

17/11/2022

Version 4

HTML5

JÉRÉMY PERROUULT



HTML

L'HISTOIRE

L'HISTOIRE DE HTML

- Internet, petit réseau informatique mondial
 - S'appuie sur le modèle réseau TCP/IP (IPv4 et IPv6), dérivé de ARPANET
 - Chaque connexion est identifiée par une adresse IP (ici IPv4)
 - Adresses IP privées (192.168.0.0, 172.16.0.0, 10.0.0.0 et 127.0.0.1)
 - Adresses publiques
 - Chaque connexion s'établit via un protocole de routage (transport) IP, comme TCP ou UDP
 - Chaque protocole de routage est utilisé par un protocole d'application (HTTP, FTP, SSH, HTTPS, MySQL, ...)
 - Chaque protocole applicatif est lié à un port d'écoute (80, 21, 22, 443, 3306, ...)
- Internet englobe tous types d'échanges protocolaires applicatif (HTTP, FTP, SSH, HTTPS, ...)
 - C'est le « réseau des réseaux »
- Un des réseaux les plus importants : World Wide Web (WWW)
 - Géré par le World Wide Web Consortium (**W3C**)

L'HISTOIRE DE HTML

- Le World Wide Web (ou Web) est un réseau de pages web
 - Ressources *hypertextes*, des ressources qui peuvent relier d'autres ressources (liens web)
 - Utilise le protocole HTTP (HyperText Transfert Protocol)
- Chaque ressource est accessible via une URL (Uniform Resource Locator)
 - Elle-même précisant
 - Le protocole utilisé
 - L'adresse IP ou le nom d'hôte
 - Le port
 - Le chemin vers la ressource
 - Exemples
 - `http://www.ma-page-internet.fr/ma-ressource.html`
 - `http://192.168.1.78:8080/index.html`
 - `mysql://localhost:3306/base`

L'HISTOIRE DE HTML

- Première ressource échangée écrite en HTML
 - HyperText Markup Language
- Texte interprété par les navigateurs
- Ressource (fichier) .html (index.html)
- De nos jours, elle peut être générée par un serveur Web (JEE, PHP, ASP, NodeJS, ...)

L'HISTOIRE DE HTML

- HTML est standardisé par SGML (Standard Generalized Markup Language)
 - MAIS ...
- XHTML est standardisé par XML (eXtensible Markup Language)
 - C'est le même langage que HTML
 - Plus strict, mais plus évolutif et plus convenable que SGML

A decorative graphic on the left side of the slide consisting of two parallel, wavy vertical lines. The inner line is a light blue color, and the outer line is white. They start from the top left and extend towards the bottom left, creating a stylized, organic shape.

HTTP

PROTOCOLE HTTP

PROTOCOLE HTTP

- Quelques commandes HTTP
 - GET
 - POST
 - PUT
 - DELETE

PROTOCOLE HTTP

- Les requêtes HTTP composées de
 - Commande HTTP, URL et version du protocole utilisé
 - Champs d'en-tête [optionnels]
 - Corps de la requête [optionnel]
- L'URL est formée de la manière suivante
 - [protocol]://[host]:[port][request_path]?[query_string]
 - http://localhost:8080/formation-web?id=34
 - http://localhost:8080/formation-web?idProduit=42&fournisseurId=2
 - https://www.google.fr/

PROTOCOLE HTTP

- La réponse HTTP est composée de :
 - Statut et version du protocole utilisé
 - Champs d'en-tête [optionnels]
 - Corps de la réponse

PROTOCOLE HTTP

Code	Type	Message	Définition
200	Succès	OK	Requête traitée avec succès
400	Erreur (côté client)	Bad Request	La syntaxe de la requête est erronée
401		Unauthorized	Une authentification est requise
403		Forbidden	Le serveur refuse d'exécuter la requête. S'authentifier n'y changera rien.
404		Not Found	La ressource n'a pas été trouvée
405		Method Not Allowed	Méthode de requête non autorisée
408	Erreur (côté serveur)	Request Time-out	Temps d'attente d'une réponse serveur écoulé
500		Internal Server Error	Erreur interne du serveur
502		Bad Gateway	Mauvaise réponse envoyée à un serveur intermédiaire par un autre serveur
503		Service Unavailable	Service temporairement indisponible

