



jQuery

SOMMAIRE



I. Introduction

1. Introduction à jQuery
2. Inclure jQuery
3. Découverte de la syntaxe de base de jQuery

II. Manipulation du DOM avec jQuery

1. Sélectionner les éléments en jQuery
2. Se déplacer dans le DOM
3. Manipuler les contenus HTML et textuels
4. Insérer du contenu dans un élément
5. Insérer du contenu avant ou après un élément
6. Insérer du contenu autour d'un élément
7. Copier des éléments HTML
8. Supprimer des éléments et des contenus HTML
9. Remplacer des éléments HTML

II. Manipulation du DOM avec jQuery

10. Récupérer la valeur d'un attribut ou définir un attribut
11. La méthode prop() et les propriétés du DOM
12. Récupérer ou mettre à jour la valeur courante d'éléments
13. Les méthodes permettant de manipuler spécifiquement les attributs class
14. La méthode css()
15. Les méthodes liées aux dimensions des éléments
16. La méthode offset()
17. La méthode each()

SOMMAIRE



III. Gestion d'évènement avec jQuery

1. La gestion des événements avec jQuery
2. Les méthodes liées à la souris
3. Les méthodes liées au clavier
4. Les méthodes spécifiques aux formulaires
5. Les méthodes on() et off()
6. Déclencher un évènement avec trigger et triggerHandler()
7. Les propriétés de l'objet event

IV. Création d'effets et d'animations avec jQuery

1. Créer des effets de fondu en jQuery
2. Créer des effets de slide en jQuery
3. Afficher ou cacher un contenu avec show(), hide() et toggle()
4. Créer des effets ou animations personnalisés avec jQuery

IV. Création d'effets et d'animations avec jQuery

5. Gérer la file d'attente en jQuery
6. Stopper une animation avec les méthodes jQuery stop() ou finish()
7. Ajouter un délai avant l'exécution d'animations avec delay()

V. AJAX avec jQuery

1. Création de requêtes Ajax avec jQuery



I. Introduction

1- Introduction à jQuery



jQuery est ce qu'on appelle une « librairie » ou une « bibliothèque » JavaScript.

Le rôle d'une librairie, en informatique, est de simplifier l'utilisation d'un certain langage de programmation en fournissant un ensemble de codes déjà prêts à l'emploi.

La librairie jQuery possède un ensemble de blocs de codes JavaScripts préconçus et qui vont être généralement enfermés dans des méthodes.

Il va donc nous suffire d'appeler ces méthodes pour exécuter le code qu'elles contiennent.

jQuery permet ainsi un gain de temps grâce à l'utilisation de méthodes déjà définies.

jQuery permet également d'assurer le bon fonctionnement sur les différents navigateurs (ce besoin est moins présent maintenant car la compatibilité entre navigateurs s'est amélioré)

2- Inclure jQuery



Pour inclure jQuery dans un projet, nous allons tout d'abord sur la page d'accueil de la documentation à l'adresse suivante : <https://jquery.com/>

Il y a plusieurs manières d'intégrer la librairie, ici nous allons utiliser la mise en place à l'aide d'un CDN. Un CDN (**Content Delivery Network**) héberge à votre place du code tiers. (nous utiliserons la version minified)

2- Inclure jQuery

- Inclure jQuery dans un fichier HTML
- Durée 5 minutes



2- Inclure jQuery

- Correction

Exercice



3- Découverte de la syntaxe de base de jQuery



Pour accéder aux méthodes de jQuery, nous allons pouvoir utiliser 2 objets qui sont `jQuery` et `$` qui sont des propriétés de l'objet `Window`

Ces 2 objets sont des alias, cela veut dire que nous pouvons utiliser l'un ou l'autre de manière interchangeable.

Généralement nous utiliserons `$` car il est plus rapide à écrire.

Pour utiliser une méthode de jQuery, nous allons d'abord devoir sélectionner un élément en utilisant n'importe quel sélecteur CSS de la manière suivante :

`$("p");` Permet de sélectionner les paragraphes

`$(".test");` Permet de sélectionner les éléments possédant la classe test

Exemple

3- Découverte de la syntaxe de base de jQuery



On pourra alors ensuite chaîner les méthodes qui devront s'exécuter sur l'élément sélectionné.

Pour l'exemple, nous allons utiliser la méthode `hide()` qui permet de cacher le ou les éléments sélectionnés.

Pour manipuler le DOM, il est important que celui-ci existe et donc que l'intégralité de la page HTML soit chargé.

C'est la raison pour laquelle, en Javascript, nous mettons les scripts à la fin de la balise `body`.

Exemple

3- Découverte de la syntaxe de base de jQuery



En jQuery, il y a une fonction de rappel avec laquelle nous commencerons systématiquement et qui permet justement de s'assurer que le code HTML est entièrement chargé avant d'exécuter le code jQuery. Il s'agit de la méthode `ready()`.

Exemple

3- Découverte de la syntaxe de base de jQuery

- Dans un fichier html ayant 2 paragraphes, mettre en place jQuery
- Utiliser une méthode pour s'assurer que le DOM est bien chargé avant de la manipuler
- Cacher les paragraphes à l'aide d'une méthode jQuery
- Durée 5 minutes



3- Découverte de la syntaxe de base de jQuery

- Correction

Exercice





II. Manipulation du DOM avec jQuery

1- Sélectionner les éléments en jQuery *write less, do more.*

La méthode has()

Cette méthode appliquée sur un sélecteur permet de préciser la sélection en sélectionnant uniquement les éléments ayant des éléments enfants correspondants aux éléments mis en arguments.

```
$("p").has('span').css('color', 'blue');
```

Dans cet exemple, nous sélectionnons uniquement les éléments p ayant comme élément enfant un élément span.

