

HTML, CSS et JavaScript : les langages du Web

Séminaire Crans du lundi 5 décembre 2022
Préparé et présenté par stersky

Sommaire

- Un peu d'histoire sur le Web
- Comment ça marche une page Web
- Le HTML : la structure d'une page
- Le CSS : habiller son site Web
- JavaScript : et la page prend vie !

Introduction

Le Web : définition

Pour l'élève ou l'enseignant, la navigation constitue le premier pas dans le World Wide Web (WWW, Web ou toile). Il s'agit du contexte de communication le plus connu par le grand public au point d'être confondu avec l'Internet.

D'autres services de messagerie comme le courrier électronique ou le forum de discussion par exemple existent également sur l'Internet

Techniquement parlant, il s'agit d'un protocole d'échange hypertexte d'information (HTTP) utilisé sur l'Internet. Le trait dominant de la toile est donc le lien hypertexte qui lie tout document « Web » au réseau.

Source : <https://eduscol.education.fr/chrgrt/guidejuriscol.pdf> ; page 21

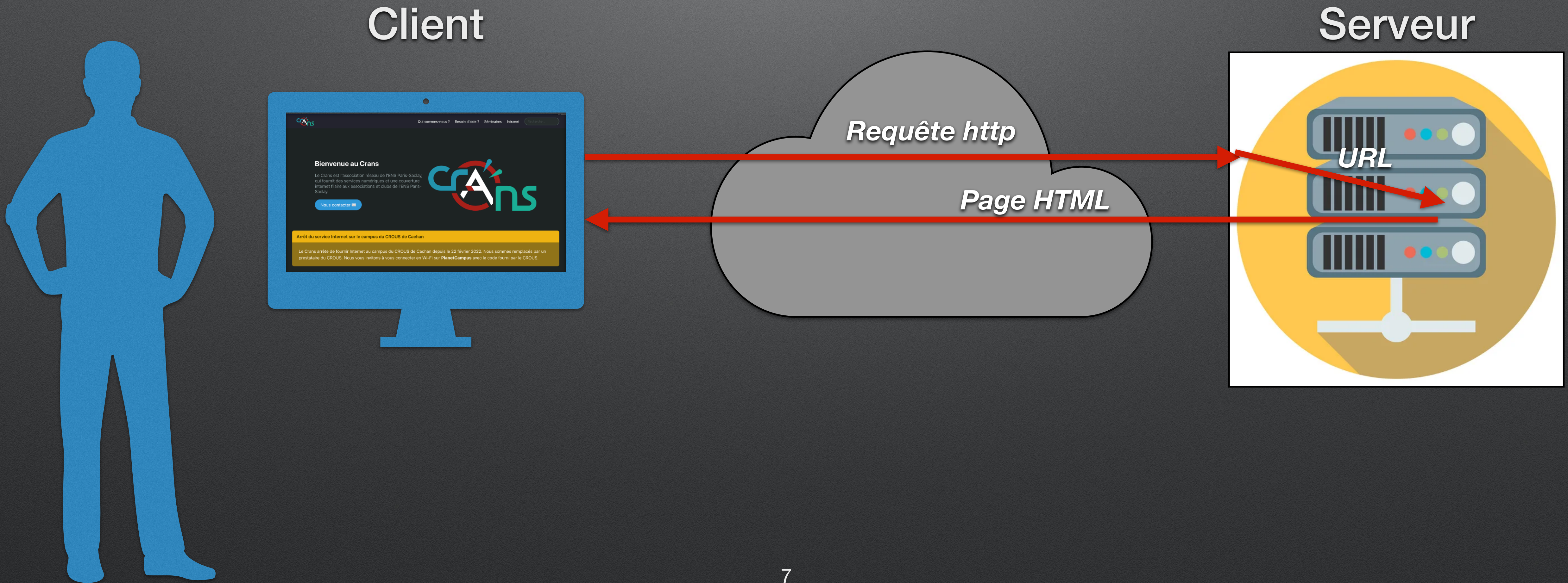
Une chronologie du Web

- 1965 : invention et programmation du concept d'hypertexte par Ted Nelson
- 1989 : naissance du Web au CERN par Tim Berners Lee
- 1993 : mise dans le domaine public, disponibilité du premier navigateur Mosaic
- 1995 : mise à disposition de technologies pour le développement de site Web interactif (langage JavaScript) et dynamique (langage PHP)
- 2001 : standardisation des pages grâce au DOM (Document Object Model)
- 2010 : mise à disposition de technologies pour le développement d'applications sur mobiles








Comment ça marche une page Web ?

