fichier site-specialites.zip sur site

1 Problématique

Dans un souci de rendre *la page de présentation des spécialités* plus attrayante pour les visiteurs, on se propose d'apporter de l'interactivité dans le contenu. En effet, cette *page statique* ne propose pour l'instant que des liens vers des pages web externes. Par exemple, il pourrait être intéressant d'afficher sur la même page, divers contenus en fonction des clics.

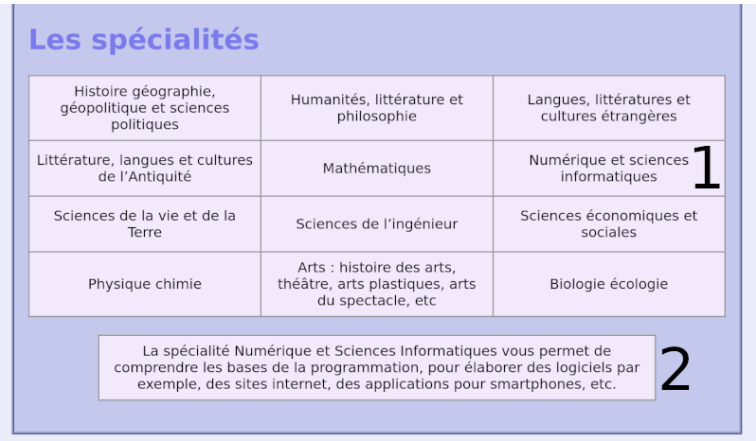


FIGURE 1 – En cliquant dans la cellule **1** la description de la spécialité apparaît dans la cellule **2**.

Comment modifier une page web de manière interactive ?

2 Requête http

2.1 Principe général

HTTP (HyperText Transfer Protocol) est un protocole élaboré par Tim Berners-Lee en même temps que le système d'adresse *URL (Uniform Ressource Locator)* et le langage *HTML (HyperText Markup Language)*, pour créer le *World Wide Web*.

À retenir

Ce protocole permet de récupérer des ressources telles que des documents HTML. Il est à la base de tout échange de données sur le Web. C'est un protocole de type *client-serveur* (figure 2), ce qui signifie que les requêtes sont initiées par le destinataire (qui est généralement un navigateur web).

Mozilla Developpement Network

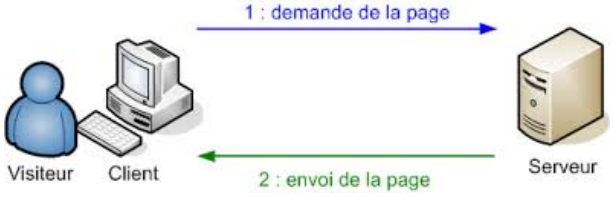


FIGURE 2 – Protocole HTTP

Christophe Viroulaud

2.2 Détails des requêtes

En premier approche nous pouvons considérer que la requête s'effectue en deux étapes :

— Requête

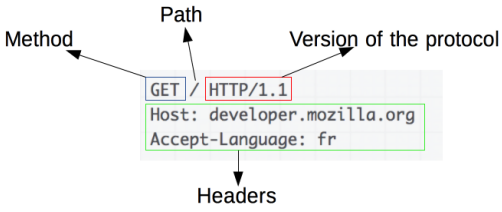


FIGURE 3 – Le client (le navigateur) demande la page web au serveur.

— Réponse

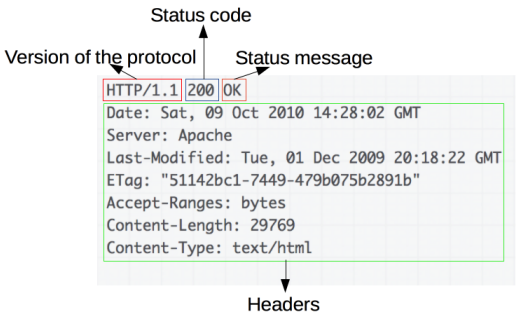
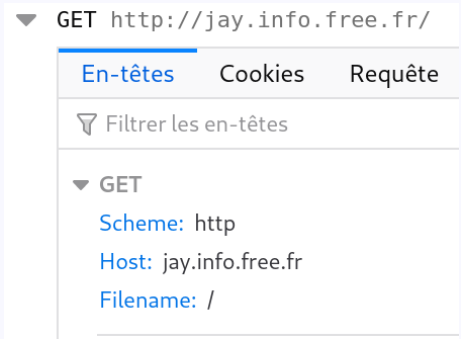


FIGURE 4 – Le serveur renvoie la page demandée.

