

Technologies Web

Pr. Fatima MOURCHID



Frameworks

Javascript



Frameworks Javascript: jQuery

JQuery

- ▶ Bibliothèque JavaScript open-source et multiplateforme créée pour faciliter l'écriture de scripts côté client
 - ▶ Première version lancée en janvier 2006 par John Resig
 - ▶ Comportement classique des pages Web avec javascript
 - ▶ Avantages dont notamment la flexibilité de la gestion de compatibilité inter-navigateurs
 - ▶ Fonction de cette bibliothèque est la fonction : '\$'

Inclure JQuery

- ▶ Télécharger une copie et l'inclure dans les fichiers du projet Web <https://jquery.com/download/>

```
<script type="text/javascript" src="CheminVersLeFichierjQuery.js"> </script>
```

- ▶ Inclure JQuery via CDN (Content Delivery Network)

- ▶ CDN jQuery
- ▶ CDN Google
- ▶ CDN Microsoft

```
<script src="//code.jquery.com/jquery-3.6.0.min.js">
```

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

CDN: réseaux d'ordinateurs qui coopèrent afin de mettre à disposition du contenu ou des données à des utilisateurs

Fonctionnement JQuery

- ▶ Sélectionner des éléments dans une page Web
- ▶ Intégrer des événements
- ▶ Utiliser des méthodes jQuery

Sélectionner des éléments d'une page Web

- ▶ jQuery permet d'identifier et de sélectionner des éléments (sélecteurs de base)
 - ▶ Sélectionner par type d'élément : h1,h2,div, p,img, etc.
`$('p')` ou `jQuery('p')` : sélectionner un paragraphe `<p> </p>`
 - ▶ Sélectionner par classe (avec l'attribut class=): .starred,.featured, etc.
`$('.starred')` ou `jQuery('.starred')` : sélectionner un paragraphe `<p> </p>`
 - ▶ Sélectionner par ID: #about, #contact, etc.
`$("#paragraphe1")` : sélectionner un paragraphe dont l'id 'paragraphe1'
 - ▶ Sélectionner tous les éléments: `$("*")`
 - ▶ Sélectionner des éléments avec différents sélecteurs
`$(".starred, p, #paragraphe1")`

Sélectionner des éléments d'une page Web

► Types de sélection

► Sélection par relation

`$("ancêtre descendant")`

`$("parent > enfant")`

`$(élément ~ frères)`

Exemple : `$("ol li")` // sélectionner tous les 'li' d'un 'ol'

`$("ul > li")` // sélectionner les enfants de l'élément 'ul'

`$("li#premier ~ li")` // sélectionner les frères 'li' du premier 'li' de la liste

Sélectionner des éléments d'une page Web

```
<ol>
```

```
  <li>Premier élément</li>
```

```
  <li>Deuxième élément</li>
```

```
  <ul>
```

```
    <li>Élément sans ordre</li>
```

```
  </ul>
```

```
</ol>
```

```
<ul>
```

```
  <li id="premier">Un</li>
```

```
  <li id="deuxième">Deux</li>
```

```
  <li id="troisième">Trois</li>
```

```
  <li id="quatrième">Quatre</li>
```

```
</ul>
```

Sélectionner des éléments d'une page Web

- Filtres: sur l'arborescence, de contenu , d'attribut, de visibilité

`$("p:first")` // sélectionner la première occurrence de l'élément paragraphe 'p'

`$("p:last")` // sélectionner la dernière occurrence de l'élément paragraphe 'p'

`<p>Premier texte</p>`

`<p>Deuxième texte</p>`

`<p>Troisième texte</p>`

Sélectionner des éléments d'une page Web

`$("p:contains('test')")` // sélectionner tous les éléments 'p' qui contiennent le texte 'test'

`$("li:has(ul)")` //sélectionner les éléments d'item de liste 'li' contenant au moins une liste

`$("[align!=center]")` // sélectionner tous les éléments qui ont l'attribut 'align' différent de 'center'

`$("p:visible")` //sélectionner les éléments 'p' visibles

Intégrer des évènements

```
$('#p').on( 'click', function () {  
    alert(« Vous avez cliqué sur un paragraphe!")  
});
```

// ou encore

```
$('#p').click(function () {  
    alert (« Vous avez cliqué sur un paragraphe!")  
});
```

