

Interactions entre l'homme et la machine sur le Web

J. Boucher

Lycée Pierre-Paul RIQUET, Première NSI

4 janvier 2025

Plan

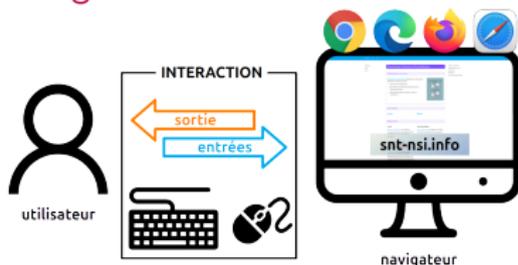
- 1 I. L'interface Homme-Machine
 - 1. Rendu d'une page Web
 - 2. Modification dynamique du rendu
 - 3. Gestion d'événements

1. Rendu d'une page Web

- Le **World Wide Web**, littéralement « toile d'étendue mondiale », est un ensemble de documents reliés entre eux par des *liens hypertexte*, les **pages Web**.



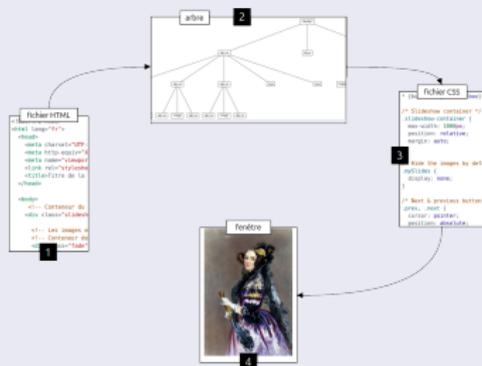
- Un utilisateur peut interagir avec une page Web, elle-même rendue dans la fenêtre graphique d'un **navigateur Web**.

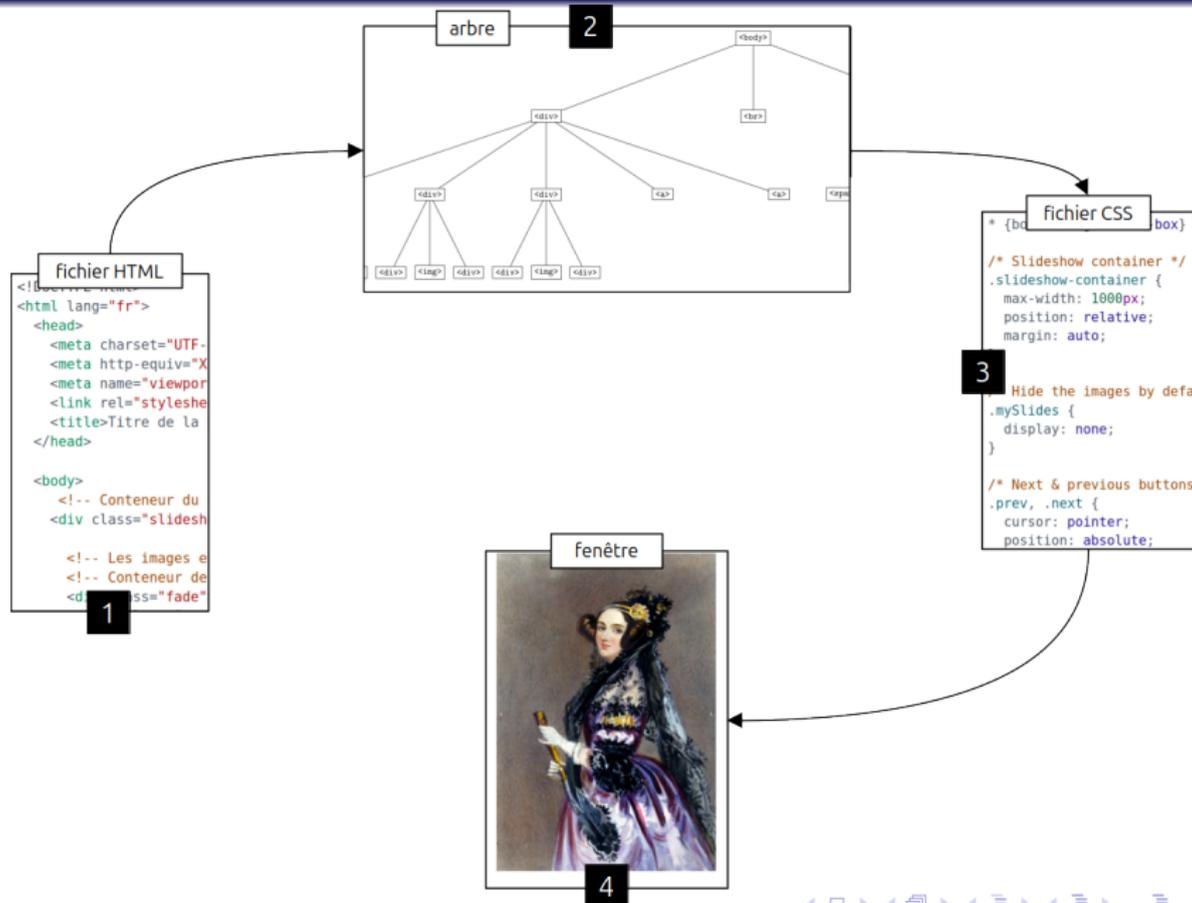


Processus de rendu d'une page Web dite *statique*

Pour rendre une page Web, le **navigateur** :

- 1 télécharge le fichier **HTML** correspondant à l'URL demandée ;
- 2 transforme la page en structure de donnée d'**arbre** ;
- 3 parcourt les nœuds de l'arbre construit en appliquant les règles **CSS** ;
- 4 génère le rendu graphique de la page qui est affiché dans la **fenêtre**.