PROTOCOLE HTTP

- Le protocole HTTP est un protocole dit « déconnecté »
 - Nativement, le serveur ne garde pas en mémoire l'historique des requêtes d'un client
 - Il n'y a pas de cohérence client / serveur
 - Exemple
 - Un client devra envoyer son nom d'utilisateur et son mot de passe à chaque requête pour que le serveur puisse le reconnaître
 - Avec toute autre information nécessaire au bon fonctionnement de l'application Web
 - Pour assurer cette cohérence, l'utilisation de 2 mécanismes est primordiale
 - D'abord côté client, avec les Cookies
 - Puis côté serveur, avec les Sessions

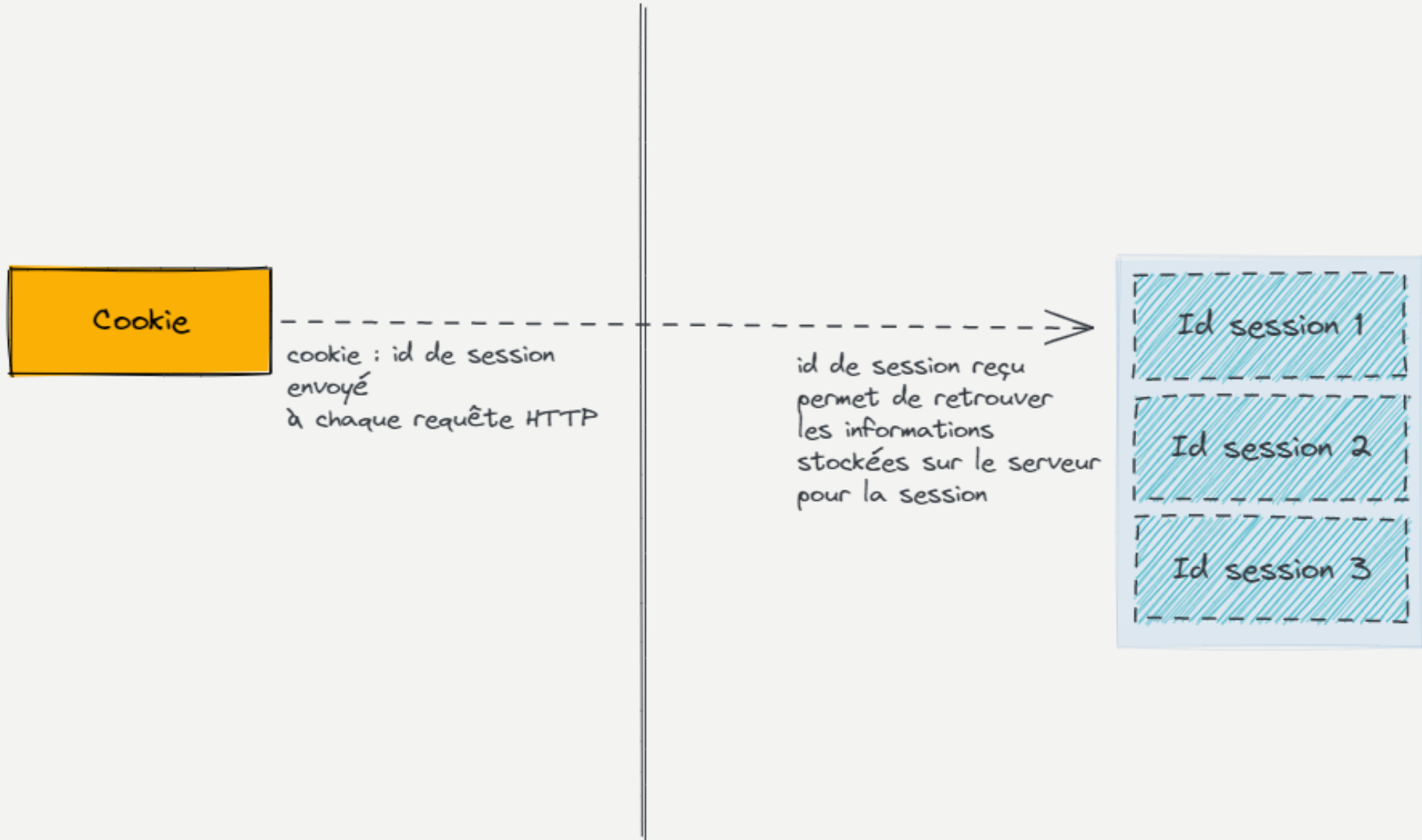
PROTOCOLE HTTP

- Le cookie est une donnée stockée sur le poste client (navigateur)
- Envoyé à chaque requête vers l'hôte pour lequel le cookie a été stocké

PROTOCOLE HTTP

- Espace alloué sur le serveur (donnée stockée sur le serveur)
 - Permet la persistance de données
 - En Java, les informations sont stockées dans l'objet persistant **HttpSession**
 - Permet de maintenir la cohésion entre utilisateur et la requête
-
- L'identifiant de session est stocké dans un cookie du navigateur !