Intégrer des évènements

Evènements 1 ^{ère} méthode	Evènements 2 ^{ème} méthode
<ul style="list-style-type: none"><code>.on('click', function() { ... })</code><code>.on('scroll', function() { ... })</code><code>.on('hover', function() { ... })</code><code>.on('mouseover', function() { ... })</code><code>.on('mouseout', function() { ... })</code><code>.on('mouseenter', function() { ... })</code><code>.on('mouseleave', function() { ... })</code><code>.on('keydown', function() { ... })</code><code>.on('keyup', function() { ... })</code><code>.on('keypress', function() { ... })</code><code>.on('focus', function() { ... })</code><code>.on('blur', function() { ... })</code><code>.on('resize', function() { ... })</code>	<ul style="list-style-type: none"><code>.click()</code><code>.scroll()</code><code>.hover()</code><code>.mouseover()</code><code>.mouseout()</code><code>.mouseenter()</code><code>.mouseleave()</code><code>.keydown()</code><code>.keyup()</code><code>.keypress()</code><code>.focus()</code><code>.blur()</code><code>.resize()</code>

Utiliser des méthodes JQuery

- ▶ Sélectionner un élément converti en un objet jQuery avec `$()`, auquel on applique des méthodes afin de
 - ▶ Générer des effets sur la page Web
 - ▶ Manipulation du contenu
 - ▶ Parcours du DOM
 - ▶ Modifier la taille et position des éléments

Exemple JQuery

```
<script type="text/javascript" src="//code.jquery.com/jquery-3.6.0.min.js"></script>
<script>

    $(document).ready(function(){
        $("p").on("click", "", clickParagraphe);
    });

    function clickParagraphe(){
        this.style.color="red";
    }

</script>
```

TPs: énoncés

▶ TP10:

- ▶ Afficher deux paragraphes
- ▶ Ajouter la bibliothèque jQuery

```
<script src="//code.jquery.com/jquery-3.6.0.min.js">
```

```
<script type="text/javascript" src="./js/jquery-min.js"></script>
```

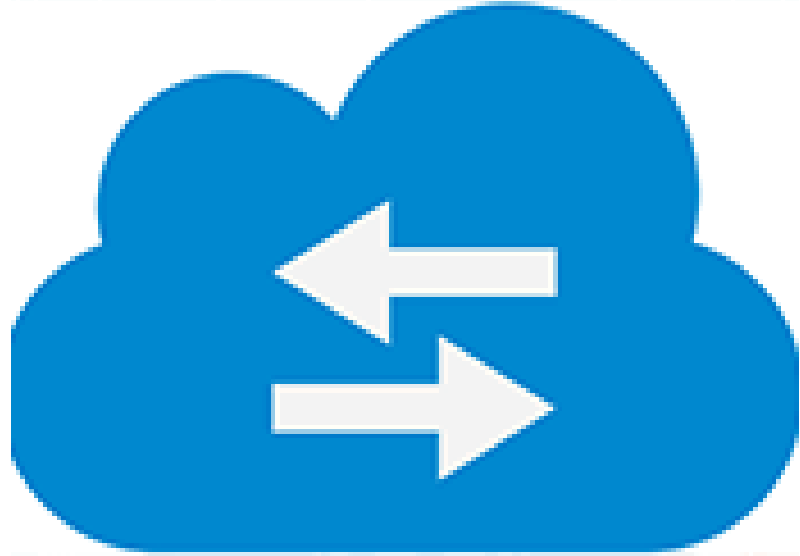
- ▶ Au chargement de la page, affecter aux deux paragraphes la fonction clickParagraphe associée au clic, afin de changer la couleur de l'élément cliqué en rouge.

Un paragraphe qui va changer de couleur...

Et un deuxième paragraphe.

Un paragraphe qui va changer de couleur...

Et un deuxième paragraphe.



Web API

API Web

API Web

- ▶ API (Application Programming Interface)
 - ▶ Interface qui permet à des applications de communiquer entre elles
 - ▶ Web API, API navigateur, API Serveur Web, ...
- ▶ APIs navigateur
 - ▶ Ensemble de fonctions développées par les navigateurs afin de permettre des opérations “complexes”
 - ▶ Historique
 - ▶ Géolocalisation
 - ▶ ...

