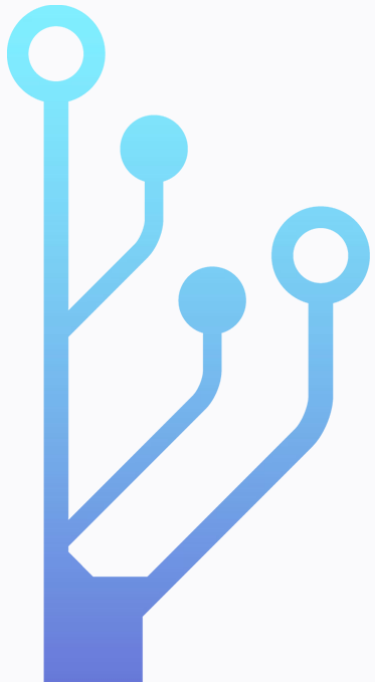


Formation Dev 1

Découverte du développement front-end

by Tanguy Boisset



Sommaire

- 1 - Comment fonctionne un site web?
- 2 - HTML
- 3 - CSS
- 4 - Javascript



DISCLAIMER !

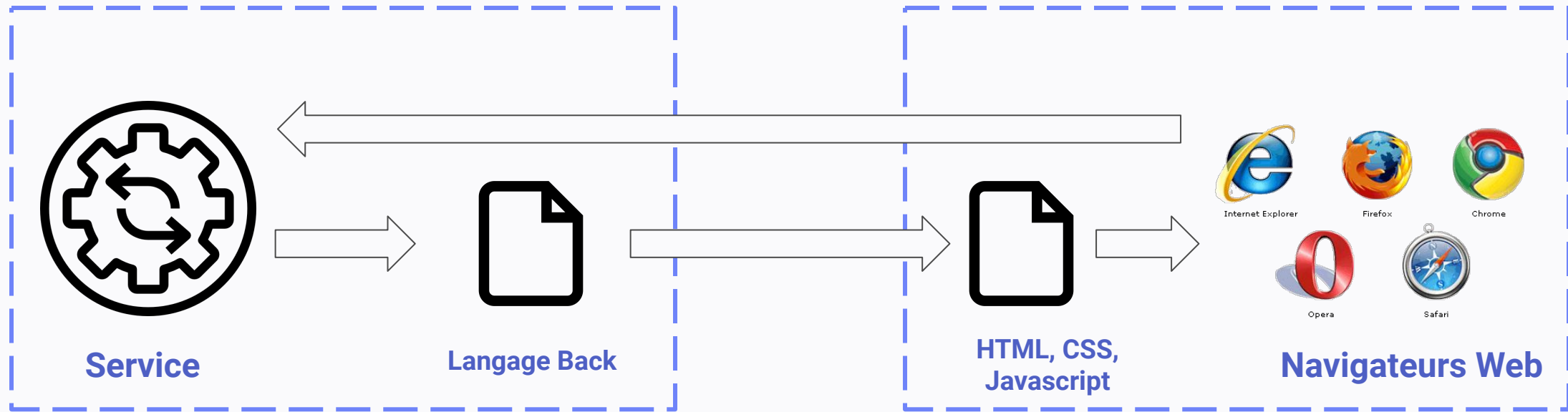
Cette “formation”, qui est plutôt une **initiation**, ne prétend pas vous apprendre le dev web en 1h30.

ViaRézo et moi-même ne pourrions être tenus responsables de la création de vos sites moches.

Toi après ce cours :



1 - Comment fonctionne un site web ?



Serveur



Client

2 - HTML



What you need :



What you get :



Un site (très) moche mais fonctionnel



2 - HTML : introduction

L'HTML est un langage de balises qui permet de gérer le **contenu** d'une page web.

Chaque balise permet de gérer un type de contenu : un titre, un paragraphe, une image...

Syntaxe : `<balise> ... </balise>` **ou** `<balise/>`

2 - HTML : structure

`<head>`

...

`</head>`

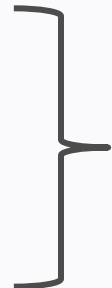


Le head (invisible) contient diverses infos sur la page web : le titre, le favicon, des liens vers d'autres fichiers...

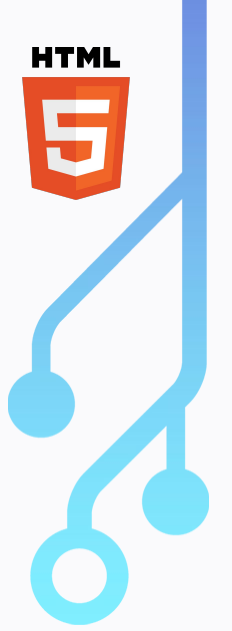
`<body>`

...

