Numérique et science informatique
Classe de première

Lycée Hoche

Année scolaire 2025-2026

Contents

1 Le Web-le protocole HTTP

Le Web consiste en un vaste ensemble de documents: les pages Web .
On y accède par le biais de l'Internet.

Chaque page est un document hypermedia (hyper pour lien activable renvoyant vers d'autres pages web et media car le contenu d'une page peut comporter des éléments hétérogènes:texte,images,vidéos)

Le Web fonctionne grâce aux serveurs et aux navigateurs.

Un navigateur est une application qui permet d'accéder à une page Web et de l'afficher.
Le navigateur contacte le serveur concerné pour obtenir une copie de la page Web demandée.

Une page Web contient donc un mélange de texte et d'autres éléments tels que les images,les vidéos.Ces éléments sont représentés en HTML (HyperText Markup Language).

Un fichier HTML contient des balises (tags).
Une balise est encadrée par " < > ".

1.1 Les URL

Chaque page Web est identifiée par un nom : son URL (**Uniform Resource Locator**).
Le nom commence par la description de la méthode d'accès , le protocole de transfert :

http://nom_de_l'hote[:numero_port]/chemin d'_acces :

La chaine de caractères *nom_de_l'hote* représente le nom du domaine (par ex www.education.gouv.fr) ou l'adresse IP de la machine sur laquelle s'exécute le serveur.

numero_port est un numéro de port optionnel (en fait on le spécifie lorsque le serveur n'utilise pas le port 80).

La chaîne de caractères *chemin d'_acces* identifie un document particulier sur le serveur.

1.2 Le protocole de transfert HTTP

Le protocole utilisé pour communiquer entre le navigateur et le serveur.

HTTP fonctionne comme une *application*.Il présuppose l'existence d'un protocole de transport fiable comme **TCP** , mais l'application ne garantit pas elle-même la fiabilité de la transmission.

Lorsque'une session de transport est établie entre le client et le serveur , le navigateur envoie une *requête* HTTP à laquelle le serveur répond.

Chaque requête est indépendante de la précédente (sans mémoire).

L'échange entre le navigateur et le serveur est bidirectionnel : Le serveur envoie une demande pour une page Web et le serveur en fournit une copie au navigateur.

Le navigateur peut envoyer des données au serveur par l'intermédiaire de **formulaires**.

HTTP permet au navigateur et au serveur de préciser certaines caractéristiques de l'échange comme la police de caractères utilisée lors du transfert .

Pour améliorer les temps de réponse , un navigateur utilise une mémoire cache où il stocke une copie des pages demandées.
Si l'utilisateur redemande une des pages , HTTP interroge le serveur pour connaître les modifications éventuelles depuis la dernière requête.

Pour résumer

Le navigateur prend un URL , en extrait la partie *nom_ de_ l'hote* et utilise le *DNS* pour convertir le nom de l'hôte en une adresse IP qui permet d'établir une connexion TCP avec le serveur.
Une fois la communication établie , le navigateur et le serveur communiquent à l'aide du protocole HTTP.

Pour demander une page Web le navigateur envoie la commande HTTP **GET**:

GET http://www.mathgreen.fr/page01.html .

Si la connexion est déjà établie , un simple GET page01.html suffit.

Les messages d'erreur:

Dans la plupart des cas , le navigateur affiche ce que lui rendra le serveur y compris des messages d'erreur (en cas où la page demandée n'existe pas par exemple).

Les autres méthodes disponibles sont:

HEAD ,POST ,PUT ,DELETE ,CONNECT ,OPTIONS ,TRACE ,PATCH.

PUT, DELETE et PATCH permettent de modifier des données sur le serveur: Ceci nécessite une authentification pour réaliser les changements et le serveur doit être configuré pour autoriser ces changements.

HEAD permet d'obtenir seulement l'en-tête de la réponse.

POST permet d'envoyer une ressource au serveur : cette méthode est souvent utilisée lors du remplissage d'un formulaire ,avec une balise *form*, voir les activités sur HTML5 et CSS3).

Les *codes de statut* commencent par 100,200,300,400 ou 500.