Exemple

1- Sélectionner les éléments en jQuery *write less, do more.*

La méthode filter()

Permet d'affiner la sélection en sélectionnant uniquement les éléments ayant le sélecteur passé en argument

```
$('span').filter('.green').css('color', 'green');
```

Dans cet exemple, nous sélectionnons uniquement les éléments span qui ont la classe green.

Exemple

1- Sélectionner les éléments en jQuery *write less, do more.*

La méthode not()

Permet d'affiner la sélection en sélectionnant uniquement les éléments n'ayant pas le sélecteur passé en argument

```
$('span').not('.green').css('color', 'red');
```

Dans cet exemple, nous sélectionnons uniquement les éléments span qui n'ont pas la classe green.

Exemple

1- Sélectionner les éléments en jQuery *write less, do more.*

La méthode first()

Permet d'affiner la sélection en ne sélectionnant que le premier élément

```
$('span').first().css('color', 'purple');
```

Dans cet exemple, nous sélectionnons uniquement le premier span

La méthode last()

Permet d'affiner la sélection en ne sélectionnant que le dernier élément

```
$('span').last().css('color', 'orange');
```

Dans cet exemple, nous sélectionnons uniquement le dernier span

Exemple

1- Sélectionner les éléments en jQuery *write less, do more.*

La méthode eq()

Permet d'affiner la sélection en ne sélectionnant que l'élément selon sa position

```
$('span').eq(0).css('color', 'grey');
```

Dans cet exemple, nous sélectionnons uniquement le span ayant la position 0, c'est-à-dire le premier span.

La valeur peut-être négative dans ce cas, on commence par la fin sachant que -1 est le dernier élément, -2 l'avant dernier ...

Exemple

1- Sélectionner les éléments en jQuery

- Dans un fichier html créer les éléments suivant
 - Créer 10 paragraphes
 - Sur le 2ème paragraphe et le 4ème ajouter à l'intérieur un span englobant une partie du texte
 - Sur le 5ème et le 7ème ajouter un span avec la classe test
 - Sélectionner les paragraphes ayant un span et donnez aux paragraphes concernés la couleur rouge
 - Sélectionner le 6ème paragraphe et donner lui la couleur jaune
 - Sélectionner le premier paragraphe et le dernier et donnez leur la couleur bleu
 - Sélectionner les span ayant la classe test et donnez leur la couleur orange
 - Sélectionner les span n'ayant pas la classe test et donnez leur la couleur grise
- Durée 15 minutes



1- Sélectionner les éléments en jQuery

- Correction

[Exercice](#)

2- Se déplacer dans le DOM

La méthode parent()

Permet d'accéder à l'élément parent (ancêtre direct) ou aux éléments parents du ou des éléments ciblés.

Un sélecteur css peut également être mis en argument.

Si l'élément parent ne correspond pas au sélecteur css mis en argument, rien n'est retourné.

La méthode parents()

Permet d'accéder à tous les ancêtres du ou des éléments ciblés.

Un sélecteur css peut également être mis en argument et agira comme un filtre (voir la méthode filter).

Exemple

2- Se déplacer dans le DOM

La méthode parentsUntil()

Permet d'accéder à tous les ancêtres du ou des éléments ciblés jusqu'à l'ancêtre correspondant aux sélecteurs mis en argument.

L'ancêtre correspondant ne sera pas inclus.

Un deuxième sélecteur css peut également être mis en 2^{ème} argument et agira comme un filtre.

La méthode closest()

Permet d'accéder au premier ancêtre correspondant au sélecteur css que nous lui avons passé en argument

Exemple

2- Se déplacer dans le DOM

La méthode children()

Permet d'accéder aux éléments enfants (les descendants directs) de l'élément ciblé. On peut également ajouter en argument un sélecteur css et dans ce cas, seules ceux correspondant aux sélecteurs seront sélectionnés.

La méthode find()

Permet d'accéder à tous les descendant (enfants, petits-enfants...) de l'élément ciblé. On doit obligatoirement ajouter en argument un sélecteur css qui servira de filtre et qui ne sélectionnera alors que les éléments correspondant à ce sélecteur. Si vous voulez choisir tous les éléments descendants, il est possible de mettre en argument le sélecteur * .

Exemple

2- Se déplacer dans le DOM

La méthode siblings()

Permet d'accéder aux éléments frères de l'élément ciblé.

Un élément frère est un élément ayant le même élément parent direct.

On peut également ajouter en argument un sélecteur css et dans ce cas, seules ceux correspondant aux sélecteurs seront sélectionnés.

La méthode next()

Permet d'accéder à l'élément frère suivant immédiatement l'élément ciblé.

On peut également ajouter en argument un sélecteur css et dans ce cas, il retournera l'élément frère suivant uniquement s'il correspond au sélecteur.

Exemple

2- Se déplacer dans le DOM

La méthode nextAll()

Permet d'accéder à tous les éléments frères suivant l'élément ciblé.

On peut également ajouter en argument un sélecteur css et dans ce cas, seules ceux correspondant aux sélecteurs seront sélectionnés.

La méthode nextUntil()

Permet d'accéder à tous les éléments frères suivant l'élément ciblé jusqu'à rencontrer l'élément frère correspondant au sélecteur mis en argument.

Dans ce cas, l'élément frère ayant le sélecteur ne sera pas sélectionné.