`</body>`



Le body (visible) contient l'intégralité du contenu à afficher sur la page : textes, images, vidéos...



2 - HTML : exemples

Quelques exemples :

<code><title> ... </title></code>	titre de la page	<code> ... </code>	lien
<code><link/></code>	lien vers un fichier externe	<code></code>	image
<code><script> ... </script></code>	ajout de script	<code> ... </code>	début/fin de liste
<code><p> ... </p></code>	paragraphe	<code> ... </code>	début/fin de liste ordonnée
<code><h1> ... </h1></code>	titre (d'importance 1)	<code> ... </code>	élément de liste
<code><h6> ... </h6></code>	titre (d'importance 6)	...	...

2 - HTML : attributs

Certaines balises nécessitent des paramètres supplémentaires pour fonctionner correctement :

Exemple : ``, `<a href="..."> ... </a>`

Ces paramètres s'appellent des **attributs**.

Mais certains attributs peuvent être placés sur n'importe quelle balise pour les personnaliser.

<code>style="..."</code>	personnalise le style	<code>title="..."</code>	donne un nom au survol
<code>alt="..."</code>	texte si l'élément est cassé	<code>href="..."</code>	relie à une URL
<code>class="..."</code>	donne une classe	<code>id="..."</code>	donne un id (unique)

2 - HTML : mise en évidence de problèmes

Problème 1 :

Ceci est un paragraphe → *Ceci est un paragraphe*

```
<p style="font-family: 'Comic Sans MS'; color: red; text-decoration-line: underline; text-decoration-color: red; font-weight: bold; font-style: italic;">Ceci est un paragraphe </p>
```

⇒ **Très lourd, rend le code illisible**

Problème 2 :

```
<p style="..."> paragraphe 1 blablabla </p>
...
<p style="..."> paragraphe 2 blablabla </p>
```

⇒ **Il faut recopier le style de chaque élément...**

3 - CSS



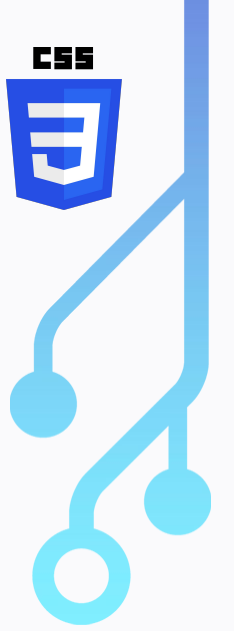
What you need :



.classes & #id

What you get :

Un site plutôt joli mais peu interactif





3 - CSS : structure

Relier le fichier HTML à la feuille de style CSS :

Ajouter dans le head du fichier HTML :

```
<link rel="stylesheet" href="nom_fichier.css" />
```

Syntaxe :

élément {

attribut : *valeur* ;

}

3 - CSS : un premier exemple

Ceci est un paragraphe



Ceci est un paragraphe

p {

font-family : Comic Sans MS';

color: red;

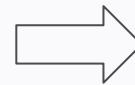
text-decoration-line: underline;

text-decoration-color: red;

font-weight: bold;

font-style: italic;

}



Ceci affectera tous les paragraphes (balises p)

Comment personnaliser un seul élément ?



3 - CSS : utilisation de classes et id

```
<p>Hello</p>
```

```
<p class="superClasse">Salut</p>
```

```
<p id="idDeGenie">Guten Tag</p>
```

balise	classes (multiple)	id (unique)
p { color: red; }	.superClasse { color: blue; }	#idDeGenie { color: green; }

Ordre de style : **balise** → **classe** → **id**

3 - CSS : exemples

color : rgb(100,30,180)	couleur de l'élément	width : 33%	largeur de l'élément
text-align : center	aligne l'élément	font-size : 1.5em	taille du texte
margin : 5%	règle la marge externe	background-color: blue	couleur de l'arrière-plan
padding : 3px	règle la marge interne	...	

Vous pouvez découvrir l'intégralité des attributs et leurs valeurs possibles sur : <https://www.w3schools.com/css/default.asp>

4 - JAVASCRIPT

What you need :

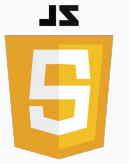


a strong will to live

What you get :



Un site dynamique et interactif
:0

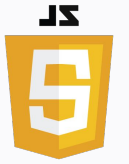


4 - JAVASCRIPT : introduction

Javascript est un langage de programmation, particulièrement apprécié en développement web, que ce soit pour le front ou le back.