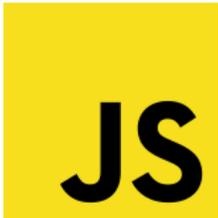




2. Modification dynamique du rendu

Le langage Javascript

- **JavaScript** est un langage de programmation utilisé principalement dans les pages Web interactives mais aussi pour écrire des applications exécutées par les serveurs Web.
- Javascript a été créé en 1995 par **Brendan Eich** pour le navigateur Netscape (qui deviendra ensuite Firefox).
- Tout comme Python, c'est un **langage interprété** qui supporte le paradigme impératif (affectation, boucles, instruction conditionnelle...), fonctionnel et objet.



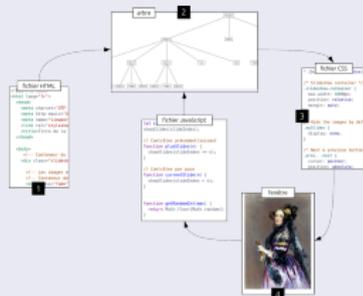
Processus de rendu d'une page Web dite *dynamique*

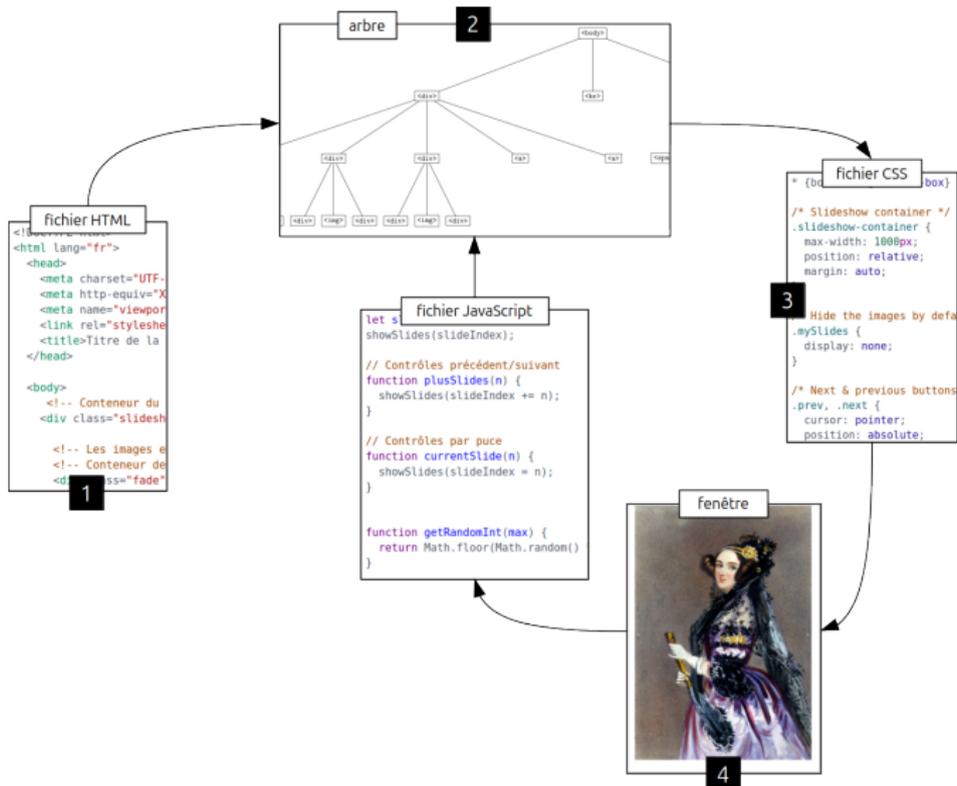
- Après la transformation du code HTML en une structure de données d'arbre, le code **Javascript est exécuté**.
- Il permet de **modifier l'arbre**, en y ajoutant ou en supprimant des noeuds, correspondant à des balises dans le document HTML, ou bien en modifiant leur attribut (class, style, src...).
- Tous ces changements sont **immédiatement** répercutés sur l'affichage.

```
function showSlides(n) {
  let i;
  let slides = document.getElementsByClassName("mySlides");

  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = slides.length}

  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slides[slideIndex-1].style.display = "block";
}
```





3. Gestion d'événements

- Le JavaScript permet d'intégrer la programmation événementielle aux pages Web, les rendant ainsi interactive.
- Le principe est de regrouper dans des **fonctions** les différentes modifications que l'on souhaite faire en réponse à un **événement**.

Exemple

Par exemple, dans le code suivant, la fonction `plusSlides` est exécutée en réponse à l'événement `onclick`, déclenché par un clic de l'utilisateur.

- code HTML :

```
<a class="prev" onclick="plusSlides(-1)">
```

- code Javascript :

```
function plusSlides(n) {  
    showSlides(slideIndex += n);  
}
```

- Un **événement** peut être associé à n'importe quel élément HTML (un bouton `<button>`, un paragraphe `<p>`, un titre `<h.>`, un bloc `<div>` ...), en l'ajoutant comme **attribut** de cet élément.
- Il existe beaucoup d'événements (plus d'une centaine), parmi les plus importants on trouve :

Événement	Description
<code>click</code>	Cliquer sur l'élément
<code>dblclick</code>	Double-cliquer sur l'élément
<code>mouseover</code>	Faire entrer le curseur sur l'élément
<code>mouseout</code>	Faire sortir le curseur de l'élément
<code>keydown</code>	Appuyer (sans relâcher) sur une touche de clavier sur l'élément
<code>keyup</code>	Relâcher une touche du clavier
<code>change</code>	Changer la valeur d'un champ de formulaire