Activité 1 : L'objectif est de retrouver les requêtes effectuées. Selon le navigateur utilisé la méthodologie est sensiblement différente

1. **Sur Firefox :**
   (a) Cliquer sur *Ctrl+Shift+E*, le panneau *réseau* s'ouvre.
   (b) Ouvrir la page `http://jay.info.free.fr`
   (c) Ouvrir la ligne *GET*.

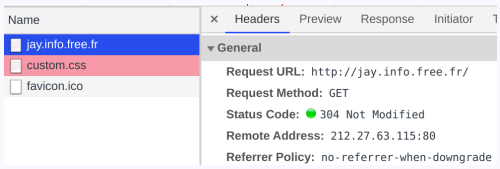


   (d) Dans *En-têtes*, retrouver alors les informations transmises et retournées.
2. **Sur Chrome :**
   (a) Cliquer sur *Ctrl+Shift+I*, un panneau s'ouvre.
   (b) Choisir *Réseau* ou *Network*.

Christophe Viroulaud

(c) Ouvrir la page http://jay.info.free.fr

(d) Cliquer sur *jay.info.free.fr*.



(e) Dans *Headers*, retrouver alors les informations transmises et retournées.

3. Quelle est la version du protocole *HTTP* utilisé ?
4. Quelle est la dernière date de modification de la page ?
5. Exécuter les mêmes opérations avec un site plus évolué : https://cviroulaud.github.io
6. Combien de requêtes *GET* le navigateur a-t-il effectué pour charger la page ?

À retenir

Une page web moderne contient de nombreux contenus stockés sur différents serveurs (texte, image, vidéo, bibliothèque...). Le navigateur effectue plusieurs requêtes pour récupérer ces données puis construire l'affichage de la page.

3 JavaScript

3.1 Contexte historique

C'est un langage de programmation de *scripts* c'est à dire qui permet de manipuler les fonctionnalités d'un système informatique. Il a été crée en mai 1995 par *Brendan Eich* pour le compte de la Netscape Communications Corporation.

3.2 Le DOM

Le **DOM (Document Object Model)** est une interface qui permet de manipuler le contenu des pages web.

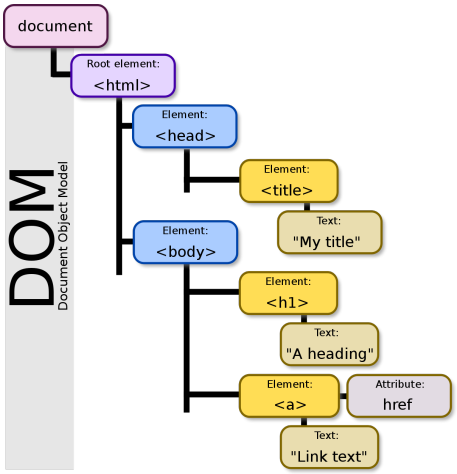


FIGURE 5 – Document Object Model

On le nomme *arborescence de nœuds (node tree)* (figure 5) car il peut être considéré comme un arbre qui se ramifie en plusieurs branches enfants, chacune pouvant avoir des feuilles.

— Le premier parent est l'élément *racine HTML*.

— Un élément peut posséder des attributs et/ou du texte.

```
1 <a href="https://cviroulaud.github.io">Super site</a>
```

Code 1 – Élément HTML

Dans le code 1 :

— *href* est un attribut,

— *Super site* est le texte affiché.