On peut également ajouter en 2ème argument un sélecteur css et dans ce cas, seules ceux correspondant aux sélecteurs seront sélectionnés.

Exemple

2- Se déplacer dans le DOM

Les méthodes prev(), prevAll() et prevUntil()

Fonctionnent de la même manière que next(), nextAll(), et nextUntil() mais avec les éléments frères précédents l'élément ciblé.

Exemple

2- Se déplacer dans le DOM

- Dans un fichier html créer les éléments suivant
 - Une liste ul avec 5 li dont le 4ème a la classe start
 - Sélectionner l'élément précédent le li ayant la classe start et donner lui la couleur bleu
 - Sélectionner tous les éléments du 1er li jusqu'au 4ème non inclu et donnez leurs une bordure d'1px noir
- Durée 15 minutes



2- Se déplacer dans le DOM

- Correction

[Exercice](#)

3- Manipuler les contenus HTML et textuels



La méthode html()

Sans argument, cette méthode retourne le contenu html du premier élément d'une sélection.

Avec argument, défini le contenu HTML de tous les éléments de la sélection.

La méthode text()

Sans argument, cette méthode retourne le contenu textuel de tous les éléments d'une sélection.

Avec argument, défini le contenu textuel de tous les éléments de la sélection.

Exemple

3- Manipuler les contenus HTML et textuels

- Dans un fichier html créer les éléments suivant
 - Une div avec à l'intérieur un h1 et 2p (les 2 paragraphes auront des identifiants différents)
 - Récupérer dans la console, le contenu html de la div
 - Modifier le contenu html du 1er paragraphe en mettant notamment à l'intérieur une balise strong sur un des mots du texte
 - Récupérer dans la console le contenu texte du 2ème paragraphe
 - Modifier le contenu texte de ce 2ème paragraphe
- Durée 15 minutes



3- Manipuler les contenus HTML et textuels

- Correction

[Exercice](#)

4- Insérer du contenu dans un élément



La méthode prepend()

Cette méthode permet d'insérer du contenu dans l'élément ciblé en tant que 1^{er} enfant de celui-ci.

La méthode append()

Cette méthode permet d'insérer du contenu dans l'élément ciblé en tant que dernier enfant de celui-ci.

Exemple

4- Insérer du contenu dans un élément



La méthode prependTo()

Cette méthode va permettre tout comme la méthode prepend() d'ajouter du contenu au début d'un élément mais la syntaxe n'est pas la même.

En effet, l'élément ciblé est l'élément que nous voulons ajouter à l'élément mis en argument. (la syntaxe est inversé par rapport à prepend())

La méthode appendTo()

Cette méthode va permettre tout comme la méthode append() d'ajouter du contenu à la fin d'un élément mais la syntaxe n'est pas la même.

En effet, l'élément ciblé est l'élément que nous voulons ajouter à l'élément mis en argument. (la syntaxe est inversé par rapport à append())

Ces 2 méthodes peuvent également permettre de déplacer un élément déjà existant

Exemple

4- Insérer du contenu dans un élément

- Dans un fichier html créer les éléments suivants
 - Une div avec à l'intérieur
 - Un titre h2
 - Insérer en jQuery au début de la div, un titre h1
 - Insérer en jQuery à la fin de la div, un paragraphe
- Durée 5 minutes



4- Insérer du contenu dans un élément

- Correction

Exercice



5- Insérer du contenu avant ou après un élément



La méthode before()

Cette méthode permet d'insérer du contenu avant l'élément ciblé.

La méthode after()

Cette méthode permet d'insérer du contenu après l'élément ciblé.

Exemple

5- Insérer du contenu avant ou après un élément



La méthode insertBefore()

Cette méthode permet de réaliser la même chose que la méthode before mais avec une syntaxe inversé.

Nous allons ciblé l'élément à ajouter et le mettre avant l'élément mis en argument.

La méthode insertAfter()

Cette méthode permet de réaliser la même chose que la méthode after mais avec une syntaxe inversé.

Nous allons ciblé l'élément à ajouter et le mettre après l'élément mis en argument.

Exemple

5- Insérer du contenu avant ou après un élément

- Dans un fichier html créer les éléments suivants
 - Une div avec un paragraphe
 - Insérer en jquery avant cette div, une autre div avec un titre h1
 - Insérer en jquery après la div écrite en HTML, une autre div avec un titre h2
- Durée 5 minutes



5- Insérer du contenu avant ou après un élément

- Correction

[Exercice](#)

6- Insérer du contenu autour d'un élément



La méthode wrap()

Cette méthode permet d'ajouter une structure HTML autour de chaque élément de notre sélection (ciblé).

La structure HTML est mise en argument de cette méthode.

La méthode wrapAll()

Cette méthode permet d'ajouter une structure HTML autour de l'ensemble de la sélection ciblé.

La structure HTML est mise en argument de cette méthode.

Attention, l'utilisation des méthodes wrap* peuvent entraîner des résultats étonnants

Exemple

6- Insérer du contenu autour d'un élément



La méthode wrapInner()

Cette méthode permet d'ajouter une structure HTML autour du contenu de chaque élément de la sélection ciblé.

La structure HTML est mise en argument de cette méthode.

Exemple

6- Insérer du contenu autour d'un élément

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 et 2 paragraphes
 - En jQuery, entourer les 2 paragraphes dans une même section
 - En jQuery, ajouter à cette section une bordure de 1px noir
- Durée 5 minutes



6- Insérer du contenu autour d'un élément

- Correction

Exercice



7- Copier des éléments HTML



La méthode clone()

Cette méthode permet de réaliser une copie d'un élément ou d'une sélection d'élément.

