

Interactions entre l'homme et la machine sur le WEB

Activité II : communication client/serveur



Principe de communication, DOM, HTTP

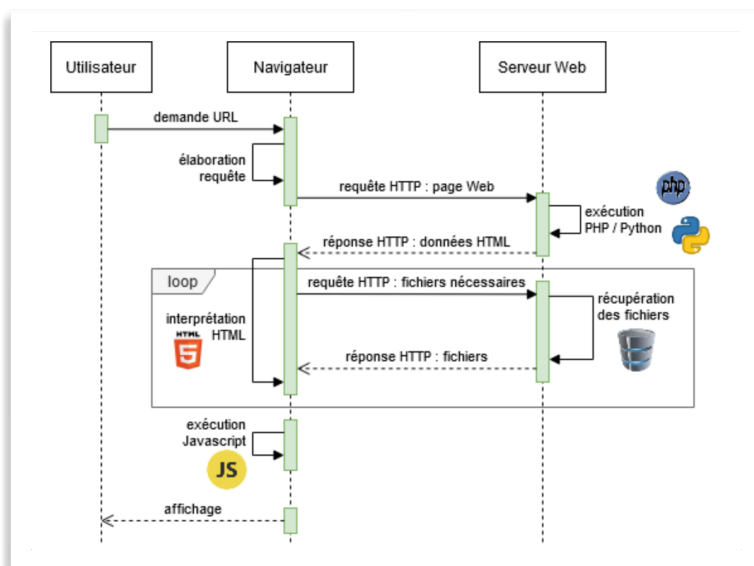
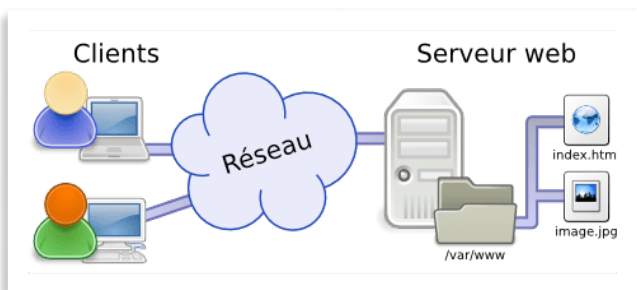
Présentation

De manière simpliste, un serveur est une machine informatisée quelconque fournissant un ou plusieurs services à des consommateurs. Un client est alors une tierce machine informatisée quelconque pouvant consommer ces services.

les Objectifs :

- Rappeler les principes du web
- Visualiser le DOM et les échanges http
- Modifier une page web

Principe de communication : client/serveur



Décrire le rôle du navigateur :

Décrire le rôle du serveur web :

Qu'envoie le clients au serveur ?

Qu'envoie le serveur au client ?

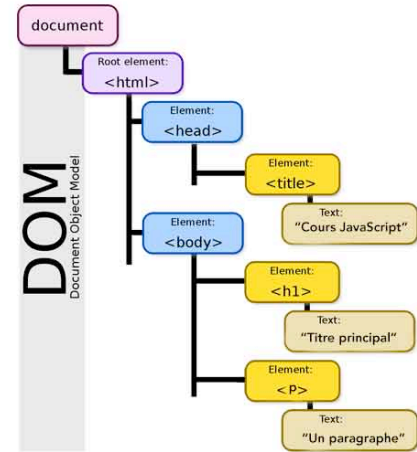
Qu'est-ce que le protocole HTTP :

