



**formation.tech**

# HTML

Romain Bohdanowicz  
<http://formation.tech/>  
Twitter : @bioub



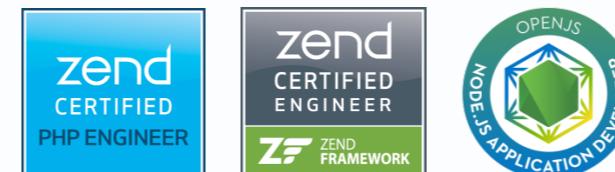
**formation.tech**

# Introduction



# Introduction - Formateur

- Romain Bohdanowicz  
Ingénieur EFREI 2008, spécialité en Ingénierie Logicielle
- Expérience  
Formateur/Développeur Freelance depuis 2006  
Près de 2000 jours de formation animées
- Langages  
Expert : HTML / CSS / JavaScript / TypeScript / PHP / Java  
Notions : C / C++ / Objective-C / C# / Python / Bash / Batch
- Certifications  
PHP / Zend Framework / Node.js
- A propos  
Premier site web à 12 ans (HTML/JS/PHP)  
Triathlète du dimanche



# Introduction - Horaires

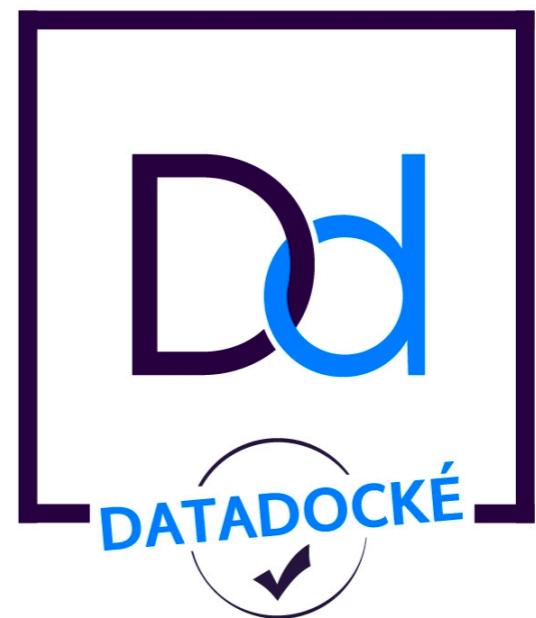


- Matin
  - 9h - 10h
  - 10h15 - 11h15
  - 11h30 - 12h30
- Après-midi
  - 13h45 - 14h45
  - 15h - 16h
  - 16h15 - 17h15
- Questionnaire de satisfaction à remplir en fin de formation :  
<https://stagiaire.formation.tech/>

# Introduction - formation.tech



- Organisme de formation depuis 2016
- Référencé DataDock
- Certifié Qualiopi
- 15 formations au catalogue
- Une dizaine de formateurs indépendants
- Formations en français ou anglais
- <https://formation.tech/>



# Introduction - WeAreDevs



- Studio de développement créé en 2017
- 1 salarié développeur senior
- Principales références
  - Cinexpert / Adeum
  - Sponsorise.me
  - Intel
  - Staytuned
  - STMicroelectronics
- <https://wearedevs.fr/>



# Introduction - Et vous ?



- Pré-requis ?
- Rôle dans votre société ?
- Intérêt / objectif de cette formation ?



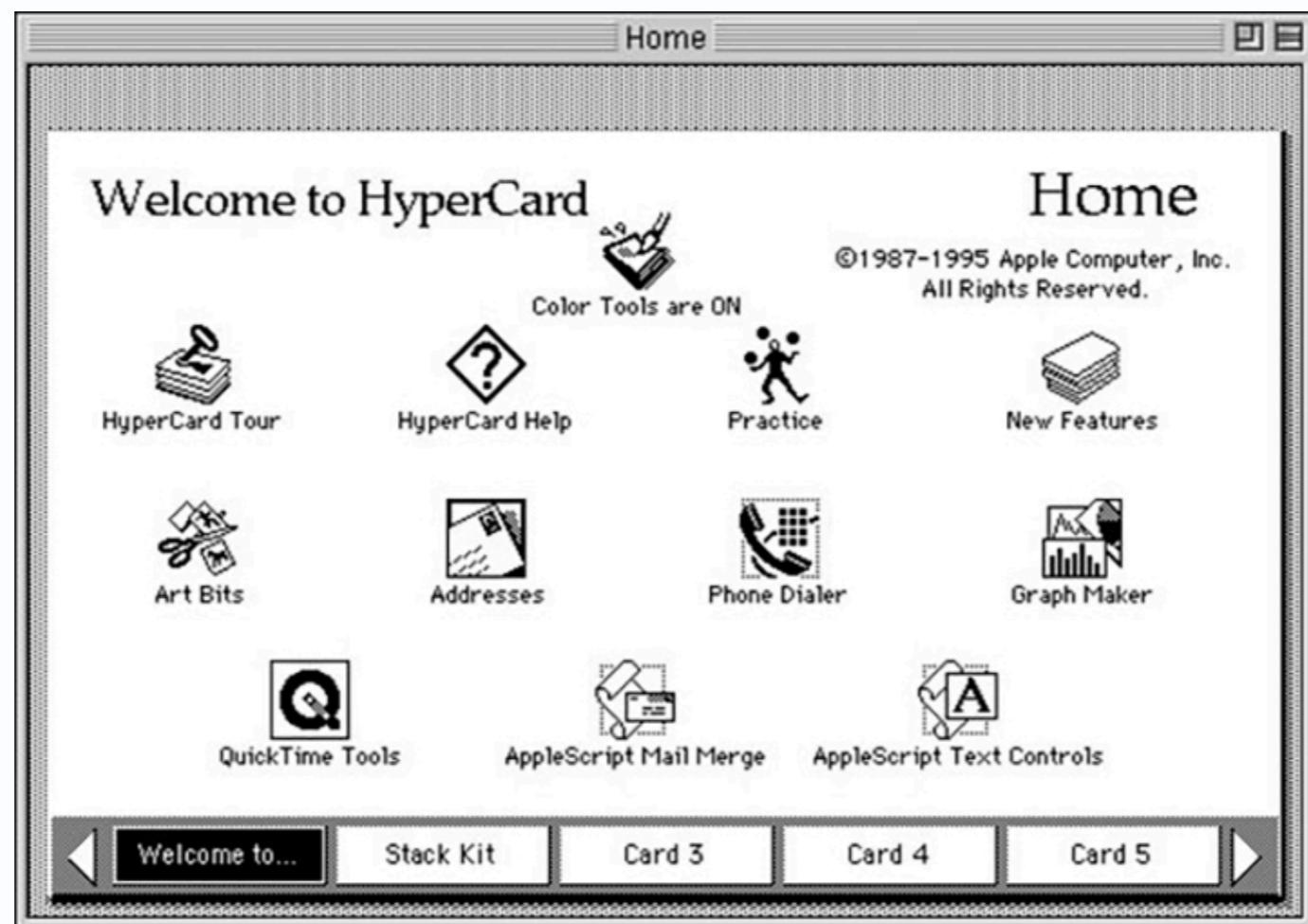
**formation.tech**

# Technologies Web



# Technologies Web - Introduction

- Tim Berners-Lee, chercheur au CERN à Genève créé le web en 1989
- Il s'inspire d'un logiciel d'Apple appelé Hypercard, permettant de visualiser des "piles de cartes" en local, similaires aux pages web et aux liens
- L'idée est d'adapter le concept pour que les pages puissent être hébergées sur un serveur distant et consultables via le réseau