Sa syntaxe est relativement simple et intuitive.





4 - JAVASCRIPT : syntaxe

Javascript n'est pas aussi "permissif" que Python, voici quelques différences :

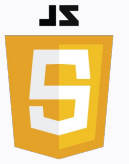
Déclaration des variables : chaque variable doit être préalablement déclarée via **let** ou **var** (ou **const**), mais pas son type ! Le type de la variable peut donc être changé.

Exemple : `var ma_variable = 20;`

Fin d'action : chaque "ligne" (ou presque) doit se terminer par ;.

Exemple : `ma_variable = "ceci est du texte";`

Ouverture/Fermeture de boucle : en JS, l'indentation n'a aucun effet autre qu'ergonomique : il faut ouvrir et fermer les boucles grâce à { ... }.



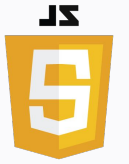
4 - JAVASCRIPT : deux exos

Relier le fichier HTML à la feuille de script JS :

Ajouter à la fin du fichier HTML :

```
<script src="nom_fichier.js"></script>
```

On va faire ensemble deux exos rapides pour se familiariser avec le langage !



4 - JAVASCRIPT : exo 1 - un bouton actif

Énoncé : Créer un bouton qui, lorsque cliqué, fait passer la page en mode nocturne (background noir et texte blanc).

Aide :

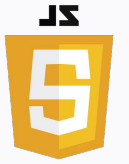
Code d'un bouton : `<button onclick="fonction_quand_clic()">CLICK ME</button>`

Syntaxe d'une fonction : `function nom_de_la_fonction(var 1,var2...) { code }`

Changer le style graphique d'une balise :

`document.nom_balise.style.nom_attribut = "...";`

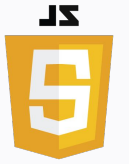
! BONUS ! : transformer le bouton en toggle (alternant mode jour et mode nuit)



4 - JAVASCRIPT : exo 1 - correction

Énoncé initial :

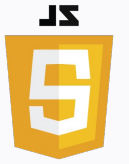
<u>HTML</u>	<u>JS</u>
<pre><button onclick="dark mode()"> DARK MODE </button></pre>	<pre>function dark mode() { document.body.style.color = "white"; document.body.style.backgroundColor = "black"; }</pre>



4 - JAVASCRIPT : exo 1 - correction

Énoncé bonus :

<u>HTML</u>	<u>JS</u>
<pre><button onclick="dark mode()"> DARK MODE </button></pre>	<pre>var dark = false; function dark mode() { if (dark) { document.body.style.color = "black"; document.body.style.backgroundColor = "white"; dark = false; } else { document.body.style.color = "white"; document.body.style.backgroundColor = "black"; dark = true; } }</pre>



4 - JAVASCRIPT : exo 2 - un bon accueil

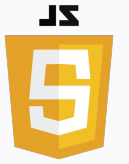
Énoncé : Lorsque le visiteur arrive sur le site, faire apparaître une pop up demandant son nom, puis l'accueillir de manière personnalisée.

Aide :

Pop up à remplir : `prompt("message", "valeur par défaut");`

Récupérer un élément grâce à son id : `document.getElementById("nom_id")`

Changer le contenu HTML d'un élément : `Element.innerHTML = "contenu"`



4 - JAVASCRIPT : exo 2 - correction

Énoncé initial :

<u>HTML</u>	<u>JS</u>
<pre><h1 id="accueil">Bonjour visiteur !</h1></pre>	<pre>var nom = prompt("Bonjour visiteur, veillez rentrez votre nom"); document.getElementById("accueil").inner HTML = "Bienvenue sur mon site " + nom + " !";</pre>

Ouverture

Vous savez maintenant réaliser de beaux sites interactifs (ou pas) !

Mais il y a encore quelques problèmes...

Allez sur votre site, clique droit sur un élément et “**inspecter l’élément**”

L’intégralité du front est accessible à TOUS les visiteurs !!!

ON NE STOCKERA AUCUNE DONNÉE SENSIBLE SUR LE FRONT

Comment alors stocker, traiter, échanger des données avec l’utilisateur ?



Remerciements

Merci à l'équipe ViaRézo pour leurs conseils

Si vous avez des questions, besoin d'aide sur un projet... :

tanguy.boisset@student-cs.fr

ou si ViaRézo vous intéresse :

venez nous parler ou nous voir au local !

Merci à vous !

