Formation

jQuery

m2iformation.fr

Sommaire

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Introduction

Qu'est-ce que jQuery ?

- jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client
- Elle contient de nombreuses fonctionnalités, notamment des animations, la manipulation des feuilles de style en cascade, la gestion des évènements ...
- L'utilisation d'Ajax est facilitée et de nombreux plugins sont présents

Histoire de jQuery

- créé en 2006 par John Resign
- jQuery suit le principe *"écris moins, fais plus"*
- jQuery est utilisé par des milliers d'applications dans le monde

Installation

1. Via téléchargement : jQuery
2. Avec NodeJS : `npm install jquery`
3. Avec un CDN : jQuery

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Les bases

Exécution du code

- Pour exécuter du code dès que le DOM est prêt à être manipulé, jQuery a une instruction appelée ready event
- Par défaut jQuery utilise `$` comme raccourci à `window.jQuery`

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

```
// syntaxe alternative
$(function () {
  console.log("document prêt!");
});
```

Sélection d'éléments

- la syntaxe `$(selecteur)` retourne un objet jQuery
- Sélection par id
  - `$("#myId");`
- Sélection par classe
  - `$(".myClass");`
- Sélection par attribut
  - `$("input[name='first_name']");`

La méthode .has()

La méthode has() permet d'affiner une sélection en ne sélectionnant que les éléments d’une sélection initiale qui possèdent des éléments enfants correspondant au sélecteur passé en argument de cette fonction

```
$(document).ready(function () {
  $("p").has("span").css("color", "orange");
});
```

La méthode .filter()

La méthode jQuery filter() permet d'affiner une sélection en ne sélectionnant que les éléments qui correspondent au sélecteur passé en argument de cette fonction

```
$(document).ready(function () {
  $("span").filter(".souligne").css("color", "green");
});
```

Les méthodes .first(), .last() et .eq()

- Les méthodes `first()` et `last()` permettent de sélectionner le premier et dernier élément
- La méthode `eq()` permet de sélectionner un élément par rapport à sa position

```
$(document).ready(function () {
  $("span").first().css("color", "green");
  $("span").eq(1).css("color", "blue");
  $("span").last().css("color", "orange");
});
```

Enchaînement

- Lorsque l'on appelle une méthode sur un objet jQuery, celui-ci renvoie un objet jQuery
- Ainsi il est possible d'enchaîner les méthodes, cette pratique est appelée « chaînage » :

```
$("#content").find("h3").eq(2).html("new text for the third h3!");
```

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Manipulation d'éléments

Méthodes d'obtention d'informations

- .html() : Obtenir ou définir le contenu HTML
- .text() : Obtenir ou définir le contenu du texte
- .attr() : Obtenir ou définir la valeur de l’attribut fourni
- .width() : Obtenir la largeur en pixels du premier élément
- .height() : Obtenir la hauteur en pixels du premier élément
- .position() : Position par rapport à son première ancêtre
- .val() : Obtient ou définit la valeur des éléments de formulaire

Déplacement, copie et suppression d'éléments

.insertAfter()
.insertBefore()
.appendTo()
.prependTo()

```
// Déplacer le première item de la liste à la fin
let li = $("#myList li:first").appendTo("#myList");
```

Création d'élément

- Création de nouveaux éléments en utilisant la même méthode que celle utilisée pour effectuer des sélections

```
// Créer un élément avec une chaîne html
$("<p>This is a new paragraph</p>");
$('<li class="new">new list item</li>');
```

```
// Créer un élément avec des attributs
$("<a/>", {
  html: "This is a <strong>new</strong> link",
  class: "new",
  href: "foo.html",
});
```

Ajouter un élément au DOM

- Lorsque l'on créé un nouvel élément, il n’est pas immédiatement ajouté à la page
- Il existe plusieurs façons d’ajouter un élément à la page une fois qu’elle a été créée
- En cas d'ajout de multiples éléments HTML, il vaut mieux les concaténer puis les ajouter en une seule fois

```
var myNewElement = $("<p>New element</p>");

myNewElement.appendTo("#content");
```

Manipulation des attributs

- Manipulation simple avec une valeur

```
$("#myDiv a:first").attr("href", "newDestination.html");

$("#myDiv a:first").attr({
  href: "newDestination.html",
  rel: "nofollow",
});
```

- Manipulation à l'aide d'une fonction

```
$("#myDiv a:first").attr("href", function (idx, href) {
  return "/new/" + href;
});
```

Déplacement dans le DOM

- Les méthodes parent(), parents(), parentsUntil() et closest() vont nous permettre d'accéder aux parent et ancêtres d'un élément ou d'une collection d'éléments.
- Les méthodes children() et find() vont nous permettre d'accéder aux enfants et descendants d'un élément ou d'une collection d'éléments précédemment sélectionnés
- Pour sélectionner les éléments "frères" il faut utiliser les méthodes siblings(), next(), nextAll(), nextUntil(), prev(), prevAll() et prevUntil().