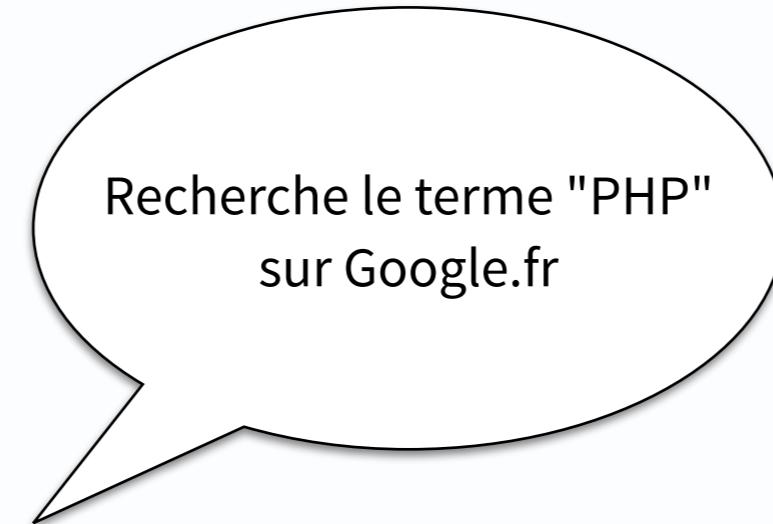




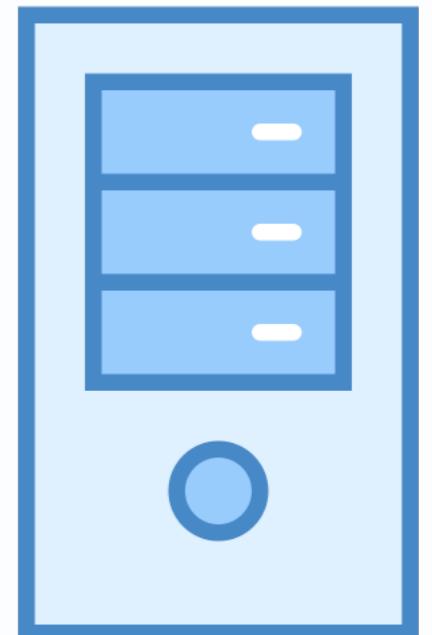
# Technologies Web - Client / Serveur



Client



Réseau



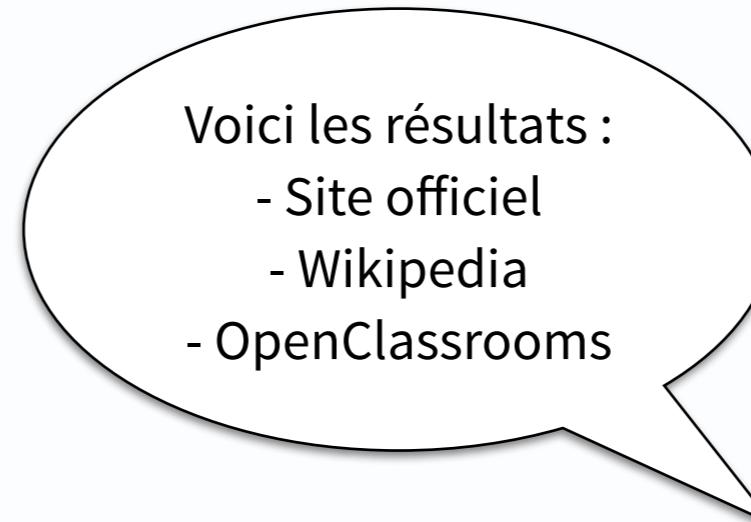
Serveur



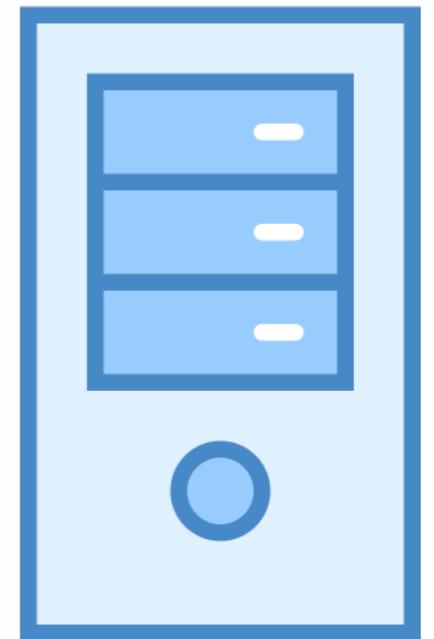
# Technologies Web - Client / Serveur



Client

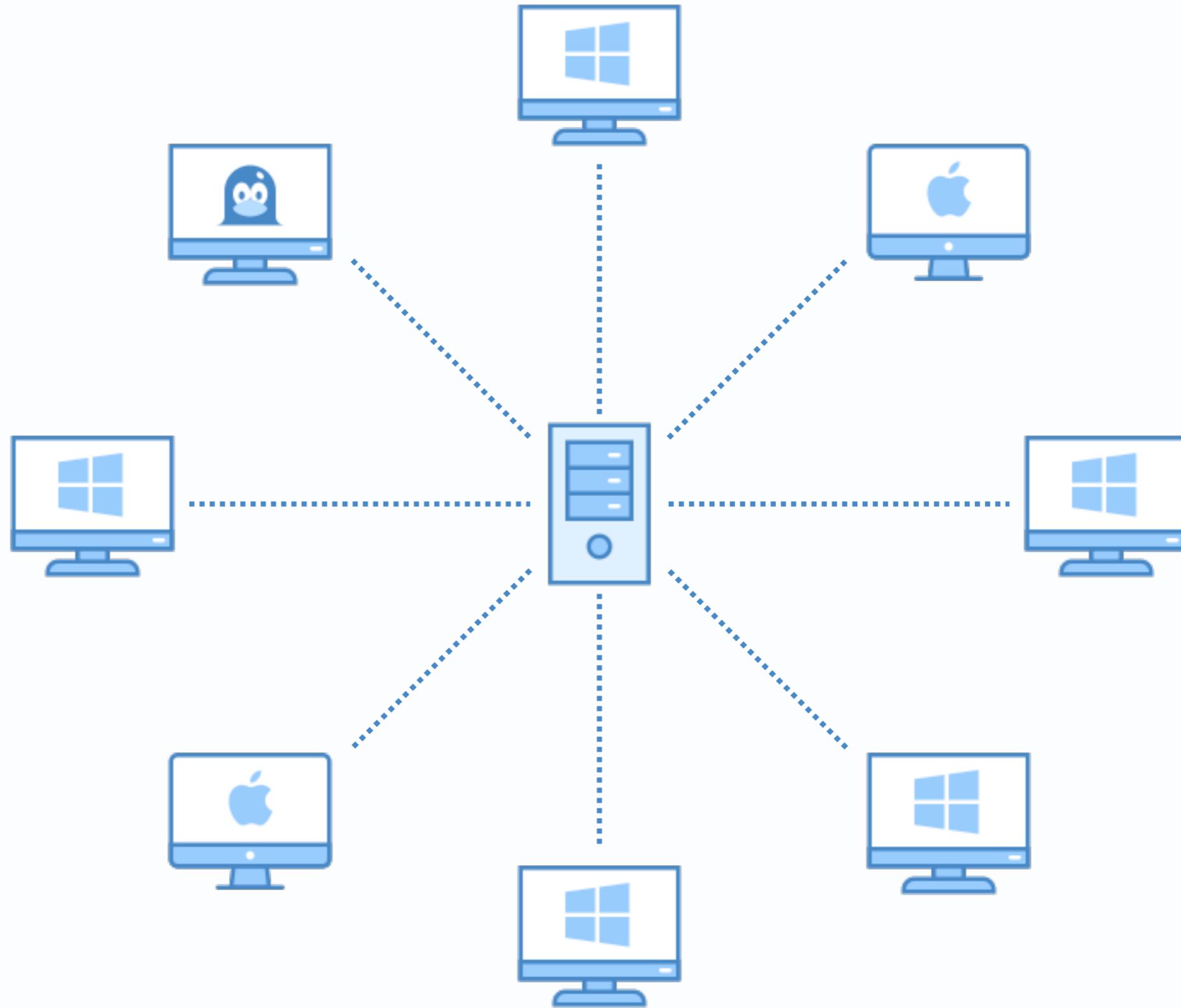


Réseau

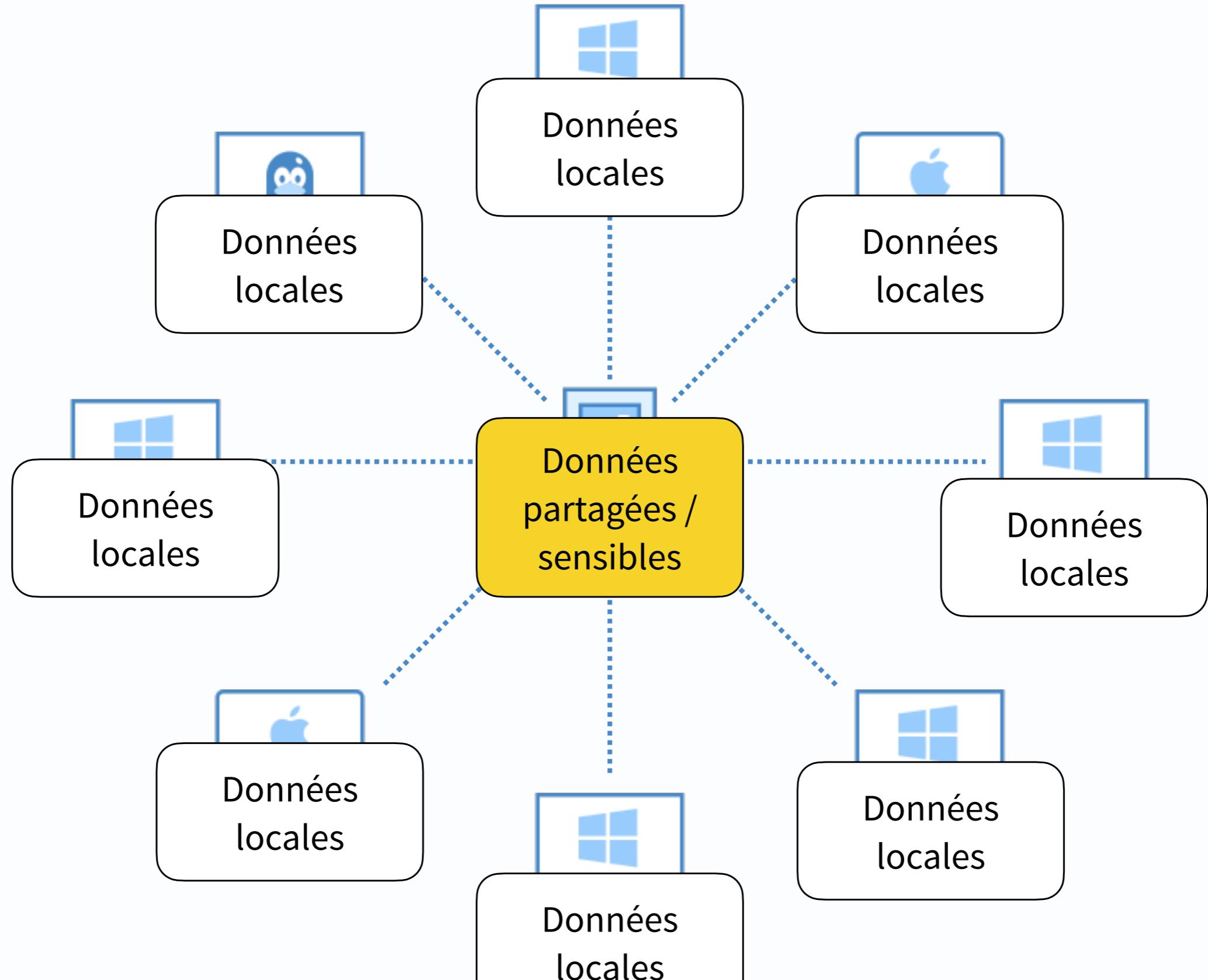


Serveur

# Technologies Web - Client / Serveur

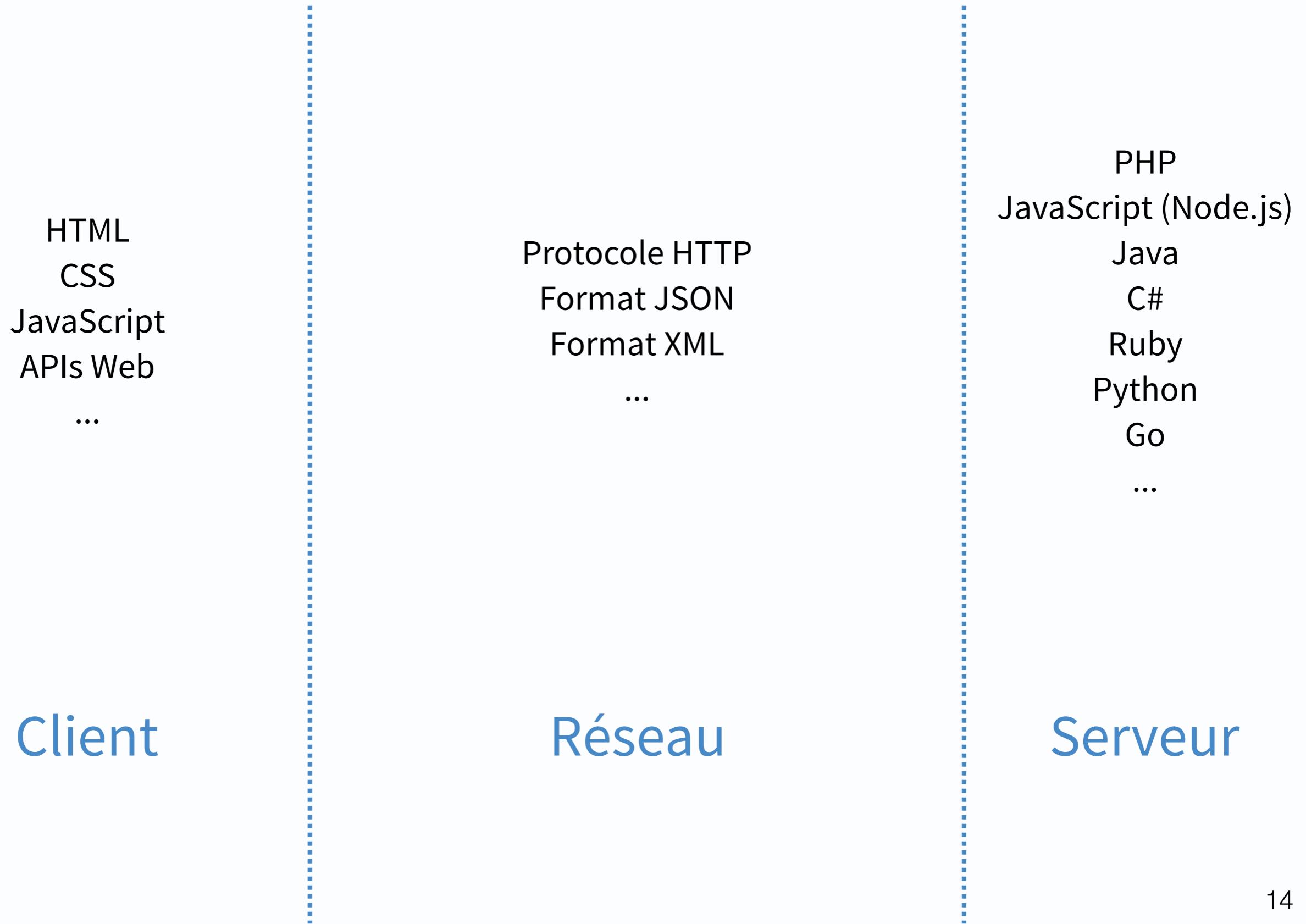


# Technologies Web - Client / Serveur





# Technologies Web - Technologies / Langages



# Technologies Web - HTTP



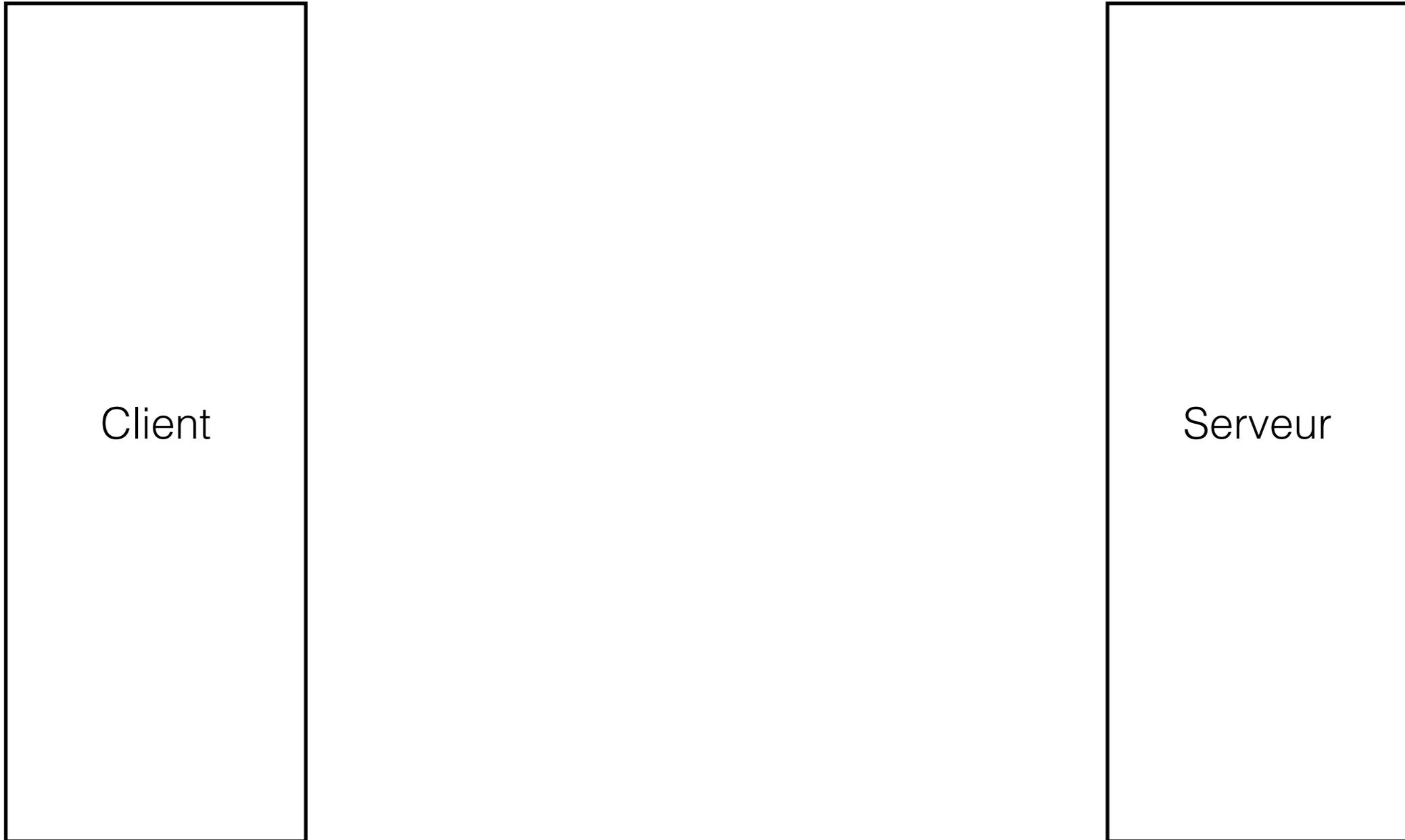
- Protocole réseau (norme réseau) permettant l'échange de données entre un client et un serveur web
- Crée en 1990 par Tim Berners-Lee alors chercheur au CERN
- Normé pour la première fois en 1996, HTTP/1.0 dans la RFC 1945  
<https://tools.ietf.org/html/rfc1945>
- Nouvelles fonctionnalités dans HTTP/1.1, RFC 2616  
<https://tools.ietf.org/html/rfc2616>  
Version la plus répandu aujourd'hui
- Evolution majeure en 2015 avec HTTP/2, RFC 7540  
<https://tools.ietf.org/html/rfc7540>