Les principes derrière une page Web



Les trois “piliers du Web”

- **HTTP (HyperText Transfer Protocol)** le protocole de communication : permet la communication entre le client et le serveur
- **URL (Uniform Resource Locator)** l’adresse : permet de préciser l’emplacement de la ressource au sein du serveur
- **HTML (HyperText Markup Language)** la structure : ce qui est reçu par le client, définit la structure de la page qui sera affichée

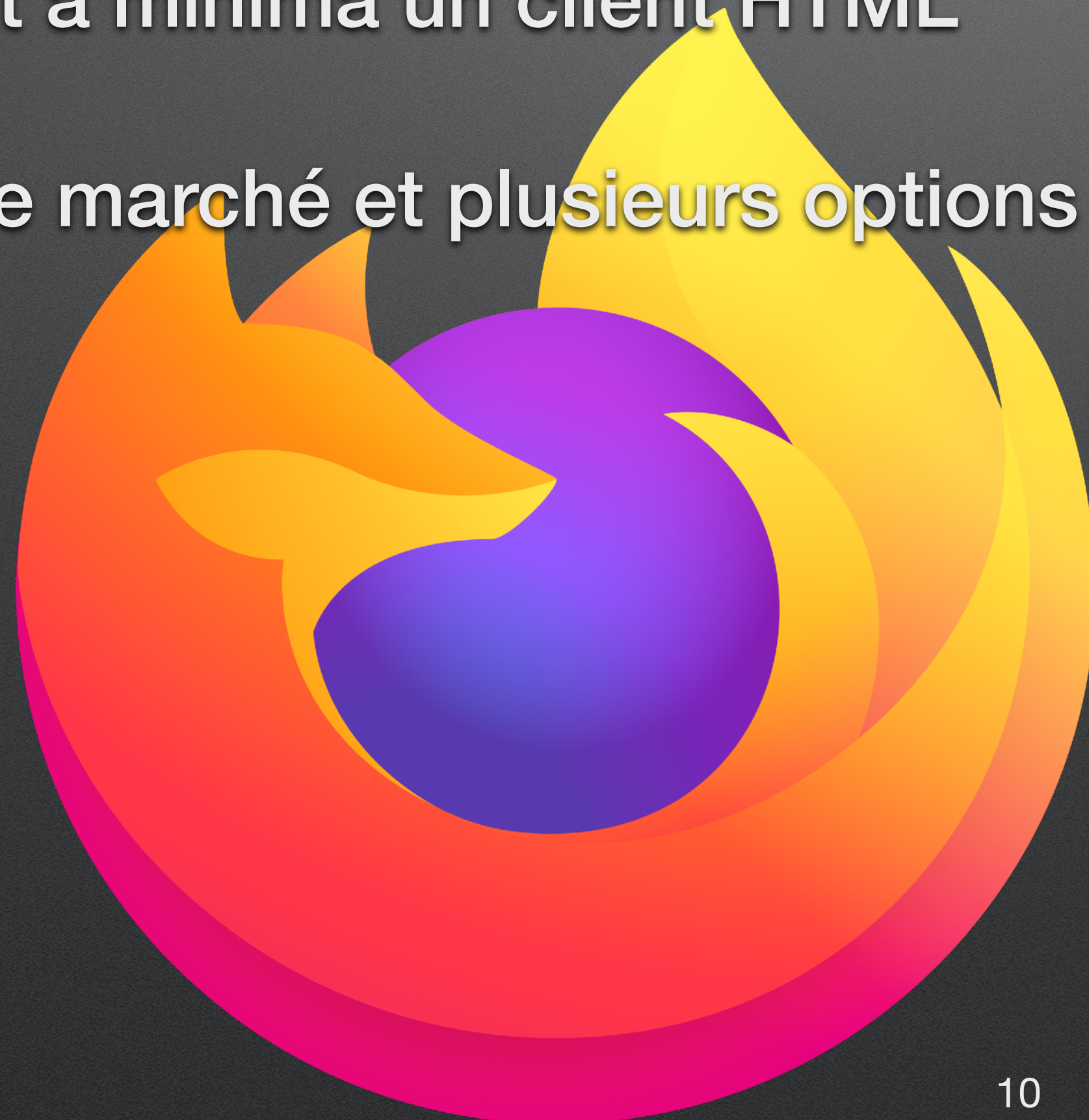
   <https://www.crans.org/about/index.html>  