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

CSS, style et dimensions

Obtenir et définir des propriétés CSS

La gestion des propriétés CSS se fait avec la méthode `.css`

- Récupération d'une propriété

```
$("h1").css("fontSize");
$("h1").css("font-size");
```

- Modification de propriétés

```
$("h1").css("fontSize", "100px");

$("h1").css({
  fontSize: "100px",
  color: "red",
});
```

utilisation de classe pour le style

- Ajouter une classe

```
h1.addClass("big");
```

- Supprimer une classe

```
h1.removeClass("big");
```

- Ajouter/Supprimer une classe

```
h1.toggleClass("big");
```

- Vérifier si une classe existe

```
h1.hasClass("big");
```

Taille des éléments

- Modifier/récupérer la largeur d'un élément

```
$("h1").width("50px");
$("h1").width();
```

- Modifier/récupérer la hauteur d'un élément

```
$("h1").height("50px");
$("h1").height();
```

- Récupérer la position relative d'un élément par rapport à son parent

```
$("h1").position();
```

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Méthodes utilitaires

Méthodes utiles

- Supprimer les espaces de début et fin

```
$.trim("    lots of extra whitespace    ");
```

- Itérer sur des tableaux et des objets

```
$.each(["toto", "tata", "titi"], (k, v) => {
  console.log(`${k} : ${v}`);
});
```

- Renvoie l'index d'une valeur dans un tableau, ou -1 si non trouvé

```
if ($.inArray(4, myArray) !== -1) {
  console.log("found it!");
}
```

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Les événements

Gestion d'événements

- jQuery permet la gestion d'événements avec la méthode `.on()`

```
$("p").on("click", function () {
  console.log("click");
});
```

- La librairie intègre également de nombreux raccourcis : `.click()`, `.focus()`, `.blur()`, `.change()`

```
$("p").click(function () {
  console.log("You clicked a paragraph!");
});
```

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Les effets

Introduction

- La bibliothèque **jQuery** fournit plusieurs techniques pour ajouter une animation à une page Web
- Ceux-ci incluent des **animations simples** et standard fréquemment utilisées et la possibilité de créer des **effets personnalisés** sophistiqués

Afficher et masquer du contenu

- jQuery permet d'afficher ou masquer le contenu instantanément avec :

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

$("div.hidden").show();
```

- Lorsque jQuery masque un élément, il affecte à sa propriété CSS `display` la valeur `none`
- Ces méthodes peuvent prendre les arguments: `slow` `normal` `fast`

Animations de fondu et de diapositive

- Le slide effect s'applique sur la propriété CSS `height`

```
$("p").slideUp(800);
$("div.hidden").slideDown(600);
```

- Le fade effect s'applique lui sur la propriété CSS `opacity`

```
$("p").fadeOut(1500);
$("div.hidden").fadeIn(750);
```

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Ajax

Histoire Ajax

- Traditionnellement, les pages Web devaient être rechargées pour mettre à jour leur contenu
- Cela présentait d'énormes inconvénients : c'était **lent** et cela nécessitait **l'intervention de l'utilisateur**
- En 2003, tous les principaux navigateurs ont résolu ce problème en adoptant l'objet XMLHttpRequest
- L'objet XMLHttpRequest fait partie d'une technologie appelée Ajax (Asynchronous JavaScript and XML)

Fonctionnement Ajax

- Les requêtes Ajax sont déclenchées par du code JavaScript
- Le code envoie une requête à une URL, et lorsqu'il reçoit une réponse il exécute une callback
- Étant donné que la demande est **asynchrone**, le code continue de s'exécuter
- La plupart des applications jQuery n'utilisent pas XML mais plutôt des données au format HTML ou JSON

GET vs POST

GET

- La méthode GET doit être utilisée pour récupérer des données du server
- Les requêtes GET peuvent être mises en cache par le navigateur
- Les arguments sont passés dans l'URL

POST

- La méthode POST est utiliser pour modifier/créer des données sur le serveur
- Les requêtes ne sont généralement pas mise en cache
- Les arguments sont passées dans le corps de la requête

Méthode Ajax

- La méthode `$.ajax()` de jQuery est un moyen puissant et simple de créer des requêtes Ajax
- elle permet d'exécuter une callback spécifique en cas de réussite ou d'échec