# Technologies Web - HTTP

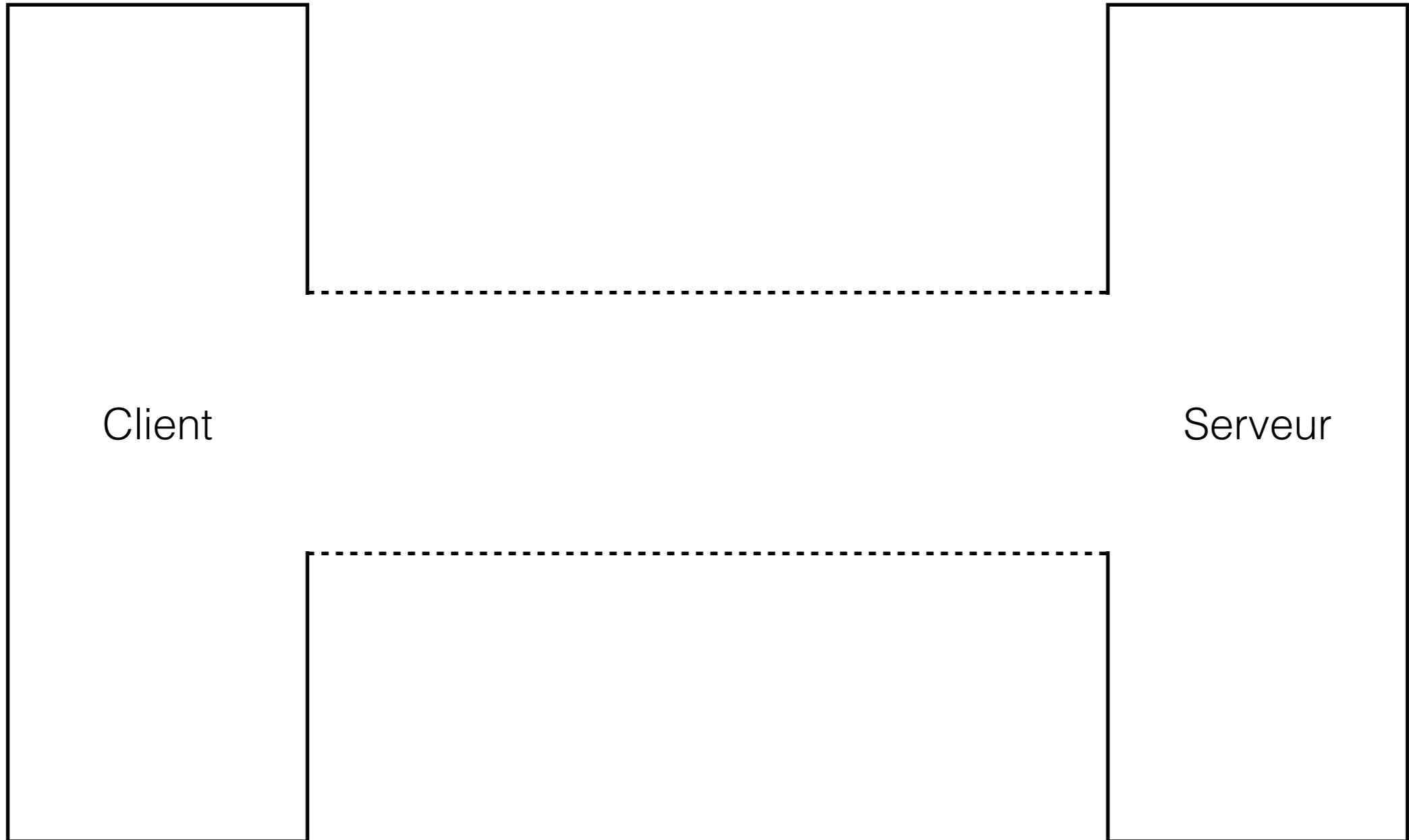
- Quatre étapes
  1. connexion du client HTTP
  2. envoi d'une requête
  3. réponse du serveur HTTP
  4. le serveur ferme la connexion pour signaler la fin de la réponse.
- La requête étant à l'initiative du client on parle de *Client Pull*, par opposition au *Serveur Push* où le serveur est à l'origine