- 200 et suivants indiquent une réussite.
- 300 et suivants indiquent un déplacement de la ressource demandée.
- 400 et suivants indiquent une erreur du client: requête mal formulée ou ressource inexistante (erreur **404**).
- 500 et suivants indiquent une erreur du serveur.

HTTP a connu plusieurs évolutions:

- Version 1.0 : c'est la version qu'on a décrite ci-dessus.
- Version 1.1 : plusieurs requêtes peuvent se succéder au sein de la même connexion TCP/IP (keep-alive) mais si une requête n'aboutit pas , elle ralentit les suivantes.Une technique d'optimisation consiste à ouvrir plusieurs connexions TCP/IP en parallèle sur le même site mais les navigateurs modernes limitent le nombre de connexions parallèles à 6.
- Version 2.0 : plusieurs requêtes en parallèle gérées au sein de la même connexion TCP/IP.
  Possibilité de *push*: les données sont envoyées au client alors qu'il ne les as pas encore demandées.
  Compression des en-têtes.

(Voir la page ci-dessous pour des informations complémentaires:

`https://developer.mozilla.org/fr/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP`

Ces changements permettent d'accélérer le chargement des pages web qui ont une tendance à l'embonpoint.
De plus les données sont mises en cache dans le navigateur.Cela permet de ne pas les recharger auprès du serveur lorqu'on utilise les boutons *Précédent* et *Suivant* du navigateur.

1.3 Un exemple concret d'interaction

:
Demande d'une page web : eduscol.education.fr par un navigateur internet:

"GET":
"scheme":"https",
"host":"eduscol.education.fr",
"filename":"/",
"Adresse":"185.75.143.93:443"

Etat : 200 OK
Version: HTTP/2
Transfert: 23,75Ko

"**En-têtes de la requête** (584 o)":
"Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
"Accept-Encoding": "gzip, deflate, br",
"Accept-Language": "fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3",
"Cache-Control": "max-age=0",
"Connection": "keep-alive",
"Cookie": "tarteaucitron=!youtube_custom=false!vimeo_custom=false!dailymotion_custom=false!twitt
col.education.fr",
"TE": "Trailers",
"Upgrade-Insecure-Requests": "1",
"User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0"

Ici on utilise la méthode **GET** pour obtenir une ressource.
Celle-ci est la racine (/) du site web "eduscol.education.fr"
Le protocole est HTTP/2.
Le navigateur web est Mozilla (firefox).
Le langage préféré est le français (Accept Language).
Les ressources peuvent être compressées.(Accept encoding)

"**En-têtes de la réponse** (852 o)":
"accept-ranges": "bytes",
"cache-control": "max-age=0, must-revalidate, no-store",
"cache-control-origin": "max-age=900, public",
"cache-tags: "HIT",
"content-encoding": "gzip",
"content-language": "fr",
"content-length": "23469",
"content-type": "text/html; charset=UTF-8",
"date", "Thu, 25 Feb 2021 13:11:47 GMT",
"etag", "W/ï1614251501ï¨",
"expires": :"Sun, 19 Nov 1978 05:00:00 GMT",
"last-modified": "Thu, 25 Feb 2021 11:11:41 GMT",
"link", "<https://eduscol.education.fr/>; rel=shortlink; <https://eduscol.education.fr/>; rel=canonical",
"server": "nginx",
"strict-transport-security": "max-age=31536000; includeSubdomains",
"vary": "Accept-Encoding",
"x-cache": "HIT",
"x-cache-age": "540",
"x-cache-hits": "123",
"x-cache-ttl": "900.000","x-cacheable": "YES",
"x-content-type-options": "nosniff",
"x-drupal-dynamic-cache": "MISS",
"X-Firefox-Spdy": "h2",
"x-frame-options": "SAMEORIGIN",
"x-ua-compatible": "IE=edge"

Le navigateur analyse la page et , pour chaque lien présent, réalise toutes les étapes précédentes avant d'afficher le résultat.Une page web moderne peut nécessiter près de 100 requêtes.

2 Les formulaires

2.1 Constructioin

Sur de nombreuses pages web, des formulaires sont utilisés que ce soit pour une authentification par identifiant et mot de passe ou pour une recherche d'informations. Le formulaire est un élément codé en HTML

```
1 <form method="get" action="action.php">
2 <label for="ident">Votre identifiant</label>
3 <input type="text" name="identifiant" id="ident"/>
4 <input type="submit" value="Envoyer" />
5 </form>
```

Un formulaire se trouve contenu dans <form ..... /form>.