Une fois cette copie réalisé, on pourra l'insérer dans le code HTML à l'aide des méthodes vues précédemment.

Exemple

7- Copier des éléments HTML

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 et 2 paragraphes
 - En jQuery, créer une copie de la div et de ce qu'elle contient et mettez la après la 1ère div
- Durée 5 minutes



7- Copier des éléments HTML

- Correction

Exercice



8- Supprimer des éléments et des contenus HTML



La méthode remove()

Cette méthode permet de supprimer un élément ou un ensemble d'élément et tout ce qu'il contient.

On peut spécifier en argument un sélecteur css pour ne supprimer que les éléments correspondant à ce sélecteur.

La méthode empty()

Cette méthode permet de vider le contenu des éléments sélectionnés sans supprimer les éléments eux-même

La méthode unwrap()

Cette méthode permet de supprimer l'élément parent d'un ou de plusieurs éléments. On peut spécifier en argument un sélecteur css pour ne supprimer que les éléments parents correspondant à ce sélecteur.

8- Supprimer des éléments et des contenus HTML

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 et 2 paragraphes
 - En jQuery, supprimer la div sans supprimer les paragraphes
 - En jQuery, supprimer le contenu du titre h1
 - En jQuery, supprimer le 2ème paragraphe
- Durée 5 minutes



8- Supprimer des éléments et des contenus HTML

- Correction

Exercice



9- Remplacer des éléments HTML

La méthode replaceWith()

Cette méthode permet de remplacer l'élément ciblé par l'élément mis en argument.

La méthode replaceAll()

Cette méthode fait la même chose que replaceWith() mais dans l'autre sens.

L'élément ciblé est l'élément que nous allons ajouter à la place de l'élément mis en argument.

Exemple

9- Remplacer des éléments HTML

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 et 2 paragraphes
 - En jQuery, remplacer le 1 er paragraphe par un titre h2
- Durée 5 minutes



9- Remplacer des éléments HTML

- Correction

Exercice



10- Récupérer la valeur d'un attribut ou définir un attribut



La méthode attr()

Cette méthode permet plusieurs choses :

- Récupérer la valeur d'un attribut
- Modifier la valeur d'un attribut
- Créer un nouvel attribut et lui associer sa valeur

Exemple

10- Récupérer la valeur d'un attribut ou définir un attribut

- Dans un fichier html créer les éléments suivants
 - Une div avec un lien vers la page de Google
 - En jQuery, récupérer la valeur de l'attribut href pour l'afficher sur la console
 - En jQuery, ajouter à ce lien l'attribut target avec la valeur `_blank`
- Durée 5 minutes



10- Récupérer la valeur d'un attribut ou définir un attribut

- Correction

Exercice



11- La méthode prop() et les propriétés du DOM



La méthode prop()

Cette méthode permet plusieurs choses :

- Récupérer la valeur d'une propriété du premier élément de notre sélection attribut
- Modifier la valeur d'une propriété de chaque éléments d'une sélection
- Créer une nouvelle propriété pour chaque éléments d'une sélection et lui associer sa valeur

Exemple

11- La méthode prop() et les propriétés du DOM



Différence entre attribut et propriété

Un attribut sert à fournir des informations supplémentaires par rapport à un élément et à son comportement. Un attribut est toujours défini dans notre fichier HTML de départ.

Une fois qu'un fichier HTML est chargé dans un navigateur, ce dernier va créer un DOM qui est une représentation de notre fichier de départ. A ce stade, en Javascript, notre fichier HTML va être vu sous la forme de noeuds ou "node" qui sont des objets. Les attributs sont alors représentés sous forme de propriétés de nos noeuds éléments.

On a donc ici deux représentations qui coexistent : notre fichier HTML avec ses éléments et ses attributs d'un côté, et le DOM avec ses noeuds et ses propriétés. Ces deux représentations existent en même temps.

L'idée ici est qu'on va ensuite pouvoir modifier le DOM, et donc modifier les différents objets et les valeurs des propriétés de celui-ci sans forcément modifier la valeur des attributs de notre fichier HTML.

Notez bien qu'il ne va y avoir de différence entre la valeur de l'attribut et celle de la propriété que pour certains éléments HTML. (les éléments dynamiques, cases à cocher...

Exemple

11- La méthode prop() et les propriétés du DOM

- Dans un fichier html créer les éléments suivants
 - Une div avec un formulaire et dans ce formulaire un input de type checkbox
 - En jQuery, récupérer la valeur de la propriété checked
 - En jQuery, modifier la valeur de cette propriété
- Durée 5 minutes



11- La méthode prop() et les propriétés du DOM

- Correction

Exercice



12- Récupérer ou mettre à jour la valeur courante d'éléments



La méthode val()

Cette méthode va permettre de récupérer ou de mettre à jour la valeur courante d'éléments.

Cette méthode est principalement utilisée avec des éléments de formulaire.

On va notamment pouvoir l'utiliser pour récupérer la valeur (le "contenu") d'un champ de texte input ou textarea ou encore la valeur liée à une case cochée ou à une option d'une liste select choisie.

Exemple

13- Les méthodes permettant de manipuler spécifiquement les attributs class



La méthode hasClass()

Cette méthode va permettre de déterminer si au moins l'un des éléments d'une sélection possède la classe mise en argument. Retourne true ou false

La méthode addClass()

Cette méthode va permettre d'ajouter une ou plusieurs classes à une sélection d'éléments. On va passer les classes à ajouter en argument.