# Technologies Web - HTTP



2 programmes souhaitent échanger des données via le réseau

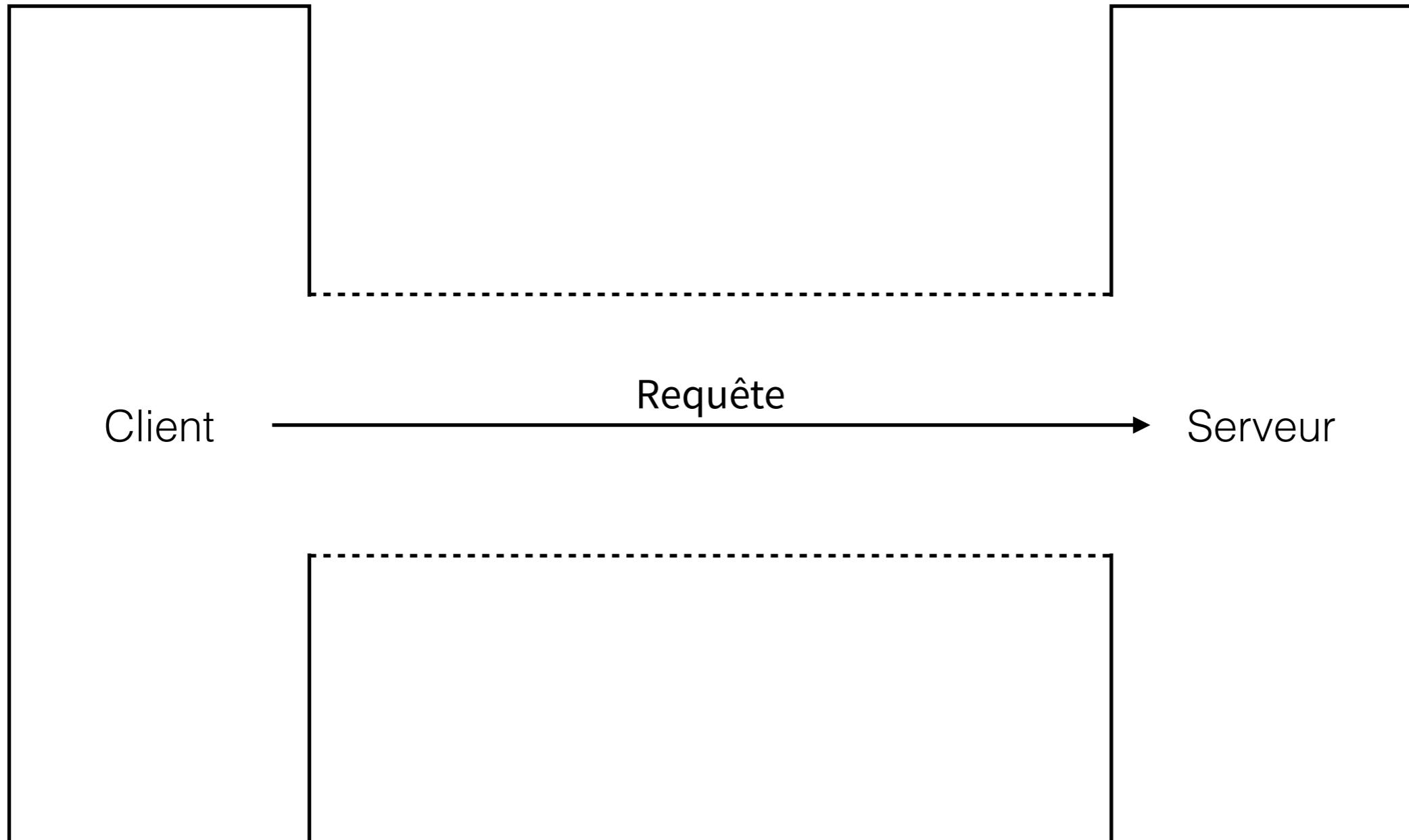
# Technologies Web - HTTP



Le client ouvre une connection (socket) entre  
lui et le serveur

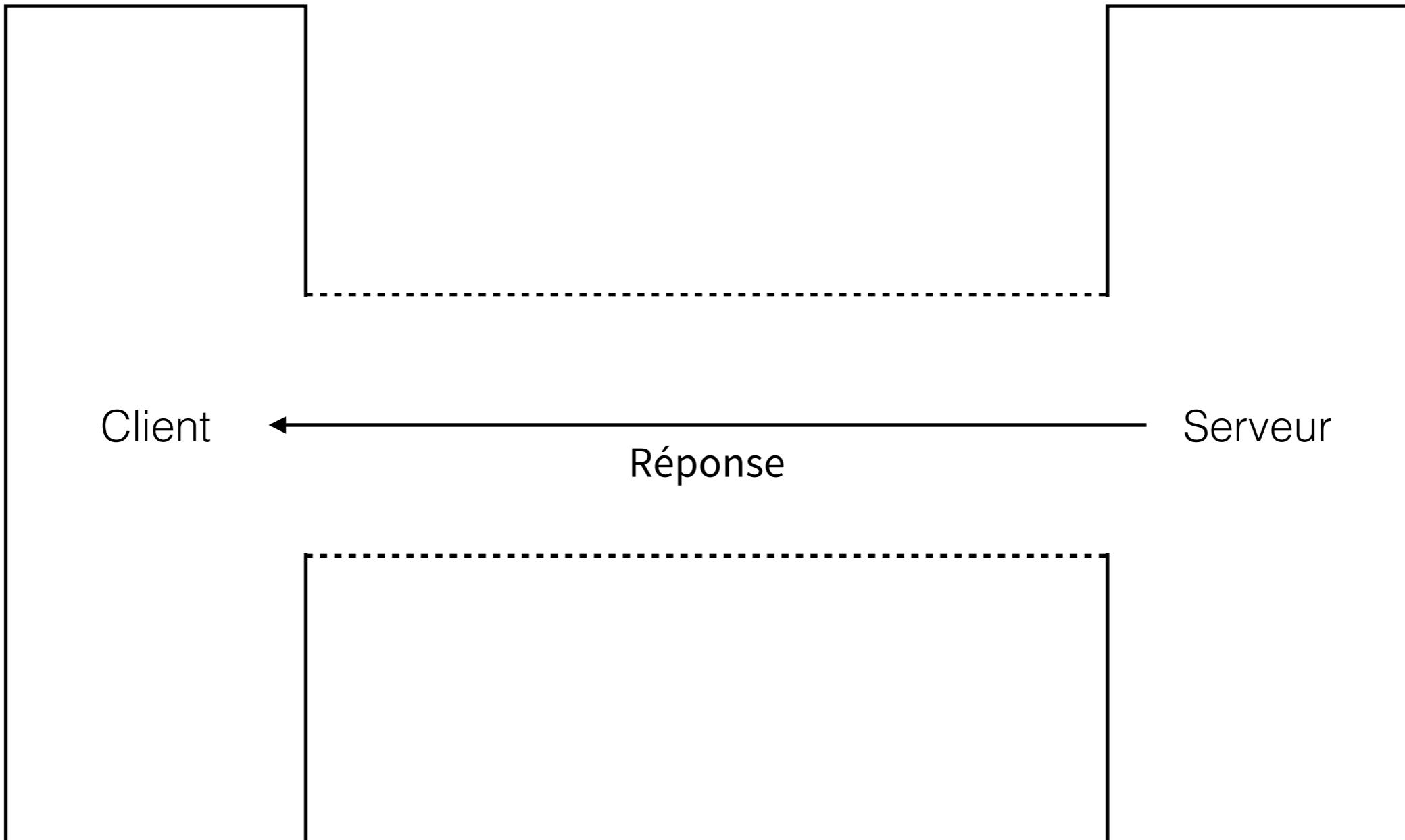


# Technologies Web - HTTP



Le client envoie une requête (request)

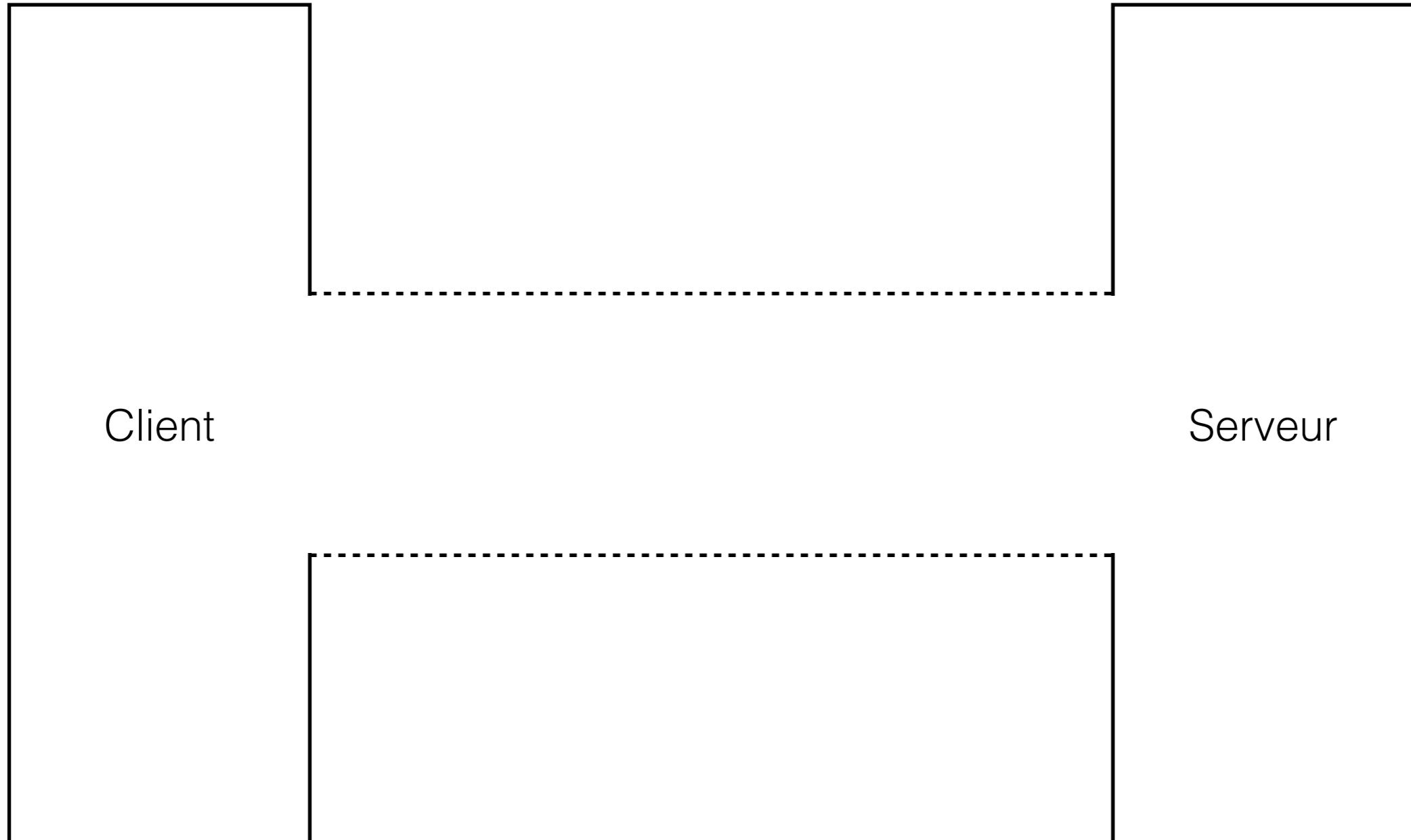
# Technologies Web - HTTP



Le serveur traite la requête et émet une réponse (response)



# Technologies Web - HTTP



Le serveur ferme la connection



# Technologies Web - HTTP

connexion du client HTTP

envoi d'une requête

réponse du serveur HTTP

le serveur ferme la connexion pour signaler la fin de la réponse.