Le principe derrière une page HTML

- ☞ L'idée : le serveur n'exécute pas de code (en statique)
- ☞ C'est le client (i.e l'ordinateur qui se connecte) qui exécute tout
- ☞ Permet d'alléger la charge du serveur
- ☞ Nécessite un logiciel spécialisé sur l'ordinateur client : un navigateur Web

Le navigateur Web

- Un logiciel permettant d'accéder au Web
- Contient à minima un client HTML
- Un vaste marché et plusieurs options



Construisons notre page HTML

L'élément de base du HTML : les balises

- Une page HTML est construite à l'aide d'éléments particuliers : les balises
- Définissent les différentes sections de la page
- Attribuent des rôles particuliers
- Une balise est un élément entre chevrons <>



Un exemple de structure

```
<!DOCTYPE HTML>
<html>
  <body>
    <h1>Salutations !</h1>
    <p>Je m'appelle Etienne Stransky</p>
  </body>
</html>
```



IL Y A UNE HIÉRARCHIE ENTRE LES ÉLÉMENTS EN HTML : IL Y A DES CONTENANTS ET DES CONTENUS

À vous de jouer !

Créez votre première page HTML !

1. Ouvrez un éditeur de texte
2. Écrivez-y un texte similaire à celui ci-contre
3. Enregistrez ce document au format .html
4. Ouvrez le document avec votre navigateur Web préféré
5. Admirez la beauté du résultat

```
<!DOCTYPE HTML>
<html>
  <body>
    <h1>Salutations !</h1>
    <p>Je m'appelle Etienne Stransky</p>
  </body>
</html>
```

Bonus : uploader sa page sur Zamok pour en faire un site public !

PRÉ-REQUIS : AVOIR UN COMPTE CRANS ACTIF

1. Se connecter à Zamok : `ssh pseudo@zamok.crans.org`
2. Créer un dossier www : `mkdir www`
3. Se déconnecter : `exit`
4. Aller dans le dossier qui contient le fichier .html : `cd ~/truc/bidule/machin`
5. L'envoyer sur Zamok : `scp fichier.html pseudo@zamok.crans.org:www/fichier.html`
6. Se connecter à sa page en entrant l'adresse suivante dans le navigateur :
perso.crans.org/psuedo/fichier.html

Quelques balises intéressantes

- `<html>` : définit le document html
- `<head>` : permet d'ajouter des en-tête
- `<title>` : définit le titre du document tel qu'il sera affiché
- `<body>` : crée le corps
- `<h1>` ; `<h2>` ; `<h3>` ; ... : crée des titres et des sections pour le document
- `<p>` : crée un nouveau paragraphe
- `
` : crée un saut de ligne
- `` : met du texte en gras
- `<!-- -->` : ne sera pas affiché -> permet des commentaires

Les attributs : personnalisons la page

- Un attribut est une information supplémentaire sur une balise HTML
- Il permet de spécifier plusieurs informations complémentaires
- Utilisation : dans une balise

`<p hidden> Ce texte ne sera pas affiché </p>`

Des attributs très utiles

- L'attribut *href*
 - Sert à spécifier un lien externe vers un élément
 - L'élément `<a>` est associé à un lien, spécifié grâce à `href`
- L'attribut *src*
 - Précise la source permettant d'ajouter l'élément
 - Utile sur les balises `` ; `<audio>` et `<video>`
- L'attribut *controls*
 - Ajoute une barre d'outils
 - Utile pour `<video>` et `<audio>`
- L'attribut *alt*
 - Précise un texte alternatif sur une image si elle n'est pas affichée
 - Utile pour ``
 - Améliore l'accessibilité d'un site pour les personnes qui auraient du mal à voir l'image