API Web

- ▶ APIs tierces
 - ▶ Intégrer ses API avant de les utiliser
 - ▶ YouTube API
 - ▶ Twitter API
 - ▶ Facebook API

API Web

- ▶ API de l'Histoire
 - ▶ Méthodes afin de manipuler l'objet 'windows.history'
 - ▶ 'window.history' contient les URLs des sites Web visités par l'utilisateur

Propriétés/Méthodes	Rôle
back()	Charger l'URL précédente dans la liste de l'historique
forward()	Charger l'URL suivante dans la liste de l'historique
go()	Charger l'URL spécifiée de la liste de l'historique
length	Retourner le nombre d'URL dans la liste de l'historique

API Web

- ▶ API de Géolocalisation
 - ▶ Objet: Geolocation
 - ▶ Obtenir la localisation géographique de l'utilisateur, sous contrainte que l'utilisateur approuve que la page Web puisse l'obtenir

Propriétés	Rôle
coordinates	Retourner la position et l'altitude de l'utilisateur
position	Retourner la position de l'utilisateur à un instant donné
positionError	Retourner l'erreur relative à l'appel des méthodes de l'objet Geolocation

API Web

Méthode	Rôle
<code>clearWatch()</code>	Libérer les ressources de localisation/erreur créées antérieurement en utilisant <code>Geolocation.watchPosition()</code>
<code>getCurrentPosition()</code>	Retourner la position actuelle de l'utilisateur
<code>watchPosition()</code>	Retourner un ID et permettre d'enregistrer une fonction de rappel qui sera appelée à chaque fois que la position change, en renvoyant la position à jour

Graphiques

- ▶ Bibliothèque Javascript pour utiliser des graphiques
 - ▶ Plotly.js
 - ▶ Chart.js
 - ▶ Google Chart : [google.charts](https://google.com/charts)

Graphiques

► Fonctionnement

- Intégrer librairie Google graphics

```
<script type="text/javascript"  
    src="https://www.gstatic.com/charts/loader.js"></script>
```

- Charger la librairie

```
google.charts.load('current',{packages:['corechart']});  
google.charts.setOnLoadCallback(drawChart); // appeler la fonction  
                                             // 'drawChart' au chargement de l'API
```

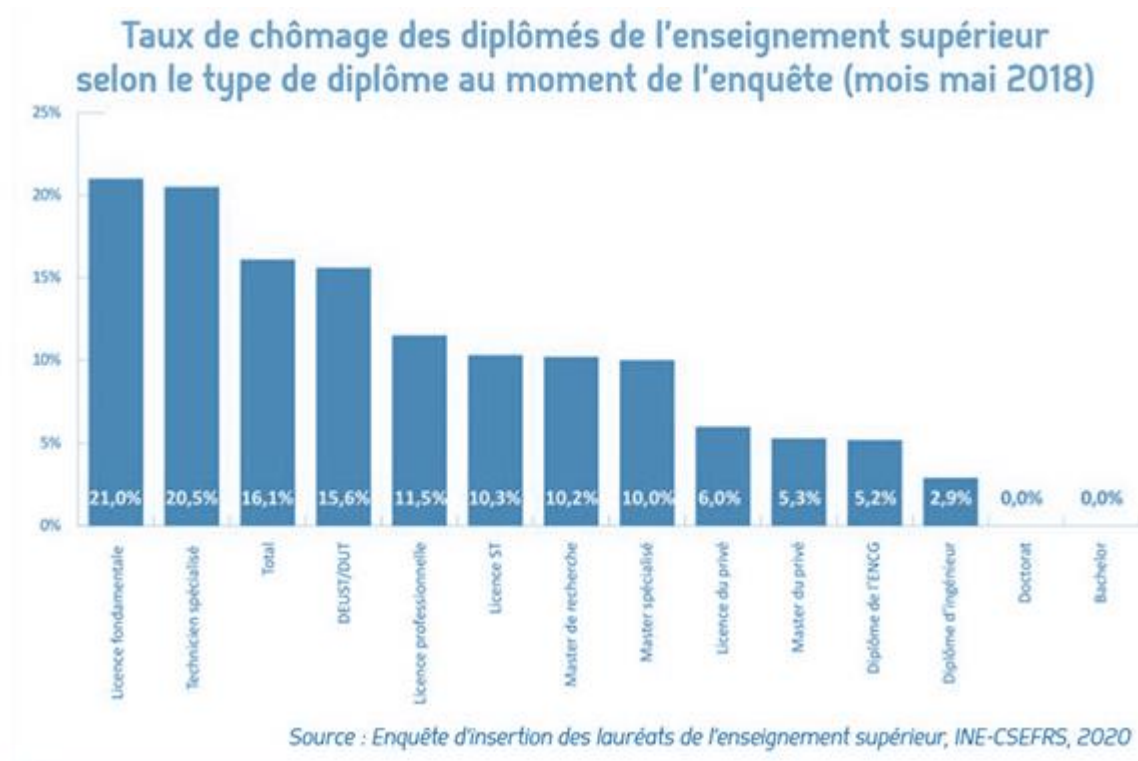
- Préparer les données (données statiques, à partir d'une base de données, fichiers csv, etc...)
- Personnaliser le graphique et le dessiner