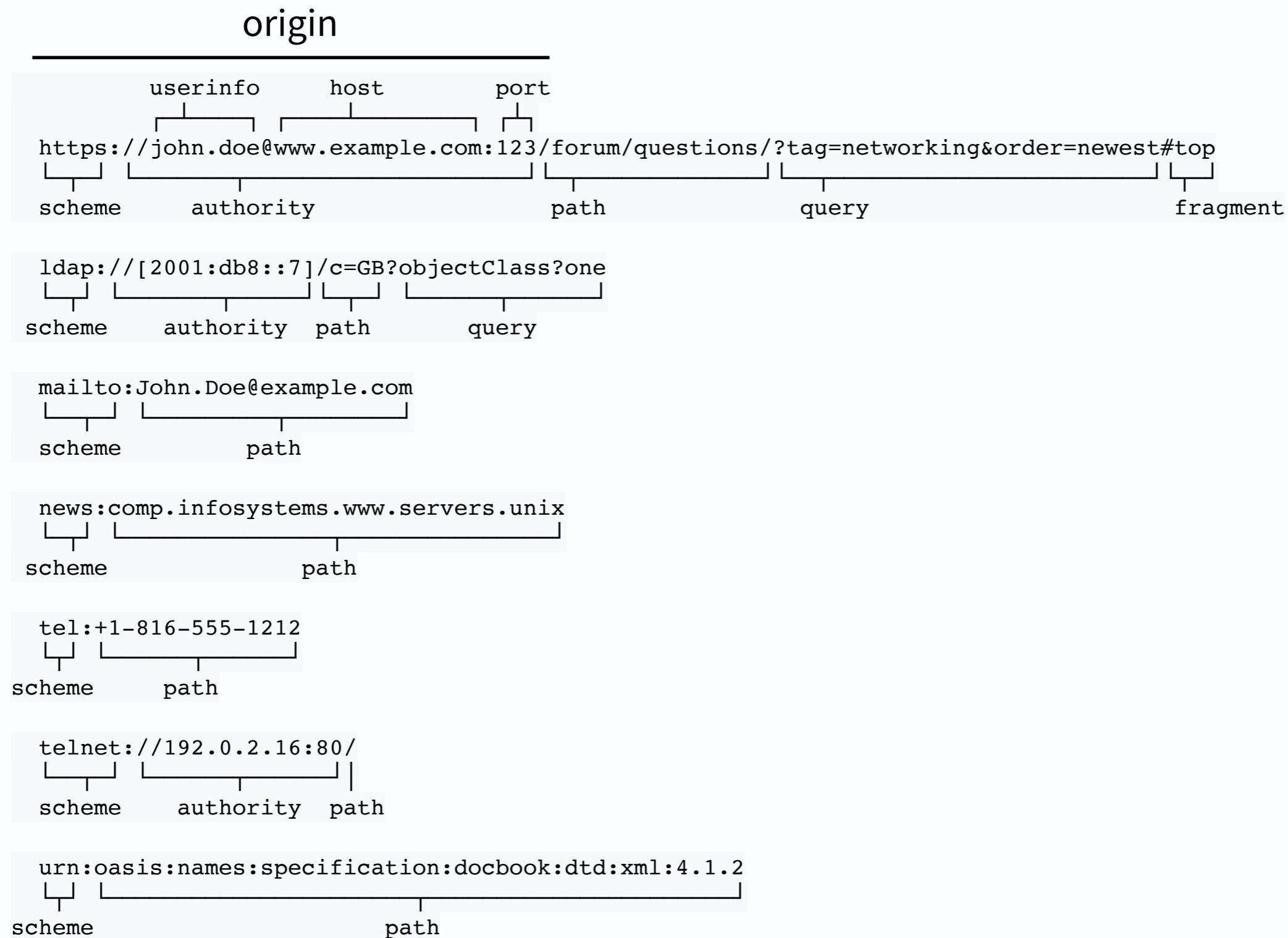
```
MacBook-Pro:~ romain$ telnet example.com 80
Trying 93.184.216.34...
Connected to example.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: example.com

HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html
Date: Sat, 11 Aug 2018 14:37:14 GMT
Etag: "1541025663"
Expires: Sat, 18 Aug 2018 14:37:14 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (dca/24D1)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270

<!doctype html>
<html>
<head>
...
</body>
</html>
Connection closed by foreign host.
```



# Technologies Web - URI



# Technologies Web - Langages



- Principaux langages compris par les navigateurs modernes
  - HTML : structure de la page (apprentissage facile)
  - CSS : mise en forme (apprentissage intermédiaire)
  - JavaScript : interactions dynamiques avec l'utilisateur ou le serveur (apprentissage difficile)
- D'autres moins connus :
  - SVG, RSS, ATOM, MathML, ...
- Langages du serveur :
  - PHP, Java, C#, Ruby, Python, Go, JavaScript...



**formation.tech**

# Normes HTML

# Normes HTML

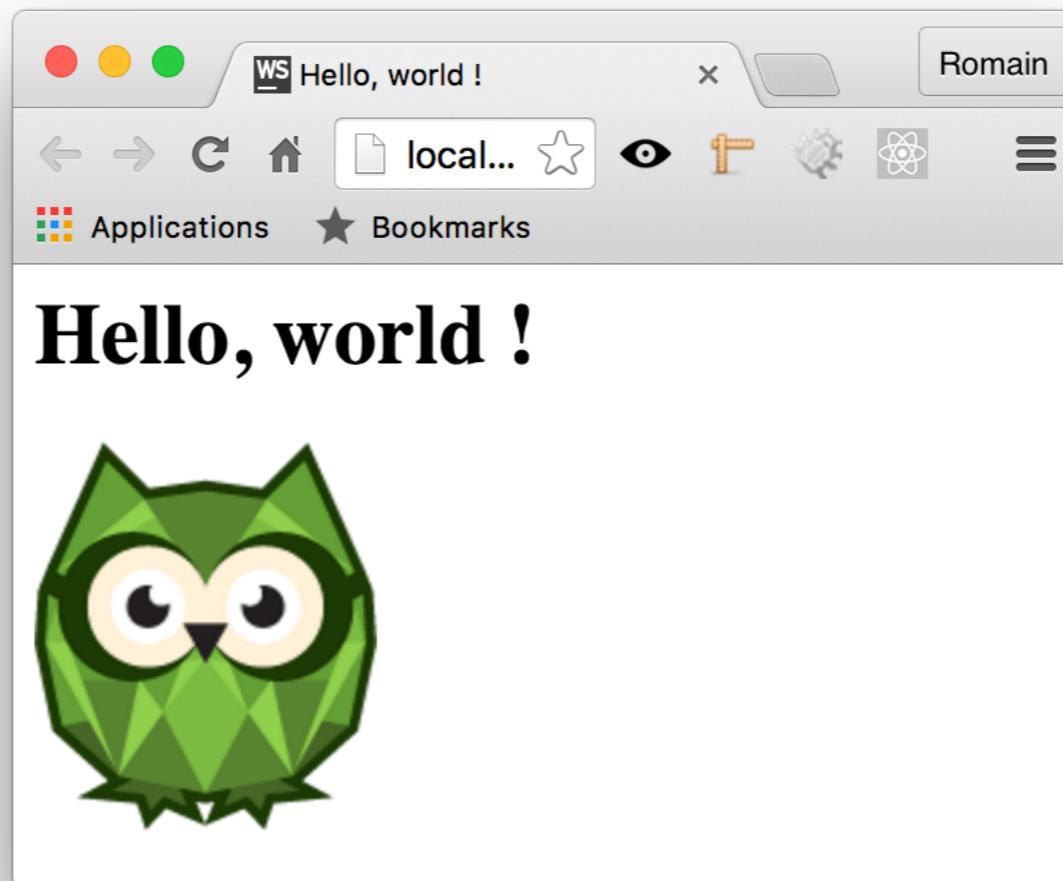


- Différentes époques :
  - Jusqu'à HTML 3.2, pas de norme
  - HTML 3.2 à HTML 4.01, norme du W3C basé sur SGML
  - XHTML 1.0 et 1.1, norme du W3C basé sur XML
  - HTML5 ou HTML Living Standard, norme du WHATWG



# HTML sans norme

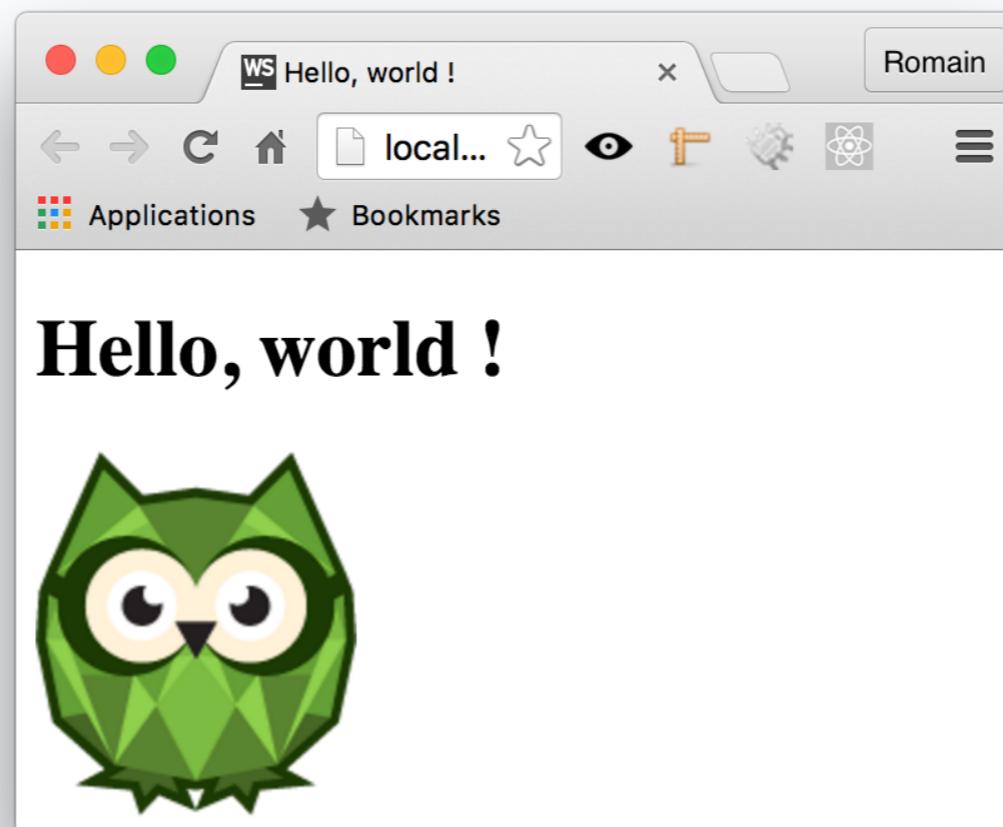
```
<HTML>
<HEAD>
    <TITLE>Hello, world !</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
    <H1>Hello, world !</H1>
    <IMG SRC="logo-formation.tech.png">
</BODY>
</HTML>
```