PROTOCOLE HTTP



PROTOCOLE HTTP – EXERCICE

- Démonstration F12



HTML5

LES BASES DE HTML5

LES BASES

- Le « code » HTML est structuré en balises et en attributs
- Une balise est encadré par un chevron inférieur (<) et par un chevron supérieur (>)
- Un attribut est un peu comme une option d'une balise

```
<balise attribut="valeur" autre="autre valeur"></balise>
```

- Une balise ouverte doit impérativement être fermée

```
<div>  
  <span>Mon texte</span>  
</div>
```

- Il existe des balises qui ne possèdent pas de contenu, elles sont « auto-fermantes »

```
<input type="text" name="demo" />
```

LES BASES

- Les balises peuvent s'imbriquer dans d'autres balises
 - Dans ce cas, la première balise ouverte sera la dernière fermée

```
<div>
  <div class="menu">
    <ul>
      <li><a href="#">Mon premier lien</a></li>
      <li><a href="#">Mon deuxième lien</a></li>
    </ul>
  </div>
</div>
```

STRUCTURE

- Générale

```
<!DOCTYPE html>  
<html lang="fr">  
  <!-- Autre contenu -->  
</html>
```

- En-têtes

```
<head>  
  <title>Titre de la page</title>  
</head>
```

Contenu

```
<body>  
  <!-- Contenu de la page -->  
</body>
```

STRUCTURE – EN-TÊTE <HEAD>

- Dans cette partie on y trouve des informations générales telles que
 - Le titre de la page
 - Des balises « meta »
 - Description
 - Autres éléments SEO (référencement Web)
 - Responsive
 - La gestion des caractères (encodage)
 - L'intégration de feuilles de style CSS
 - L'intégration de scripts JavaScript

STRUCTURE – EN-TÊTE <HEAD>

- Titre de la page

```
<title>Le titre de la page</title>
```

- Spécification de l'encodage

```
<meta charset="UTF-8" />
```

- Inclure une feuille de style CSS

```
<link rel="stylesheet" type="text/css" href="assets/css/le-fichier-css.css" />
```

- Inclure un script Javascript

```
<script src="assets/js/le-script.js"></script>
```

- Préciser une URL par défaut pour tous les liens

```
<base href="/repertoire-racine/" />
```

STRUCTURE – EN-TÊTE <HEAD>

- Responsive

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

- Indiquer à IE (ou Edge) en quelle version la page doit être interprétée (spécifique IE)

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

- Liste des balises meta influencent les navigateurs et le référencement
 - <https://www.metatags.org/all-meta-tags-overview/>

STRUCTURE – CORPS <BODY>

- Dans cette partie on y trouve tout le contenu de la page, écrit de façon à ce que le rendu sur les navigateurs soit optimal
- On peut aussi y insérer des scripts JavaScript

STRUCTURE

- HTML est interprété par les navigateurs
- Le code est téléchargé, il est lisible par tous les internautes
- Les ressources CSS, JavaScript, images, ... sont téléchargées par le navigateur pendant la lecture de la page HTML

STRUCTURE

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <base href="/">
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Le titre de la page</title>
  <link rel="stylesheet" type="text/css" href="/assets/css/global.css" />
</head>
<body>
  <div>
    <h1>Un titre</h1>
    <p>Un texte avec un <a href="autre-page.html">lien</a></p>
  </div>