HTML et DOM

Code HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours JavaScript</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Titre principal</h1>
    <p>Un paragraphe</p>
  </body>
</html>
```

Arbre DOM



Rappeler ce qu'est le HTML, une page web et un site web.

Rappeler la structure de la page HTML minimale

Comparer les deux représentations : code HTML et Arbre DOM (ressemblances, différences).

"Si vous deviez chercher 'votre chemin' au travers des deux représentations (code HTML, DOM), laquelle choisiriez-vous ? Pourquoi ?

Qui crée le DOM ?

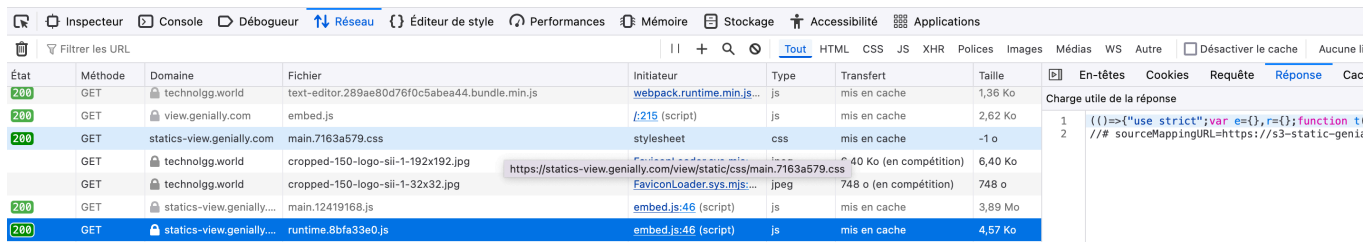
Comment pourrait-on modifier dynamiquement le contenu d'une page web

Quel(s) langage(s) de programmation connaissez-vous permettant de modifier dynamiquement le contenu d'une page web après son chargement ?

Que permet-il de manipuler ?

Les requêtes et réponses HTTP

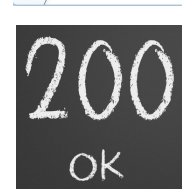
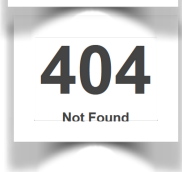
Ouvrir l'outil de développement d'un navigateur web



État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	En-têtes	Cookies	Requête	Réponse	Cac
200	GET	technolgg.world	text-editor.289ae80d76f0c5abea44.bundle.min.js	webpack.runtime.min.js...	js	mis en cache	1,36 Ko					
200	GET	view.genially.com	embed.js	/:215 (script)	js	mis en cache	2,62 Ko					
200	GET	statics-view.genially.com	main.7163a579.css	stylesheet	css	mis en cache	-1 o					
	GET	technolgg.world	cropped-150-logo-sli-1-192x192.jpg	https://statics-view.genially.com/view/static/css/main.7163a579.css		40 Ko (en compétition)	6,40 Ko					
	GET	technolgg.world	cropped-150-logo-sli-1-32x32.jpg	FaviconLoader.sys.mjs...	jpeg	748 o (en compétition)	748 o					
200	GET	statics-view.genially....	main.12419168.js	embed.js:46 (script)	js	mis en cache	3,89 Mo					
200	GET	statics-view.genially....	runtime.8bfa33e0.js	embed.js:46 (script)	js	mis en cache	4,57 Ko					

Visualiser le DOM de la page d'accueil du site de NSI (<https://technolgg.world>)

A quoi correspondent les codes de réponse :



Quelle est la méthode utilisée pour envoyer une requête ?

Combien comptez-vous d'images ?

combien de requête HTTP ?

Combien de feuilles de style ?

Que contient l'en-tête de la requête ?

Que contient l'en-tête de la réponse ?

Regarder la colonne "initiateur" et la ligne /:244(script) : Que contient la réponse ?

Quelle est la méthode utilisée pour envoyer une requête ?

Ouvrez le fichier "Post et Get.html".

Remplissez que la première partie du formulaire et envoyez avec le formulaire avec la méthode GET.

Quelle remarque pouvons nous faire sur l'URL de la page ?

Quelle est la conséquence ?

Test des méthodes GET et POST

Nom :

Email :

Mot de passe :

Nom :

Email :

Mot de passe :

Fermer et ouvrir de nouveau le fichier "Post et Get.html".

Remplissez que la deuxième partie du formulaire et envoyez avec le formulaire avec la méthode POST.

Quelle remarque pouvons nous faire sur l'URL de la page ?

Quelle conclusion pouvons nous faire sur les 2 méthodes ?