# HTML 4.01



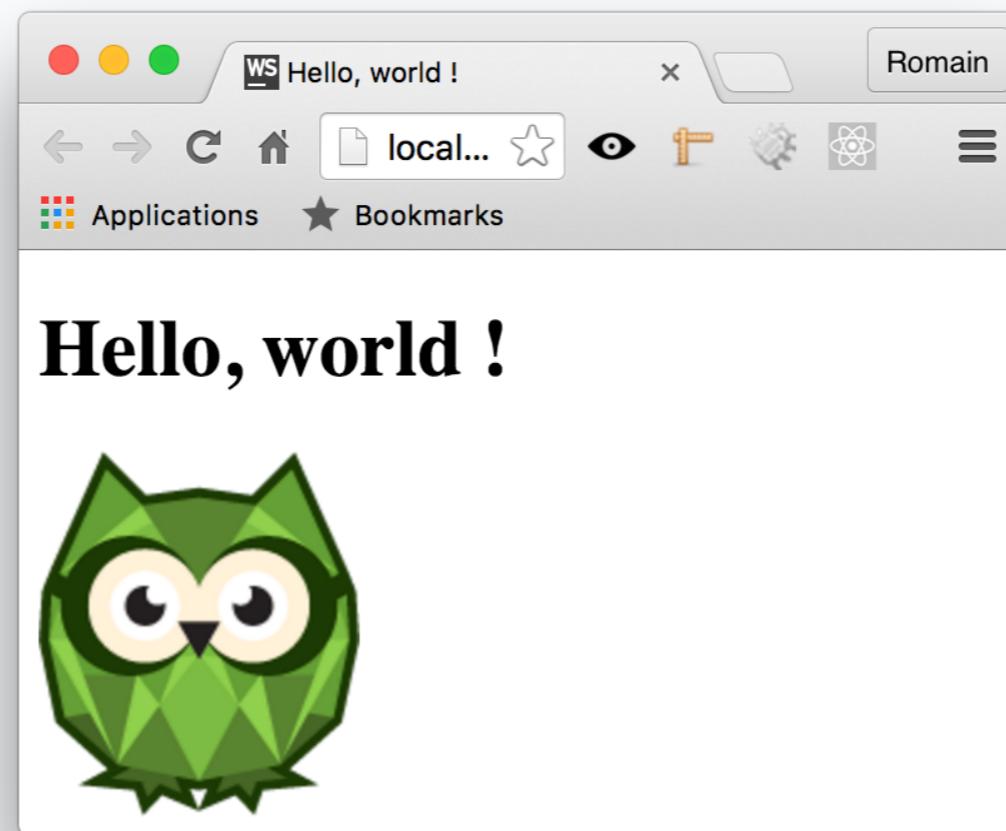
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
      "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
  <title>Hello, world !</title>  
</head>  
<body>  
  <h1>Hello, world !</h1>  
  <img src=logo-formation.tech.png alt="Logo Formation.tech">  
</body>  
</html>
```





# XHTML 1.0

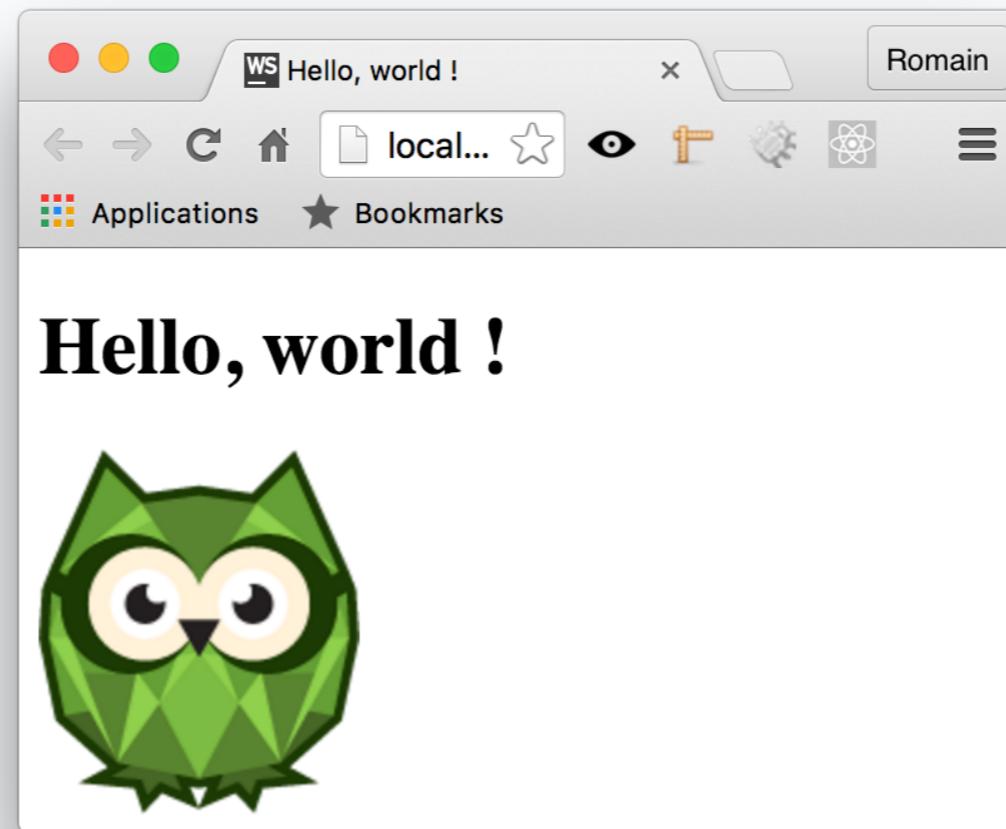
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Hello, world !</title>
</head>
<body>
    <h1>Hello, world !</h1>
    
</body>
</html>
```