Dans la balise <form>, on ajoute la méthode HTTP utilisée (ici GET) et le lien qui sera activé lors de l'envoi du formulaire (action.php).

Ensuite, grâce à <input ... />, on crée deux objets :

- un champ de texte (ligne 3)
- un bouton pour valider le formulaire (ligne 4).

Le champ de texte est précédé d'un label référencé par `ident`. Cette référence est utilisée dans `input` avec `id` afin qu'un clic sur "Votre identifiant" place le curseur dans le champ de texte correspondant.

`identifiant` est une "variable" qui recevra le texte entré dans le champ et qui sera transmise à action.php.

2.2 GET ou POST

Lors de l'utilisation de la méthode GET dans un formulaire, les données sont transmises en clair dans la barre d'adresse du navigateur.

Dans les faits, on lit dans l'URL action.php?identifiant=<texte entré>.

Cela pose des problèmes de sécurité.

En effet, si on remplace le type "text" par le type"password" alors lors de l'entrée du mot de passe, les lettres sont remplacées par des points. Mais lors de l'envoi des données, le mot de passe apparaît en toutes lettres dans la barre d'adresse !

Comme remède (partiel) on utilise la méthode POST qui transmet les données dans le corps de la requête : les données sont donc moins visibles.

2.3 Gestion de données

Pour gérer les données provenant du formulaire, on utilise un script php action.php. Le code le plus simple possible est le suivant :

```
1 <html>
2 <body>
3 <?php
4 echo $_POST['identifiant'];
5  ?>
6 </body>
7 </html>
```

On suppose que la méthode utilisée dans l'envoi du formulaire est POST. Grâce à PHP, on peut alors récupérer le texte et l'afficher. PHP est un langage de script exécuté au niveau du serveur. Celui-ci doit évidemment accepter de gérer le PHP.

3 Sécurité

HTTP transmet le texte brut.
Pour que les données soient transmises de manière sécurisée, on utilise HTTPS.Lorsque la communication avec un site web est sécurisée, la barre d'adresse commence par un cadenas.

HTTPS est une association de HTTP et de SSL.
SSL(*Secure Socket Layer*) vient en complément de TCP/IP.Ses avantages sont:

- La confidentialité : on ne peut pas espionner les données.La connexion est chiffrée.
- L'intégrité des données : on ne peut pas les modifier.
- L'authentification : On est sûr de l'identité du client et du serveur grâce à des certificats présents dans le navigateur, fournis par des sociétés tierces à qui on fait implicitement confiance.

L'évolution récente de SSL est TLS (Tranport Layer Security) qui accompagne la version 2 de HTTP.

De plus en plus de sites web basculent en HTTPS sous la pression des navigateurs internet.En effet,au lieu d'indiquer simplement qu'un site en HTTPS est sécurisé avec le fameux cadenas, les navigateurs indiquent que les sites HTTP ne sont pas sécurisés.

Il faut bien se rappeler que **seule la communication est sécurisée**:
rien ne nous assure que les données confidentielles fournies au site sont-elles mêmes protégées sur le serveur.

4 *Références*

Le descriptif de HTTP fourni par Mozilla : `https://developer.mozilla.org/fr/docs/Web/HTTP`.
Les formulaires en HTML : `https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires`.

5 Eléments de JAVASCRIPT - Evènements

5.1 Introduction

HTML (Hypertext Markup Language) permet de créer la structure d'une page web.
Grâce à un jeu de balises, il permet de décomposer la page comme un traitement de texte : titre,sous-titre,section..

On peut par exemple indiquer que l'on utilise un titre avec les balises ouvrante et fermante <h1> </h1> ou indiquer l'utilisation d'une énumération avec <ol> pour chaque item.

CSS(Cascade Style Sheet) permet la mise en forme des différents éléments HTML.On peut ainsi modifier la couleur ou la police de caractères des éléments de la page.
On peut dynamiser la page web de deux manières:

- **Côté serveur**: avec le langage PHP (Hypertext Preprocessor) qui peut, par exemple, ajouter le résultat d'une requête à une base de données dans la page qui sera fournie au navigateur.
- **Côté client** : avec JavaScript qui peut , par exemple, faire apparaître des infos-bulles contextuelles ou réaliser des animations.