3.3 Interagir avec le DOM

Les éléments du DOM sont des objets atteignables grâce au JavaScript. Il existe plusieurs méthodes disponibles présentées en détails dans la documentation en ligne. Ces quelques références donnent un aperçu des possibilités :

— https://tinyurl.com/ybuaoyuy

— https://tinyurl.com/yxnyglbg

— et bien sûr le site de la MDN : https://developer.mozilla.org/fr/docs/Web/API

Concentrons-nous sur une méthode en particulier. L'élément *a* du code HTML 2 possède un attribut *id* (pour identifiant) et un attribut *href*.

```
1 <a id="monlien" href="http://monsite.fr">Un site web</a>
```

Code 2 – Code HTML

Le code JavaScript 3 permet d'accéder à l'élément. Les lignes précédées de // sont des commentaires et ne sont pas exécutées.

```
1 // Le mot clé 'let' permet de déclarer une variable (nommée lien dans
     cet exemple)
2 let lien = document.getElementById("monlien");
```

Code 3 – Référencer un élément du DOM

Remarque

Le JavaScript est sensible à la casse. Il respecte le *CamelCase*.
Il faut également remarquer le point-virgule en fin de chaque instruction.

Le code 4 modifie le contenu de l'élément *a* référencé dans la variable *lien*.
Le code 5 modifie le lien référencé par l'attribut *href*.

Christophe Viroulaud

```
1 lien.textContent = "Le meilleur site web";
```

Code 4 – Modifier le texte d'un élément

```
1 lien.setAttribute("href", "https://cviroulaud.github.io");
```

Code 5 – Modifier la valeur de l'attribut d'un élément

Activité 2 :

1. Créer une page HTML et placer l'élément *a* du code 2.
2. Juste avant la balise fermante </body>, placer le code 6.

```
1 <script src="main.js"></script>
2 </body>
```

Code 6 – Lier le code JavaScript au fichier HTML

3. Créer un fichier *main.js* dans le même dossier que le ficher HTML.
4. Copier les codes 3, 4 et 5 dans le fichier JavaScript.
5. Ouvrir le fichier HTML depuis un navigateur et vérifier que le lien a bien été modifié.
6. Depuis le navigateur, accéder au code source de la page.
7. Remarquer que le texte et les attributs du lien contiennent les valeurs codées dans le fichier HTML initial.

3.4 Les écouteurs

Grâce au JavaScript, il est possible de modifier le contenu de la page web lors de son chargement. L'étape suivante est de pouvoir interagir avec les actions de l'utilisateur.

À retenir

Un **écouteur d'événement** surveille les actions de l'utilisateur et exécute un code spécifique quand une de ces actions est réalisée.

Les codes ci-après permettent de surveiller le clic sur un tableau HTML.

```
1 <table id="table-specialites">
```

Code 7 – Création d'un tableau dans la page HTML

```
1 let tabSpecialites = document.getElementById("table-specialites");
```


Christophe Viroulaud

Code 8 – Référencement du tableau en JavaScript

```
tabSpecialites.addEventListener('click', function(event) {
  //code exécuté lors du clic sur le tableau
  //ici une fenêtre 'bonjour' s'ouvre
  alert("bonjour');
});
```

Code 9 – Création de l'écouteur sur le tableau

La documentation en ligne, notamment sur le site de la MDN https://tinyurl.com/n5zjs4u détaille davantage les spécificités des écouteurs.

Activité 3 :

1. Télécharger le fichier compressé *site-specialites.zip* sur le site https://cviroulaud.github.io
2. Décompresser les fichiers et ouvrir le site depuis un navigateur.
3. Cliquer sur une cellule du tableau et observer la modification réalisée.
4. Depuis un IDE, observer le code du fichier *main.js*.
5. Modifier les descriptions pour chaque spécialité en prenant modèle sur celle de la NSI.
6. Modifier le code pour qu'une image différente soit affichée pour chaque spécialité. Les images devront correspondre au thème de la spécialité.

3.5 Exécution côté client

Dans les exemple précédents, le code JavaScript est exécuté *côté client*, c'est à dire par le navigateur. Deux indicateurs nous permettent de l'affirmer :

— Il n'y a pas de rechargement de la page lors d'un clic sur une cellule.

— Le code source de la page HTML (accessible par un clic-droit) contient le code initial non modifié.

À retenir

Le JavaScript modifie dynamiquement la page web. Le fichier HTML source n'est pas réécrit. Un rechargement de la page supprime les modifications réalisées par le code JavaScript.

Christophe Viroulaud