13- Les méthodes permettant de manipuler spécifiquement les attributs class



La méthode removeClass()

Cette méthode va permettre de supprimer une, plusieurs ou toutes les classes de chaque élément d'une sélection. Si on utilise cette méthode sans argument, toutes les classes des éléments de la sélection seront supprimées. Si on précise une ou plusieurs classes en argument, alors seules ces classes seront supprimées.

La méthode toggleClass()

Cette méthode va permettre d'ajouter une ou plusieurs classes à chaque élément d'une sélection si ces éléments ne possèdent pas ces classes ou au contraire de supprimer ces classes si les éléments les possèdent.

13- Les méthodes permettant de manipuler spécifiquement les attributs class

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 avec la classe monTitre et un paragraphe ainsi qu'un bouton
 - En jQuery, vérifier si h1 a bien la classe monTitre et afficher le résultat dans la console
 - En jQuery, ajouter la classe maClass au paragraphe
 - En jQuery, supprimer la classe monTitre au titre H1
 - En jQuery, ajouter ou enlever la classe red en cliquant sur le bouton (la classe red mets le texte en rouge)
- Durée 10 minutes



13- Les méthodes permettant de manipuler spécifiquement les attributs class

- Correction

Exercice



14- La méthode css()



La méthode css()

Cette méthode peut être utilisée de 2 manières :

- Pour récupérer la valeur liée à une propriété
 - Pour cela, la méthode ne prendra qu'un seul argument, celui de la propriété dont nous voulons récupérer la valeur
- Pour définir des propriétés css et leurs valeurs
 - Pour cela, la méthode prendra 2 arguments, le 1^{er} la propriété css et en 2^{ème} sa valeur
 - Il est également possible de définir la valeurs de plusieurs propriétés css en même temps à l'aide de la syntaxe suivante
 - `css({"prop1 : valeur", "prop2 : valeur"})`

14- La méthode css()



Il est préférable à chaque fois que cela est possible d'utiliser plutôt les classes que la méthode css afin de séparer au maximum le code du style et améliorer ainsi la maintenabilité.

14- La méthode css()

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1
 - En jQuery, modifier le titre pour qu'il soit en italique et soulignez-le
- Durée 5 minutes



14- La méthode css()

- Correction

Exercice



15- Les méthodes liées aux dimensions des éléments



La méthode width()

Permet de récupérer la largeur de la boîte élément du premier élément de la sélection ou bien de définir cette valeur pour tous les éléments de la sélection en lui passant dans ce cas la largeur sans unité en argument.

La méthode innerWidth()

Permet d'obtenir la largeur de la boîte élément et du padding (marge intérieure) du premier élément d'une sélection ou de définir cette largeur pour tous les éléments de la sélection en lui passant dans ce cas la largeur sans unité en argument.

15- Les méthodes liées aux dimensions des éléments



La méthode `outerWidth()`

Permet d'obtenir la largeur de la boîte élément, du padding (marge intérieure) et des bordures du premier élément d'une sélection ou de définir cette largeur pour tous les éléments de la sélection en lui passant dans ce cas la largeur sans unité en argument.

Notez qu'on va également pouvoir inclure la marge extérieure (margin) dans le calcul lors de la récupération de la largeur en passant le booléen `true` en argument de cette méthode.

15- Les méthodes liées aux dimensions des éléments

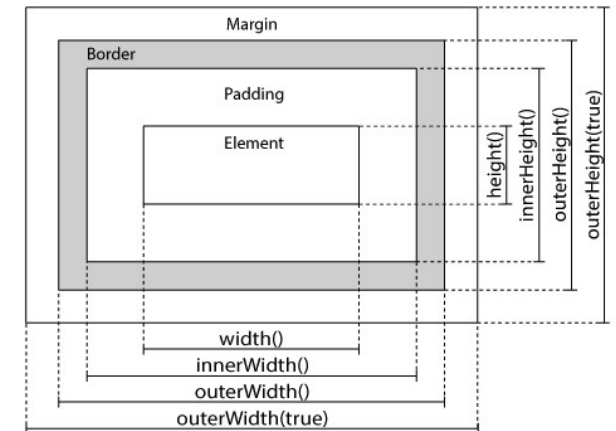


La hauteur

Les méthodes concernant la hauteur sont les suivantes:

- `height()`
- `innerHeight()`
- `outerHeight()`

Elles fonctionnent exactement comme les méthodes `width()`, `innerWidth()` et `outerWidth()`.



Exemple

15- Les méthodes liées aux dimensions des éléments

- Dans un fichier html créer les éléments suivants
 - Une div
 - En jQuery, modifier la div pour qu'elle ai un background bleu, une largeur de 400px et une hauteur de 200px
- Durée 5 minutes



15- Les méthodes liées aux dimensions des éléments

- Correction

Exercice



16- La méthode offset()



La méthode offset()

Permet de récupérer la position courante du premier élément d'une sélection ou de redéfinir la position de tous les éléments de la sélection par rapport au document.

La position récupérée ou définie par offset() va être celle du coin supérieur gauche de la boîte de bordure (boîte incluant le contenu, les marges internes et la bordure) des éléments.

La méthode offset() va renvoyer un objet qui contient deux propriétés top et left qui contiennent les coordonnées de l'élément par rapport aux bords supérieur et gauche du document.

16- La méthode offset()

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1
 - En jQuery, repositionner la div en lui ajoutant 100px à sa valeur top et 200px à sa valeur left par rapport au document
- Durée 10 minutes



16- La méthode offset()

- Correction

Exercice



17- La méthode each()



La méthode each()

La méthode each() permet d'itérer dans une sélection comme une boucle.