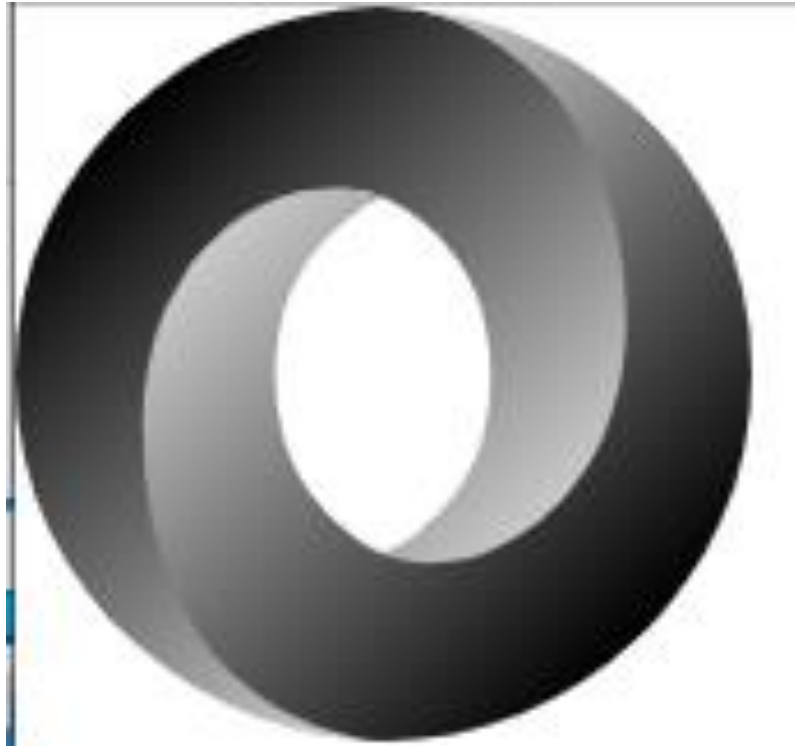
TPs Enoncés

- ▶ Reprendre le TP6 afin d'écrire les fonctions permettant d'avancer et de reculer dans l'historique du navigateur
- ▶ TP11: écrire un script qui permet d'obtenir la localisation de l'utilisateur
- ▶ TP12: intégrer Youtube dans votre page HTML
 - ▶ Ouvrir la vidéo YouTube à intégrer
 - ▶ Cliquer sur le bouton partager et choisir "intégrer"
 - ▶ Copier le code HTML de la vidéo
 - ▶ Intégrer le code dans la page HTML

TPs Enoncés

- ▶ TP13: écrire un script qui permet d'afficher le graphique suivant





JSON(JavaScript Object Notation)

Introduction

- ▶ JSON
 - ▶ JavaScript Object Notation
 - ▶ Format pour représenter et échanger les données
 - ▶ Utilisé pour envoyer les données à partir du serveur Web vers la page Web
 - ▶ JSON est indépendant des langages de programmation et de scripting, "self-describing" et facile à comprendre et à utiliser

Syntaxe JSON

- ▶ Syntaxe dérivée de la syntaxe Javascript pour les objets
 - ▶ Données : clé:Valeur
 - ▶ Données sont séparées par des virgules
 - ▶ Accolades pour définir des données de type object
 - ▶ Crochets pour définir des données de type array

Types de données JSON

- ▶ Valeurs doivent être du type :

- ▶ string
- ▶ number
- ▶ object (JSON object)
- ▶ array
- ▶ boolean
- ▶ Null

```
{"nom":"Paul"}
```

```
{"age":30}
```

```
{ "employee":{"name":"Paul", "age":30, "city":"Rabat"}}
```

```
{ "employees":["Paul", "Anna", "David"]} }
```