 **Cf. Séminaire de la semaine dernière !**

Un attribut un peu spécial

L'attribut style

- L'attribut style permet de spécifier l'apparence d'un élément

- Cela inclut :

- Sa taille
- Sa couleur
- Son centrage
- La police d'écriture
- ...

```
style = "  
height: 80px;  
color: red;  
text-align:center;  
font-family:courier;  
"
```

- Exemple :

```
<p style="font-family:Comic Sans MS; color:pink; font-size:80px">Bonjour</p>
```

De l'attributs styles au fichier CSS

Les limites de la personnalisation

- Si on veut faire une belle page Web, il faut personnaliser la plupart des éléments
 - Pour chaque balise, le texte décrivant le style peut être assez long
 - Cela peut résulter en un texte important => la page deviendrait très grosse
- Beaucoup d'éléments sont similaires : on aimerait les regrouper pour appliquer le même style
 - Cela économiserait du temps et des données

La solution: créer une feuille de styles



Les attributs class et id

- Permettent de regrouper des éléments censés avoir les mêmes styles
- Permet d'effectuer un tri entre éléments
- Ont les mêmes fonctionnalités, mais sont complémentaires
- Exemple :

```
<p class="corps_texte" id="presentation">
```

L'ananas (*Ananas comosus*) est une espèce de plantes xérophytes, originaire d'Amérique du Sud, plus spécifiquement du Paraguay, du nord-est de l'Argentine et sud du Brésil. Il est connu principalement pour son fruit comestible, qui est en réalité un fruit composé.

```
</p>
```

Comment intégrer une feuille de styles

- La feuille de style s'ajoute dans la tête de la page

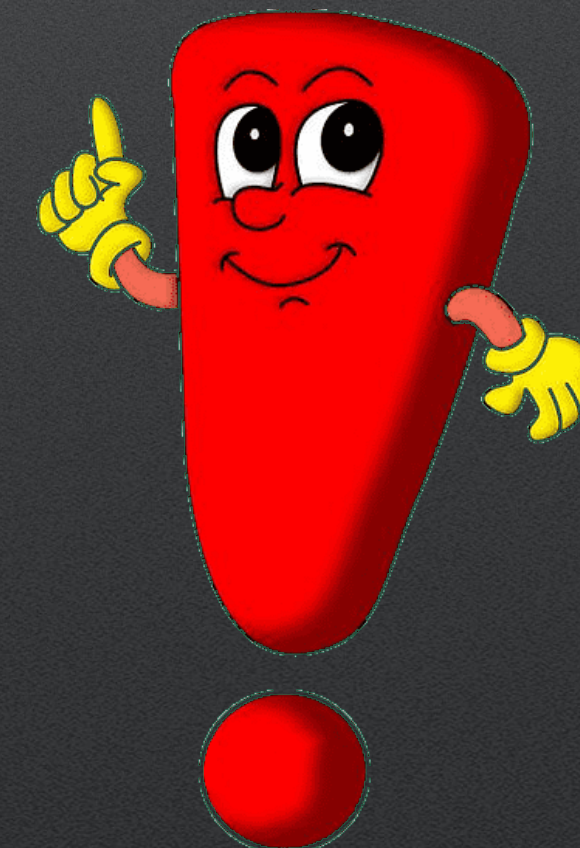
- Deux solutions :

- L'ajouter directement dans le fichier .html

```
<head>  
  <style>  
    //Notre contenu CSS  
  </style>  
</head>
```

- Ajouter une référence à un fichier extérieur en .css

```
<head>  
  <link rel="stylesheet" href="chemin/fichier.css">  
</head>
```



Si vous uploadez votre page sur un site, pensez à inclure les fichiers externes en faisant attention au chemin..