Elle reçoit en paramètre une fonction de rappel.

Elle peut permettre également d'itérer sur un tableau.



III. Gestion d'évènement avec jQuery

1- La gestion des événements avec jQuery



jQuery possède des méthodes correspondant au nom de l'évènement (exemple `click()`, `mouseenter()`, `keyup()`, `change()`, ...

jQuery possède également une méthode `on()` dont la syntaxe ressemble beaucoup à la syntaxe de la méthode `addEventListener()` en Javascript.

La méthode `on()` ne peut par défaut créer de gestionnaire d'événements que pour des éléments qui existent au moment où on met en place le gestionnaire.

Si on crée un gestionnaire d'événements avec `on()` puis qu'on crée ensuite un élément, le gestionnaire d'événements ne s'appliquera donc pas à cet élément.

2- Les méthodes liées à la souris **jQuery** write less, do more.

La méthode click()

Permet de gérer les événements au clic

La méthode dblclick()

Permet de gérer les événements au double clic

La méthode mouseenter()

Permet de gérer les événements lorsque la souris entre dans l'élément ciblé

La méthode mouseleave()

Permet de gérer les événements lorsque la souris sort de l'élément ciblé

La méthode hover()

Permet de gérer l'entrée et la sortie de l'élément ciblé. La méthode contiendra en argument 2 fonctions (1 pour l'entrée, et l'autre pour la sortie)

Exemple

2- Les méthodes liées à la souris

La méthode mousedown()

Permet de gérer les évènements lorsque le bouton de la souris est pressé sur l'élément ciblé

La méthode mouseup()

Permet de gérer les évènements lorsque le bouton de la souris est relâché sur l'élément ciblé

La méthode mousemove()

Permet de gérer les évènements lorsque la souris bouge.

Cette évènement possède un objet event qui a les propriétés pageX et pageY qui sont les coordonnées de la souris par rapport au coin supérieur gauche du document.

2- Les méthodes liées à la souris

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 et un bouton ainsi qu'un paragraphe
 - En jQuery, lorsque vous cliquez sur le bouton faites que le titre soit en bleu et qu'il redevienne noir lorsque vous recliquez sur le bouton
 - En jQuery, lorsque vous survolez le paragraphe celui-ci doit avoir une taille de police de 22px puis doit revenir à 16px lorsque vous arrêtez de le survoler
- Durée 15 minutes



2- Les méthodes liées à la souris

- Correction

Exercice



3- Les méthodes liées au clavier

La méthode keydown()

Permet de gérer les évènements lorsqu'une touche du clavier est pressée.

La méthode keyup()

Permet de gérer les évènements lorsqu'une touche du clavier est relâchée.

La méthode keypress()

Cette méthode est similaire à la méthode keydown() à l'exception des 2 points suivants:

- L'évènement keydown se déclenche avant keypress
- La méthode keypress ne sera pas déclenché par toutes les touches du clavier (alt, ctrl, shift, etc.) car c'est un évènement créé à la base pour capturer la saisie de caractères.

Ces évènements possèdent une propriété which() qui permet de récupérer la touche utilisée au format code ou bien key() qui récupère au format lettre

3- Les méthodes liées au clavier

- Dans un fichier html créer les éléments suivants
 - Une div avec un formulaire et dans ce formulaire un champ input de type text. Il y aura sous le formulaire un paragraphe
 - En jQuery, récupérer le texte saisi dans l'input et l'afficher le au fur et à mesure dans le paragraphe.
 - PS: la méthode text() permet de modifier le contenu text d'un élément
- Durée 15 minutes



3- Les méthodes liées au clavier

- Correction

Exercice



4- Les méthodes spécifiques aux formulaires



Un élément a le focus lorsqu'il est l'élément actif du document c'est-à-dire l'élément dans lequel le curseur de la souris est bloqué. Lorsqu'on clique dans un champ de formulaire, par exemple, cet élément de champ de formulaire possède le focus.

La méthode focus()

Permet de gérer les événements lorsque l'élément gagne le focus.

La méthode focusin()

Permet de gérer les événements lorsque l'élément ou l'un de ses descendants gagne le focus.

4- Les méthodes spécifiques aux formulaires



La méthode blur()

Permet de gérer les évènements lorsque l'élément perd le focus.

La méthode focusout()

Permet de gérer les évènements lorsque l'élément ou l'un de ses descendants perd le focus.

4- Les méthodes spécifiques aux formulaires



La méthode change()

L'événement change se déclenche lorsque la valeur d'un élément change.

Cet événement ne peut se déclencher qu'avec des éléments input, textarea et select.

L'événement se déclenche dès qu'un utilisateur fait une sélection dans le cas de cases à cocher (checkbox), de boutons radio, ou de liste d'option.

La méthode submit()

L'événement submit se déclenche dès qu'un utilisateur tente d'envoyer un formulaire. On ne va pouvoir intercepter cet événement que depuis l'élément form.

4- Les méthodes spécifiques aux formulaires

- Dans un fichier html créer les éléments suivants
 - Une div avec un formulaire et dans ce formulaire un champ input de type text ainsi qu'un bouton submit. En dessous du formulaire, un paragraphe indiquant "Le formulaire a été soumis" mais qui n'est pas visible.
 - En jQuery, lorsque l'input a le focus, modifier la couleur de la saisie en vert et faites que la couleur redevienne noire lorsque l'input perd le focus
 - En jQuery, lorsque le formulaire est soumis, rendre visible le paragraphe et invisible le formulaire
- Durée 20 minutes



4- Les méthodes spécifiques aux formulaires

- Correction

Exercice



5- Les méthodes on() et off()



La méthode on()

Cette méthode peut remplacer toutes les méthodes précédentes et possède une syntaxe similaire à la méthode Javascript `addEventListener()`

Cette méthode peut gérer également plusieurs évènements différents pour un même élément en même temps.