Traitement des objets JSON

- ▶ Données sous format JSON sont en général de type string
- ▶ Traiter ces données avec la fonction JSON.parse() afin de générer des données de type “JavaScript object”:

```
const obj = JSON.parse(' { "name":“Paul”, "age":30, "city":“Rabat”} ');
```

```
<p id="demo"></p>
```

```
<script> document.getElementById("demo").innerHTML = obj.name; //affiche Paul
```

```
</script>
```

Traitement des objets JSON

- ▶ On peut convertir un objet JavaScript en un objet JSON et l'envoyer au serveur: JSON.stringify()

```
const myObj = {name: "John", age: 31, city: "Rabat"};  
const myJSON = JSON.stringify(myObj);  
window.location = "demo_json.php?x=" + myJSON; // rediriger le  
navigateur vers une nouvelle page
```


Objets JSON

```
const myJSON = '{"name": "Paul", "age": 30, "city": "Rabat"}';  
const myObj = JSON.parse(myJSON);  
x = myObj.name;      // accéder à la valeur de la clé « name » de l'objet myObj
```

```
const myJSON = '{"name": "Paul", "age": 30, "city": "Rabat"}';  
const myObj = JSON.parse(myJSON);  
x = myObj["name"];    // accéder à la valeur de la clé « name » de l'objet myObj
```

Objets JSON

```
const myJSON = '{"name": "Paul", "age": 30, "city": "Rabat"}';  
const myObj = JSON.parse(myJSON);  
let text = "";  
for (const x in myObj) // parcourir les clés dans l'objet myObj  
{ text += x + ", "; }
```

```
const myJSON = '{"name": "Paul", "age": 30, "city": "Rabat"}';  
const myObj = JSON.parse(myJSON);  
let text = "";  
for (const x in myObj) // parcourir les valeurs des clés  
{ text += myObj[x] + ", "; }
```

Objets JSON

```
{  
  "name": "John",  
  "age": 30,  
  "cars": ["Ford", "BMW", "Fiat"]  
}  
myObj.cars[0];
```

TPs Enoncés

▶ TP14:

- ▶ Déclarer une chaîne de caractère

```
const text = '{"employees":[' + '{"firstName":"John","lastName":"Doe" },' +  
'"firstName":"Anna","lastName":"Smith" },' +  
'"firstName":"Peter","lastName":"Jones" ]}';
```

- ▶ Convertir la chaîne de caractères en objet JSON
- ▶ Afficher toutes les clés/valeurs de l'objet JSON

TPs Enoncés

▶ TP15:

- ▶ Afficher à partir d'un fichier externe les clés/valeurs de l'objet JSON
- ▶ Le fichier externe contient les données suivantes:

```
{  
  "name": "John",  
  "age": 31,  
  "pets": [  
    { "animal": "dog", "name": "Fido" },  
    { "animal": "cat", "name": "Felix" },  
    { "animal": "hamster", "name": "Lightning" }  
  ]  
}
```

Nb: puisque on va utiliser XMLHttpRequest(), on doit exécuter les pages Web sur un Serveur Web

