show();  
});
```


Les animations

Vitesse de l'effet

Le paramètre de vitesse facultatif spécifie la vitesse du masquage / affichage et peut prendre les valeurs suivantes: “slow”, “fast” ou en millisecondes.

Exemple :

```
$("#button").click(function(){  
    $("#p").hide(1000);  
});
```

Les animations

Effet “toggle”

Vous pouvez basculer entre les méthodes `hide()` et `show()` avec la méthode `toggle()`.

Les éléments affichés sont cachés et les éléments cachés sont affichés:

Exemple :

```
$("button").click(function(){  
    $("p").toggle();  
});
```

Les animations

Effet “fade”

Avec jQuery vous pouvez faire fondre un élément dans et hors de la visibilité.

jQuery a les méthodes de fondu suivantes:

- `fadeIn()` : apparition
- `fadeOut()` : disparition
- `fadeToggle()` : alternance apparition/disparition
- `fadeTo()` : changement d'opacité/transparence

Exemple :

```
$("#button").click(function(){  
    $("#div1").fadeIn();  
});
```

Remarque : Les méthodes “fade” acceptent aussi en second paramètre une vitesse (“slow”, “fast”, millisecondes)

Les animations

Effet “fade”

→ `fadeOut()` : disparition

```
$("#button").click(function(){  
    $("#div1").fadeOut();  
    $("#div2").fadeOut("slow");  
    $("#div3").fadeOut(3000);  
});
```

→ `fadeToggle()` : alternance apparition/disparition

```
$("#button").click(function(){  
    $("#div1").fadeToggle();  
    $("#div2").fadeToggle("slow");  
    $("#div3").fadeToggle(3000);  
});
```

Les animations

Effet “slide”

L'effet “slide” fonctionne de la même façon que “fade”, il existe 3 méthodes de “slide” :

- `slideDown()` : utilisée pour glisser vers le bas un élément.
- `slideUp()` : utilisée pour glisser vers le haut d'un élément.
- `slideToggle()` : bascule entre les méthodes `slideDown ()` et `slideUp ()`.

```
$("#btnTest").click(function(){  
    $("#panel").slideUp();  
    $("#panel2").slideDown("slow");  
    $("#panel3").slideToggle("fast");  
});
```

Les animations

Toggle

Vous pouvez même spécifier la valeur d'animation d'une propriété comme «show», «hide» ou «toggle»:

Exemple :

```
$(document).ready(function(){  
  $("button").click(function(){  
    $("div").animate({  
      height: 'toggle'  
    });  
  });  
});
```

Les animations

Exercice

Crée une div dans un container avec une marge de 30% à droite et à gauche de l'écran. La couleur de fond sera "DeepSkyBlue". Le texte sera centré, et la div aura un padding de 10px en haut et en bas.

Nom : Wolff
Prénom : Toto

+

Nom : Wolff
Prénom : Toto

-

Profession : Dirigeant de Mercedes Motorsport
Email : totowolf@mercedes.com

Trick !

Jusqu'à présent, nous avons écrit des instructions jQuery une à la fois (l'une après l'autre).

Cependant, il existe une technique appelée chaînage, qui nous permet d'exécuter plusieurs commandes jQuery, l'une après l'autre, sur le même élément.

Exemple :

```
$("#p1").css("color", "red").slideUp(2000).slideDown(2000);
```

```
// Autre syntaxe possible
$("#p1").css("color", "red")
  .slideUp(2000)
  .slideDown(2000);
```