La méthode off()

Cette méthode va permettre de supprimer des gestionnaires d'évènements.

5- Les méthodes on() et off()

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 ainsi qu'un bouton
- En jQuery, en utilisant la méthode on(), lorsque je survole le titre le mettre à une police de 40px et revenir à la taille initiale quand je ne la survole plus
- En jQuery, lorsque je clique sur le bouton, supprimer l'évènement precedent afin qu'il ne soit plus utilisable
- Durée 15 minutes



5- Les méthodes on() et off()

- Correction

Exercice



6- Déclencher un évènement avec trigger et triggerHandler()



Les méthodes trigger() et triggerHandler()

Ces méthodes vont nous permettre de déclencher “manuellement” un évènement. On va leur passer en argument le nom de l’évènement qu’on souhaite déclencher.

Différence entre trigger() et triggerHandler()

- La méthode triggerHandler(), à la différence de trigger(), va exécuter le gestionnaire d’évènement de l’évènement déclenché mais ne va pas déclencher l’action par défaut de l’évènement. Par exemple, triggerHandler("submit") n’enverra pas le formulaire
- trigger() fonctionnera sur tous les éléments possédant les caractéristiques ciblées tandis que triggerHandler() n’affectera que le premier élément ciblé rencontré
- les événements déclenchés avec triggerHandler() ne bouillonneront pas c’est-à-dire qu’elle ne se propagera pas aux éléments plus haut

6- Déclencher un événement avec trigger et triggerHandler()

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 ainsi que 2 boutons
- En jQuery, modifier la couleur du titre h1 au survol et revenir à sa couleur d'origine quand on ne survole plus le titre
- En jQuery, lorsque je clique sur le 1er bouton, déclencher l'évènement de survol du titre
- En jQuery, lorsque je clique sur le 2ème bouton, déclencher l'évènement lorsqu'on sort du titre
- Durée 15 minutes



6- Déclencher un événement avec trigger et triggerHandler()

- Correction

Exercice



7- Les propriétés de l'objet event **jQuery** write less, do more.

La propriété target

Cette propriété va nous permettre de connaître l'élément qui a déclenché l'événement. Cette propriété va s'avérer très utile dans des situations de bouillonnement d'un événement (propagation de l'évènement), afin de déterminer quel est l'élément déclencheur.

La propriété type

Cette propriété va nous permettre de connaître le type d'événement déclenché. Cette propriété peut être très utile dans le cas où plusieurs gestionnaires pour plusieurs événements sont accrochés à un même élément.



IV. Création d'effets et d'animations avec jQuery

1- Créer des effets de fondu en jQuery write less, do more.

La méthode fadeOut()

Cette méthode va permettre de cacher les éléments ciblés en les rendant progressivement transparent puis en leur appliquant à la fin un `display:none`.

La méthode fadeIn()

Cette méthode va permettre d'afficher progressivement des éléments cachés en les opacifiant progressivement.

La méthode fadeToggle()

Cette méthode va permettre d'inverser l'état de visibilité d'un élément, en faisant apparaître progressivement un élément caché au départ ou au contraire en faisant progressivement disparaître un élément opaque au départ avec un effet de fondu.

1- Créer des effets de fondu en jQuery *write less, do more.*

La méthode fadeTo()

Cette méthode va permettre de définir un niveau d'opacité seuil (entre 0 et 1) auquel l'élément devra parvenir progressivement. Si l'élément est plus transparent au départ que ce niveau, il s'opacifiera progressivement et inversement.

1- Créer des effets de fondu en jQuery

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre h1 ainsi que 3 h2 avec pour chque h2, un paragraphes caché
- En jQuery, lorsqu'on clique sur le h2, son paragraphe s'affiche, lorsqu'on re-clique sur le h2, son paragraphe disparaît
- Durée 10 minutes



1- Créer des effets de fondu en jQuery

- Correction

[Exercice](#)

2- Créer des effets de slide en jQuery *write less, do more.*

La méthode slideDown()

Cette méthode va permettre de “déplier” un élément progressivement.

La méthode slideUp()

Cette méthode va permettre de “replier” un élément progressivement.

La méthode slideToggle()

Cette méthode va permettre de déplier progressivement un élément plié ou au contraire de replier progressivement un élément déplié.

2- Créer des effets de slide en jQuery

- Dans un fichier html créer les éléments suivants
 - un header avec une nav et des éléments du menu ainsi que des éléments de sous-menu
- En jQuery, lorsqu'on survole l'un des éléments du menu ayant un sous menu, ce dernier s'affiche. Il disparaît lorsqu'on ne le survole plus
- Durée 20 minutes



2- Créer des effets de slide en jQuery

- Correction

Exercice



3- Afficher ou cacher un contenu avec show(), hide() et toggle()



La méthode show()

Cette méthode va permettre d'afficher un contenu HTML caché.

La méthode hide()

Cette méthode va permettre de cacher un contenu HTML visible.

La méthode toggle()

Cette méthode va permettre d'afficher un contenu HTML caché ou cacher un contenu HTML visible.