Le contenu du fichier CSS

- Une feuille de styles CSS contient tous les éléments de l'attribut "style" d'une balise HTML
- Pas besoin de préciser "style"
- On précise toutes les caractéristiques des objets concernés
- On peut affecter les éléments par type / classe / Id

Élément

```
p{  
  color:green;  
  font-family: fantasy;  
  text-align:center;  
}
```

Classe

```
.paragraphes{  
  font-size:15px;  
  font-style: italic;  
}
```

Id

```
#introduction{  
  border: 2px solid blue;  
  Text-align:left;  
}
```


À noter

- Il y a une hiérarchie dans les attributions de style incompatibles
 - Plus le style concerne un élément spécifique, plus il est prioritaire
 - Attribut style > id > classe > css pour tous les éléments
- Il y a énormément de personnalisations possibles pour le CSS
 - Exemples: effets de fenêtres, changements en cas de passage de la souris
 - Avec assez de patience, on peut créer la plupart des sites
- Il est possible d'avoir des feuilles de style personnalisées selon l'appareil utilisé (plus de compatibilité)
- Des différences peuvent exister en fonction des navigateurs, notamment Firefox

Le JavaScript : et la page prend vie !

JavaScript : comment et pourquoi ?



- JavaScript est un langage de programmation intégré aux sites Web
- Il s'exécute sur la machine client : ce n'est pas le serveur qui fait les calculs
- Il permet d'exécuter de nombreuses fonctions et de réaliser des interfaces avec différents éléments de la page
- La syntaxe est basée sur le C mais avec beaucoup de libertés permises
- C'est un langage très simple à exécuter : quasiment toutes les machines peuvent exécuter un code JavaScript



JavaScript n'est PAS pareil
que Java !

Ajouter un script à sa page : l'élément `<script>`

- On l'ajoute dans le corps de la page
- Contient du code qui s'exécute au moment du chargement de la page
- Sa position peut avoir une importance (pour vérifier que les éléments ont bien été créés...)
- Comme pour les fichiers CSS, on peut :
 - Intégrer le code à la page
 - Faire appel à un fichier externe

```
<body>  
  <script>  
    alert("Bonjour");  
  </script>  
</body>
```

```
<body>  
  <script scr="code.js">  
</body>
```

Les interactions JavaScript/Reste de la page

- JavaScript prend tout son sens s'il interagit avec la page HTML
- Des fonctions JavaScript peuvent être déclenchées selon l'état de la page
- Un code JavaScript peut modifier la page HTML et **changer l'affichage**
- Très nombreuses utilisations possibles, notamment pour créer une interface graphique simplement

Un script simple : déclencher une alerte

- On utilise l'élément "bouton" pré-conçu
- Un exemple de code :

```
<body>
  <button onclick="alerte_bouton()">Je suis un bouton</button>

  <script>
    function alerte_bouton(){
      alert("Vous avez cliqué !");
    }
  </script>
</body>
```

Une autre possibilité

```
<body>
  <button id="super_bouton">Bouton</button>
  <script>
    function fonction_droite(){
      alert("Bonjour ! Vous avez cliqué sur le bouton de droite !");
    }
    bouton_droite = document.getElementById("bouton_droite");
    bouton_droite.addEventListener("mousedown", fonction_droite, false);
  </script>
</body>
```


Bilan

- Le Web permet d'accéder à différents sites
- Un site statique se contente de renvoyer les informations sur une page à l'utilisateur, qui interprète via son navigateur
- Les sites sont construits à l'aide de balises HTML
- Les feuilles CSS permettent de personnaliser les pages
- JavaScript permet de rendre les pages plus dynamiques

Des sites utiles pour se lancer dans l'édition de pages Web

- W3schools : <https://www.w3schools.com/html/default.asp>
- W3cschools : <https://w3cschool.com/introduction-to-html>
- Mozilla : <https://developer.mozilla.org/fr/>
- Wikibooks : https://fr.wikibooks.org/wiki/Le_langage_HTML
- StackOverflow (toujours...) : <https://stackoverflow.com>

+ Votre moteur de recherche favori !

Merci pour votre attention !!!

Des questions ?