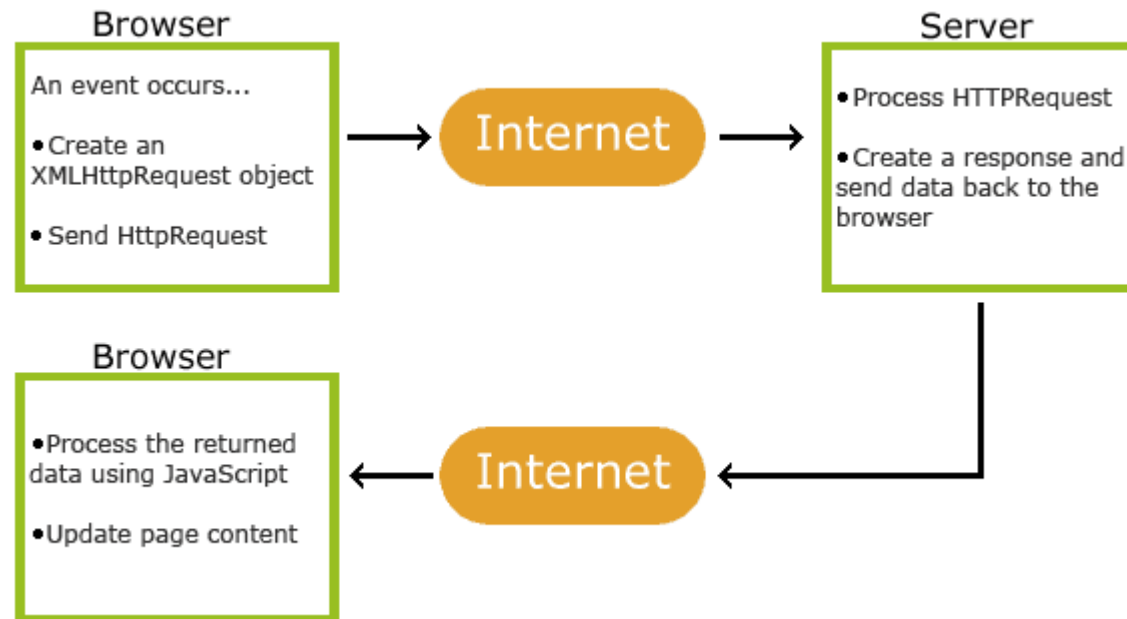


Frameworks Javascript: AJAX

Introduction

- ▶ AJAX = Asynchronous JavaScript And XML
- ▶ AJAX
 - ▶ N'est pas un langage de programmation
 - ▶ Utilise une combinaison
 - ▶ Objet prédéfini : XMLHttpRequest object ou Fetch API , afin de récupérer les données auprès du serveur
 - ▶ Code JavaScript et HTML DOM, afin de traiter les données et/ou les afficher

Fonctionnement AJAX



Fonctionnement AJAX

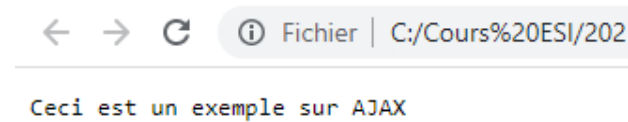
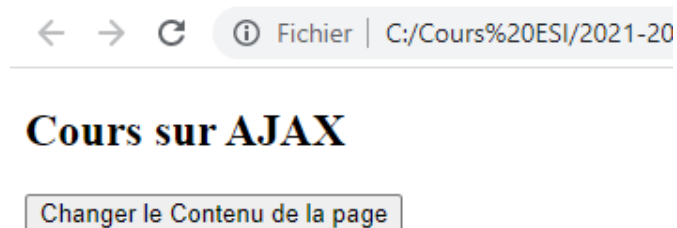
- ▶ Créer un objet XMLHttpRequest
- ▶ Définir une fonction callback qui contient le code à exécuter lorsque la réponse est obtenue à partir du serveur
- ▶ Obtenir l'objet XMLHttpRequest et envoyer une requête au serveur: méthodes open() et send()
 - ▶ Objet XMLHttpRequest est utilisé pour l'échange de données avec le serveur, ainsi certaines parties de la page Web peuvent être modifiées sans la recharger à partir du serveur

Fonction Callaback: fonction passée comme paramètre pour une autre fonction

TPs énoncés

▶ TP 16:

- ▶ Créer une page HTML qui contient une section div et un bouton
- ▶ En réponse au clic sur le bouton, une fonction affiche le contenu d'un fichier texte à partir du serveur



Bibliographie

- ▶ JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- ▶ Le Modèle Objet de Document (DOM: Document object Model): https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model/Introduction
- ▶ Document Object Model (DOM) Specification: <http://www.w3.org/TR/dom/>
- ▶ Le site officiel de jQuery: <https://jquery.com/>
- ▶ Introduction à jQuery: <https://openclassrooms.com/fr/courses/3504441-introduction-a-jquery>
- ▶ Le site officiel de Angular: <https://angularjs.org/>
- ▶ Introduction to Web Development: <https://www.coursera.org/learn/web-development>
- ▶ Introduction to Web Development with HTML, CSS, JavaScript: <https://www.coursera.org/learn/introduction-to-web-development-with-html-css-javascript>
- ▶ HTML Tutorial: <https://www.w3schools.com/html/default.asp>
- ▶ JavaScript Tutorial: <https://www.w3schools.com/js/default.asp>
- ▶ JSON Tutorial: https://www.w3schools.com/js/js_json_intro.asp