3- Afficher ou cacher un contenu avec show(), hide() et toggle()

- Dans un fichier html créer les éléments suivants
 - Une div avec 2 titres h2 et en dessous de chacun de ces h2, 2 paragraphes (le 2ème paragraphe étant cache). Ajouter ensuite un bouton avec écrit "Lire la suite"
- En jQuery, lorsqu'on clic sur le bouton "Lire la suite", le 2ème paragraphe apparaît et le bouton "Lire la suite" devient "Replier". Lorsqu'on re-clique sur le bouton tout redevient comme initialement
- Durée 25 minutes



3- Afficher ou cacher un contenu avec show(), hide() et toggle()

- Correction

[Exercice](#)

4- Créer des effets ou animations personnalisés avec jQuery



La méthode animate()

Cette méthode va permettre de créer des effets ou d'animer n'importe quelle propriété CSS à valeur numérique. On va lui passer les propriétés qu'on souhaite animer en argument obligatoire.

Limitation de la méthode animate()

Cette méthode ne permet pas d'animer des propriétés CSS non numérique comme la couleur à moins d'utiliser l'extension jQuery.color.

Les notations raccourcies des propriétés CSS comme font, border, etc. vont être mal supportées par animate(). On préférera donc toujours utiliser les versions complètes de ces propriétés lorsqu'on voudra les animer.

4- Créer des effets ou animations personnalisés avec jQuery



La file d'attente ou queue des animations

Il est possible de chaîner les animations.

Dans ce cas, chaque animation attendra que la précédente se termine avant de s'exécuter. Cependant même comme cela il est possible de forcer une animation de s'exécuter tout de suite avec l'option `queue`. Dans ce cas, cette animation n'est pas dans la file d'attente d'exécution.

4- Créer des effets ou animations personnalisés avec jQuery

- Dans un fichier html créer les éléments suivants
 - Une div avec un titre H1 et un bouton
- En jQuery, lorsqu'on clic sur le bouton le titre doit s'agrandir jusqu'à 40px pendant 2 secondes puis automatiquement revenir à 32px pendant 2 secondes
- Durée 10 minutes



4- Créer des effets ou animations personnalisés avec jQuery

- Correction

Exercice



5- Gérer la file d'attente en jQuery *write less, do more.*

La méthode queue()

Cette méthode retourne les différentes fonctions de la file d'attente pour un élément ou va nous permettre de manipuler la file d'attente de différentes façons selon le nombre d'arguments qu'on va lui passer.

La méthode queue() pour récupérer des infos

Nous allons pouvoir utiliser cette méthode pour récupérer des informations comme le nombre d'élément dans la file d'attente et notamment le nombre d'élément dans cette file d'attente

5- Gérer la file d'attente en jQuery write less, do more.

La méthode `queue()` pour manipuler la file d'attente

La méthode peut avoir comme argument une fonction callback qui peut permettre ainsi d'inclure dans la file d'attente des méthodes qui ne le sont pas normalement.

Il est important de chaîner cette méthode avec à la fin la méthode `dequeue()` afin que le reste du code puisse s'exécuter.

La méthode `clearQueue()`

Cette méthode va permettre de supprimer toutes les fonctions actuellement dans la file d'attente.

6- Stopper une animation avec les méthodes jQuery stop() ou finish()



La méthode stop()

Cette méthode va par défaut arrêter l'animation en cours.

Cette méthode peut également prendre un argument qui est par défaut false, et qui peut permettre de supprimer la file d'attente en mettant la valeur true.

Cette méthode peut également prendre en 2^{ème} argument, un booléen qui indique si l'animation en cours doit être complétée immédiatement (true) ou pas (false)

La méthode finish()

Cette méthode va par défaut stopper immédiatement l'animation en cours et supprimer toutes les animations dans la file d'attente.

La méthode finish() diffère de la méthode stop() en un point : lorsqu'on l'utilise, les valeurs relatives aux propriétés CSS définies dans les animations vont être immédiatement définies sur leurs valeurs finales.

Exemple

7- Ajouter un délai avant l'exécution d'animations avec delay()



La méthode delay()

Cette méthode va permettre de définir un temps avant lequel certaines animations ne doivent pas être lancées. On va très souvent chaîner cette méthode entre deux animations. L'argument est le temps en ms.



I. AJAX avec jQuery

1- Création de requêtes Ajax avec jQuery



La méthode \$.ajax

Cette méthode va prendre un objet de configuration en argument ou vous pourrez notamment pouvoir spécifier :

- l'url avec la propriété `url`
- La méthode (par défaut GET) avec la propriété `method`
- Le type de donnée (avec la propriété `dataType`) qu'on attend de recevoir
- Les données à envoyer au serveur dans le cas d'une requête de type POST avec la propriété `data`
- `contentType` permet d'indiquer le type de données lors d'un envoi au serveur (requête de type POST)
- ...

1- Création de requêtes Ajax avec jQuery



Une fois la requête envoyée, on va vouloir récupérer la réponse du serveur et prendre en charge les différents cas de succès ou d'échec de cette requête.

La méthode `done()`

La méthode `done()` va contenir une fonction de rappel qui sera exécutée si la requête aboutit.

Cette fonction reçoit les données de réponse (converties en un objet JavaScript si le type de données est JSON).

La méthode `fail()`

La méthode `fail()` va contenir une fonction de rappel qui sera exécuté si la requête n'aboutit pas.

1- Création de requête Ajax avec jQuery

- Créer à l'aide de l'API <https://jsonplaceholder.typicode.com/todos> une requête AJAX pour récupérer une liste de Todos
- Durée 10 minutes



1- Création de requête Ajax avec jQuery

- Correction

[Exercice](#)