```
$.ajax({
  url: "users.php",
  type: "GET",
  dataType: "json",
}).done(function (json) {
  console.log(json);
});
```

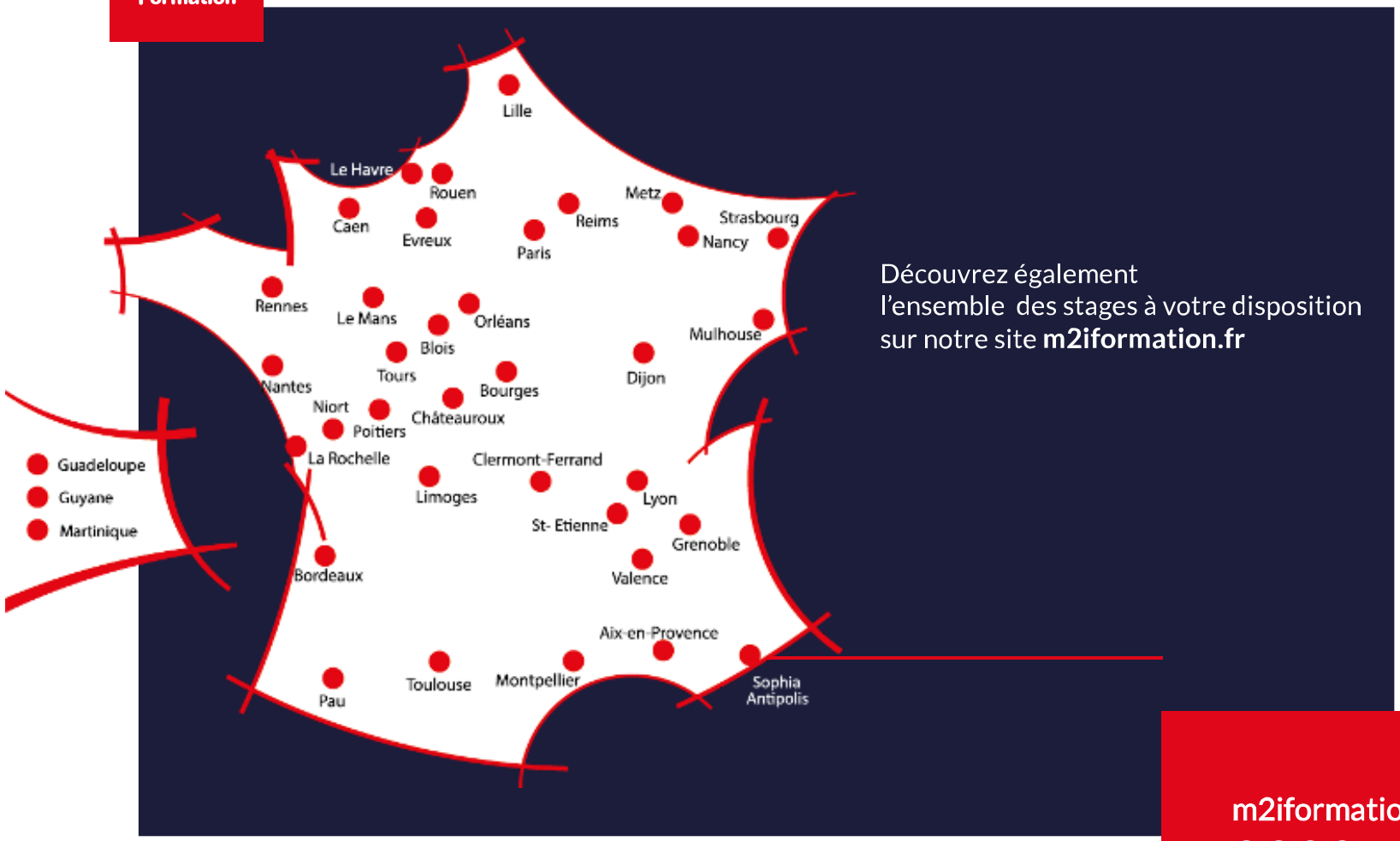
Méthodes de commodités

- jQuery fournit des méthodes raccourcis pour effectuer des appels Ajax dans le cas où la gestion des erreurs n'est pas importantes
- **$.get** : Effectue une requête GET
- **$.post** : Effectue une requête POST
- **$.getJSON** : Effectue une requête GET et attend un JSON
- **$.getScript** : Ajoute un script à la page

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Merci pour votre attention

Des questions ?



Découvrez également
l'ensemble des stages à votre disposition
sur notre site **m2iformation.fr**

m2iformation.fr