  <script src="/assets/js/app.js"></script>
</body>
</html>
```

BALISES DE CONTENU

Type de contenu	Balises	Attributs	Auto-fermantes
Zone	div, section, nav, article, aside, header, footer		
Texte	p, span, label, b, i, u, small, strong		
Liste de valeur	ul, ol, li		
Image	img	src, title, alt	X
Lien	a	href, target	
Titres (de contenu)	h1, h2, h3, h4, etc.		
Retour à la ligne	br		X
Zone de saisie de texte	input	type, value, name	X
Liste déroulante	select, option	name, value, selected	
Tableau	table, thead, tr, th, tbody, td, tfoot		

ATTRIBUTS

Attribut	Définition
id	Identifiant de l'élément
style	Définir un style CSS
class	Appliquer un style défini dans une classe CSS (séparateur espace si plusieurs classes)
src	Source de la ressource (image, script, ...)
alt, title	Titre ou description de l'élément, facilite le SEO
href	Lien vers
target	Précise si le lien doit s'ouvrir sur la même page ou non
value	Valeur

EXERCICE

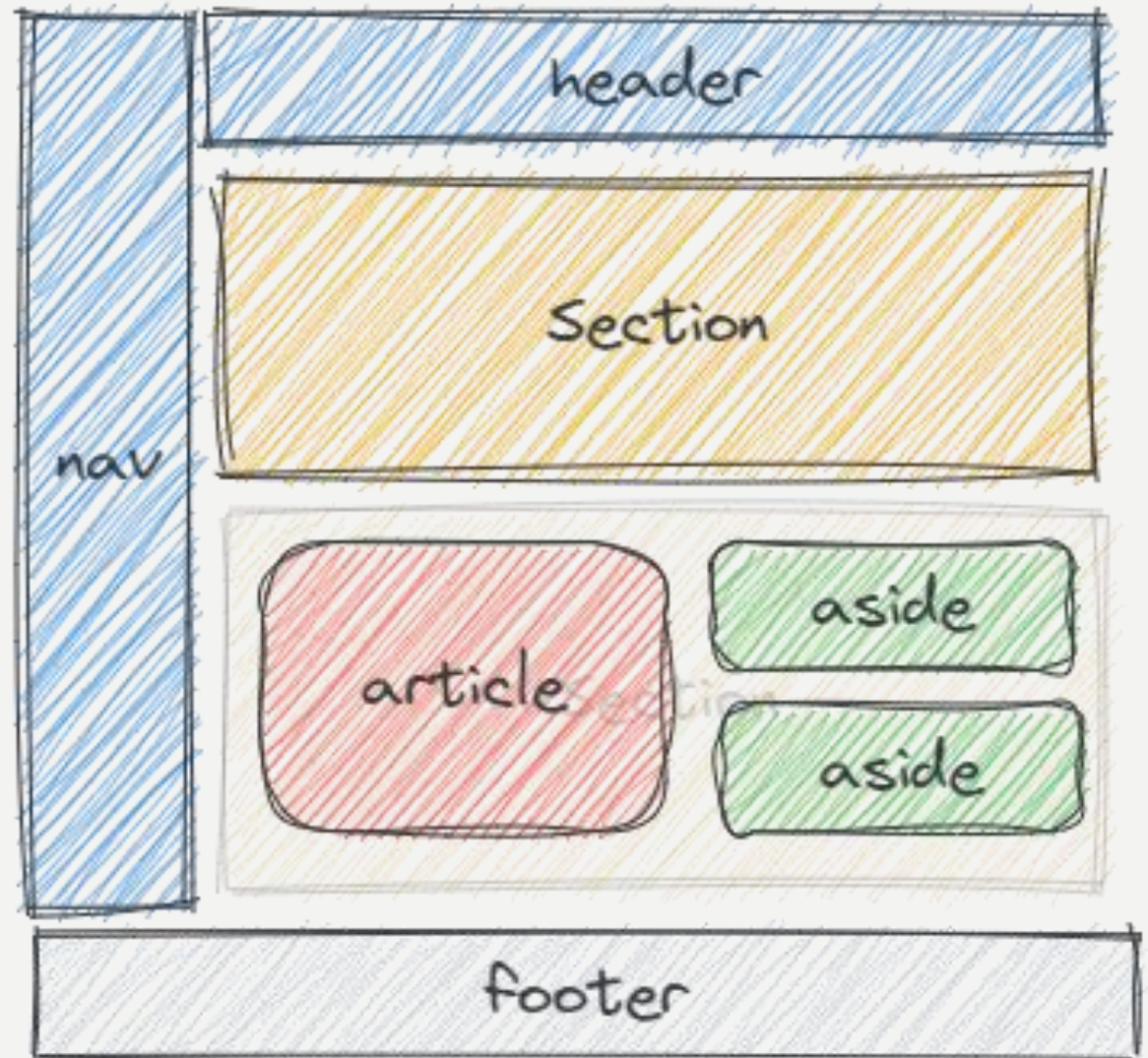
- Utilisation de VSCode avec l'extension « Live Server »
 - Possibilité d'utiliser « Emmet » pour générer du code depuis VSCode

EXERCICE

- Construire une page d'accueil
 - Titre → Tableau de bord
 - Description → Consulter les statistiques sur ce tableau de bord
 - Menu avec 2 liens (Utiliser les balises "ul" et "li")
 - « accueil » (index.html)
 - « Liste des clients » (client/liste.html)
 - Titre (de contenu) → Tableau de bord
 - Paragraphe → Visualiser les statistiques en temps réel
 - Le mot « temps réel » doit apparaître en **gras**
 - Voir <https://www.w3schools.com/html/>

STRUCTURE – SÉMANTIQUE

- Les balises sémantiques de HTML5
 - header
 - main
 - footer
 - nav
 - section
 - article
 - aside
 - details



STRUCTURE – SÉMANTIQUE

- L'intérêt d'utiliser le sémantique permet d'informer sur le type de contenu
 - Pour le référencement
 - Pour les développeurs

EXERCICE

- Modifier l'exercice précédent
 - Placer le menu dans l'espace de navigation
 - Placer le titre dans l'espace d'en-tête
 - Placer le contenu dans une section
 - Ajouter un pied de page
 - « Copyright formation.fr »

EXERCICE

- Créer une page « client/liste.html »
 - Répéter le menu
 - Ajouter un tableau HTML de 2-3 clients
 - Nom, Prénom, Adresse e-mail, Montant C.A.

LES FORMULAIRES

- Balises
 - form
 - method Méthode HTTP (GET ou POST)
 - action Pour la page de destination
 - input
 - name Important, c'est lui qui est utilisé comme nom de paramètre
 - type Type de saisie (texte, email, couleur, date, nombre, case à cocher)
 - label
 - select
 - name Important, c'est lui qui est utilisé comme nom de paramètre
 - option
 - value A utiliser dans le select
 - value Valeur envoyée en paramètre si sélectionné
 - input de type submit Bouton d'envoi du formulaire

LES FORMULAIRES