5.2 Présentation de JavaScript

JavaScript est un langage créé en 1995 par Brendan Eich qui travailla it chez Netscape Communication Corporation , principal concurrent à l'époque d'internet explorer.JavaScript est alors transmis à l'**ECMA** (European Computer Manufacturers Association) pour standardisation sous la dénomination d'*ECMAScript* abrégé en *ES*.

Les deux versions 5 , sortie en 2009, et 6 , sortie en 2014, sont implémentées dans les différents navigateurs présents sur le marché.Les versions suivantes 7,8 et 9 sont des mises-à-jour "mineures" de l'ES6 et les fonctionnalités ajoutées ne sont pas supportées par tous les navigateurs.

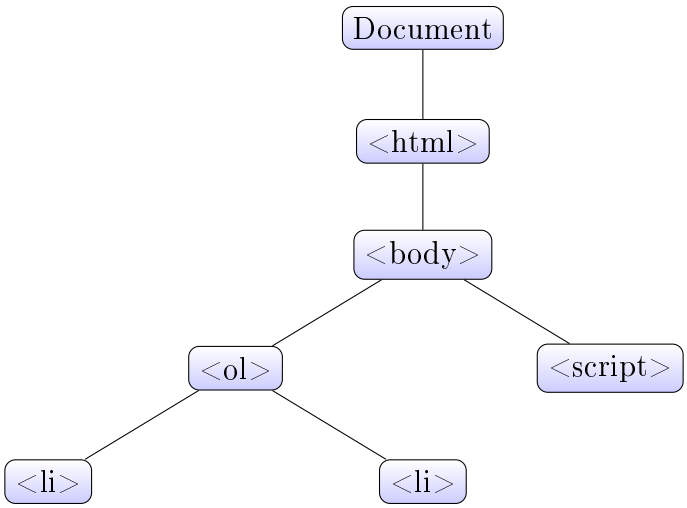
Javascript dont le nom est une référence à **Java** de Sun Microsystems,ne doit pas être confondu avec java.En effet, Java est fortement typé et précompilé alors que Javascript est à typage dynamique et interprété.(Voir l'annexe 2 , à la fin de ce cours , sur les différences entre un langage compilé et un langage interprété).

Notons pour finir cette introduction : **JavaScript est un langage orienté objet**.
Annexe 1:

Le DOM

Lors de la création du rendu de la page HTML, le DOM(Document Object Model) est créé.
Il représente la structure de la page HTML sous la forme d'un arbre ramifié.Chaque niveau est un noeud et le niveau hiéarchique le plus faible est le texte.

Cela permet de comprendre la manière dont on accède,en JavaScript, au texte associé à chaque élément de la liste en utilisant la propriété en lecture seule *firstChild* qui renvoie le premier noeud de l'objet puis la propriété *nodeValue* qui permet de lire ou d'écrire la valeur de ce premier noeud.
Lors de l'appel de la fonction , le paramètre transmis utilise le mot clé *this* qui permet de faire référence à l'élément à partir duquel l'appel a été réalisé.



Annexe 2

1. Les langages compilés sont traduits en langage machine (assembleur) que le processeur peut exécuter. Ils sont plus rapides et plus efficaces à exécuter que les langages interprétés. Ils permettent également au développeur de mieux contrôler les aspects matériels, comme la gestion de la mémoire et l'utilisation du processeur (CPU).

   Les langages compilés doivent d'abord être compilés manuellement. Vous devez "reconstruire" le programme à chaque fois que vous devez apporter une modification. Dans

   Des langages tels que C, C++, Erlang, Haskell, Rust et Go sont des langages compilés.

2. Dans un langage interprété, le code source n'est pas directement traduit par la machine cible. Au lieu de cela, un programme différent, appelé l'interpréteur, lit et exécute le code.

   Les langages interprétés étaient autrefois nettement plus lents que les langages compilés. Mais, avec le développement de la compilation just-in-time (juste-à-temps), cet écart se réduit.

   Des langages tels que PHP, Ruby, Python et JavaScript sont des langages interprétés.