```
<form method="POST" action="autrepape.php">
  <div>
    <label for="nom">Nom</label>
    <input type="text" id="nom" name="nom" />
  </div>

  <div>
    <input type="submit" value="Envoyer !" />
  </div>
</form>
```

LES FORMULAIRES

- Contraintes de validation sur les éléments input

```
<input type="text" name="fournisseur" pattern="amazon|gopro" />
```

```
<input type="text" name="fournisseur" pattern="[Aa]mazon|[Gg]opro" />
```

- Utilisation de l'attribut « required »

```
<input type="text" name="fournisseur" pattern="amazon|gopro" required />
```

- Utilisation de min / max

```
<input type="number" name="prix" min="0" max="20" />
```

- Ajouter une information lorsque la valeur n'a pas été saisie

```
<input type="text" name="fournisseur" placeholder="Saisir le nom du fournisseur" />
```

LES FORMULAIRES

- Boite de sélection
 - Contient des « options », qui est la liste de valeurs disponibles
 - Chaque option peut avoir une valeur, c'est la valeur que le paramètre « name » prendra à l'envoi
 - L'attribut « selected » permet de sélectionner l'option par défaut

```
<select name="fournisseur_id">  
  <option value="1" selected>Amazon</option>  
  <option value="2">GoPRO</option>  
  <option value="3">Autre</option>  
</select>
```

LES FORMULAIRES

- Choix unique

```
<label for="genre0">Homme</label>  
<input type="radio" name="genre" value="0" id="genre0" />  
  
<label for="genre1">Femme</label>  
<input type="radio" name="genre" value="1" id="genre1" />
```

LES FORMULAIRES

- Choix multiples

```
<label for="cgv">Accepter les CGV</label>
<input type="checkbox" name="accept_cgv" value="1" id="cgv" />

<label for="news">Recevoir la newsletter</label>
<input type="checkbox" name="accept_newsletter" value="1" id="news" />
```

```
<label>
  Accepter les CGV
  <input type="checkbox" name="accept_cgv" value="1" />
</label>
```


EXERCICE

- Créer une page « client/ajouter.html » avec un formulaire HTML
 - Genre (homme ou femme)
 - Nom
 - Prénom
 - Adresse e-mail
 - Date de naissance
 - Type de client (Pas encore client, client, grand client)
 - Description (textarea)

LES +

- ContentEditable
 - Utilisé pour éditer du contenu (éditeur de texte WYSIWYG par exemple, éditeur Monaco, etc.)
- Web Storage
 - Stockage d'information sur le navigateur
- Web SQL Database
 - Stockage de données dans une base SQLite, disponible sur le navigateur
- Web Socket
 - Echanges d'information en temps-réel
- Geolocalisation
- Drag n Drop
- SVG
 - Dessins vectoriels
- Canvas
 - Dessiner des graphiques
- Video / Audio