# HTML5 / HTML Living Standard



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Hello, world !</title>
</head>
<body>
  <h1>Hello, world !</h1>
  <img src=logo-formation.tech.png alt="Logo Formation.tech">
</body>
</html>
```





# A propos d'HTML5

- **HTML5**
  - Mésentente entre le W3C et les éditeurs de navigateurs qui finissent par créer leur propre organisme de normalisation : le WHATWG
  - Normer l'existant
  - Demander aux utilisateurs du langage de quoi ils avaient besoin
  - Normer la correction d'erreur
- **HTML5, un terme marketing**
  - Le successeur de Web 2.0
  - Nouveautés HTML
  - Nouveautés CSS (principalement CSS3)
  - Nouveaux APIs du Navigateur (Géolocalisation, Notifications, ...)

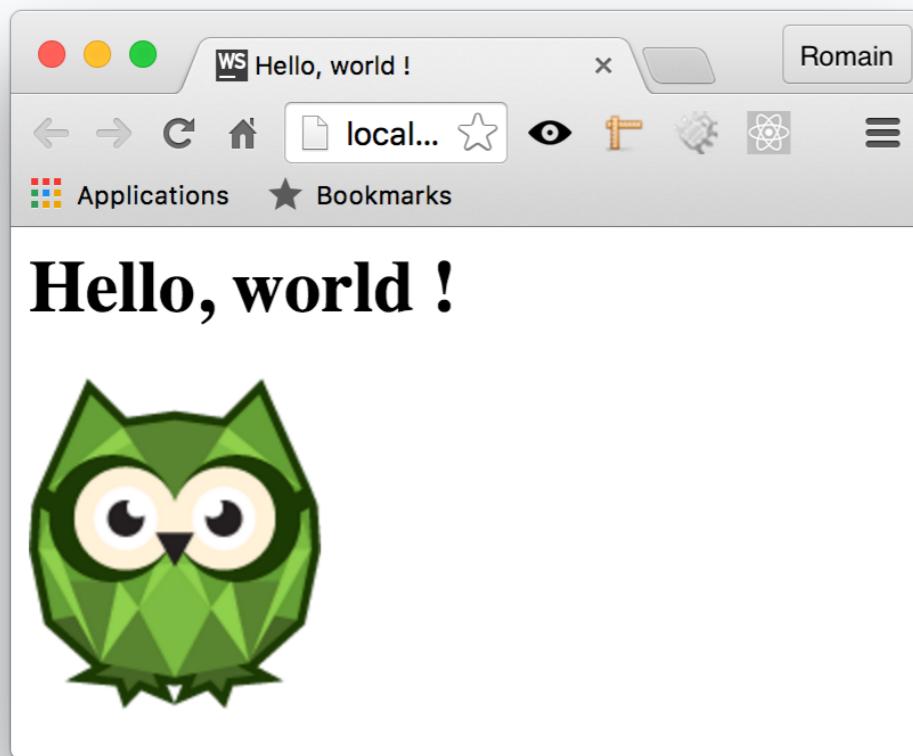
# Différences de rendu



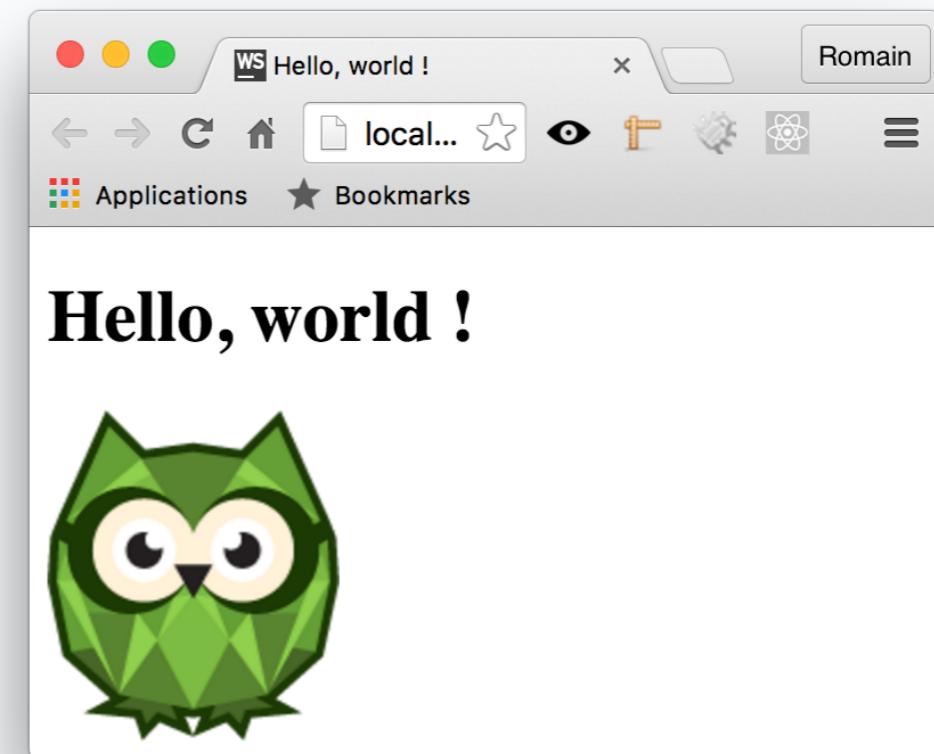
- ▶ Différences de rendu

- Le calcul des marges est différent
- N'oubliez pas le DOCTYPE !

HTML non normé  
(mode quirks)



HTML normé  
(mode standard)





# Valider son HTML

- Vérifier que son Navigateur respecte bien la norme annoncée :
  - <http://validator.w3.org>
- Pour être valide :
  - Respecter la syntaxe (<, >, ", ...)
  - Respecter la grammaire (mots clés, structure...)

The screenshot shows a web browser window with the URL 'validator.w3.org' in the address bar. The page title is 'Markup Validation Service' with the subtitle 'Check the markup (HTML, XHTML, ...) of Web documents'. There are three tabs at the top: 'Validate by URI' (selected), 'Validate by File Upload', and 'Validate by Direct Input'. Below these tabs is a section titled 'Validate by URI' with the sub-instruction 'Validate a document online:'. It features a text input field labeled 'Address:' and a link 'More Options'. At the bottom of this section is a large blue 'Check' button. A descriptive note at the bottom of the page states: 'This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are other'.



**formation.tech**

# Structure d'une page web



- Un fichier HTML est une succession de « balises », imbriquées les unes dans les autres
- Une balise peut être ouvrante, fermante, ou vide (pas de contenu), il faut connaître le langage pour savoir celle qui sont vide ou non.
- Format d'une balise avec une partie ouvrante puis fermante :
  - <**nomDeLaBalise**></**nomDeLaBalise**>
  - <**nom-de-la-balise**></**nom-de-la-balise**>
- Format d'une balise vide :
  - <**baliseVide**>

# Imbrication



- Les balises d'une page web sont imbriquées les unes dans les autres
- Exemple :

```
<baliseParent>
    <baliseEnfant>Contenu</baliseEnfant>
</baliseParent>
```

- Pour faciliter la lecture du HTML, à chaque fois que l'on imbrique une balise dans une autre, on décale la balise enfant de 2 espaces (parfois 4, parfois des tabulations) : on appelle cela l'indentation.
- Les balises ne peuvent pas se chevaucher (penser à des dossiers dans des dossiers) :

```
<baliseParent>
    <baliseEnfant></baliseParent>
</baliseEnfant>
```



# Attributs

- › Parfois les balises contiennent des informations secondaires qu'on ne souhaite pas afficher à l'écran
- › Exemple :

```
<balise attribut="valeur" autre="hello">Contenu à afficher</balise>
```

# Structure minimale



- A minima une page HTML contient les balises suivantes :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
  </head>
  <body>
    </body>
</html>
```

- DOCTYPE pour indiquer la norme utilisées (attention sans DOCTYPE le rendu est différent)
- html : balise racine qui englobe obligatoirement toutes les autres
- head : toutes les données secondaires qu'on affiche pas sur la pages, ou indirectement (chargement de fichiers CSS, titre de l'onglet...)
- body : contenu de la page

# Exercices



- ▶ Exercices
  - Créer un nouveau fichier HTML : index.html
  - Remplir ce fichier avec la structure minimale d'une page web



**formation.tech**

# Entêtes d'une page web (balise head)



# Balise head

- Entête de la page

Le contenu de la balise head n'a pas vocation à s'afficher sur la page, mais est plutôt utilisé par les programmes qui vont lire le HTML (navigateurs, moteurs de recherche, mobiles...)



# Encodage

- L'encodage de la page

Un ordinateur ne manipule que des nombres (computer en anglais, « machine qui compte »), les caractères n'existent pas réellement, on leur a donc attribué un code sous la forme d'un nombre, la lettre A majuscule a par exemple le code 65 quand le e minuscule le code 101.

- Différents encodages

Aujourd'hui l'encodage le plus universel est l'UTF-8, mais il en existe d'autres : windows-latin-1 ou iso-8859-1, ce qui explique que parfois les lettres accentuées n'apparaissent pas correctement lorsque l'encodage est mal renseigné.

- Spécifier l'encodage d'un page web dans les 512 premiers caractères de la page avec :

```
<meta charset="UTF-8">  
<meta charset="ISO-8859-1">
```



# Titre de la page

- Titre de la page
  - S'affichera dans l'onglet du navigateur
  - Titre utilisé par les moteurs de recherche ou autres programmes

<**titletitle**>

The screenshot shows a Mac OS X desktop environment with a web browser window open. The browser's title bar reads "w3c - Recherche Google". The address bar contains the URL "https://www.google.fr/search?q=w3c&o...". Below the address bar are "Applications" and "Bookmarks" buttons. The main content area shows the Google search interface with the query "w3c" entered into the search bar. The "Tous" tab is selected. Below the search bar, it says "Environ 69 000 000 résultats (0,36 secondes)". A search result for "The W3C Markup Validation Service" is displayed, with the URL "https://validator.w3.org/" and a "Traduire cette page" link. A brief description follows: "World Wide Web Consortium validator, maintained at W3C by W3C staff and collaborators. Checks the markup validity of Web documents in HTML, XHTML, ...". At the bottom of the result, there are links for "Valid CSS", "Valid CSS!", "W3C Validator Suite", "W3C mobileOK Checker", and "About". The footer of the browser window displays "World Wide Web Consortium (W3C)".

# Moteurs de recherche



- › Utile pour les moteurs de recherche :

- Mots clés (de moins en moins utilisés)

```
<meta name="keywords" content="HTML, HyperText Markup Language, Validation,  
W3C Markup Validation Service">
```

- Description

```
<meta name="description" content="W3C's easy-to-use  
markup validation service, based on SGML and XML parsers.">
```

- Robots (noindex : ne pas indexer la page,nofollow : ne pas suivre les liens présents dans la page)

```
<meta name="robots" content="noindex, nofollow">
```



- Dans la balise head on peut inclure un CSS présent dans un fichier externe avec la balise link

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

- Ou bien directement dans la balise style

```
<style>
  body {
    background: beige;
  }
</style>
```

# JavaScript



- Dans la balise head on peut inclure un JavaScript présent dans un fichier externe (ne pas oublier la balise </script> fermante)
- Ou bien directement dans la balise script  
`<script src="fichier.js"></script>`

```
<script>
  window.addEventListener('load', function() {
    console.log('Chargement terminé');
  });
</script>
```

# Exercices



## ▶ Exercices

- Créer un nouveau fichier style.css et le charger dans la balise HEAD

Contenu du fichier :

```
body {  
    background-color: beige;  
}
```

- Créer un nouveau fichier script.js et le charger dans la balise HEAD

Contenu du fichier :

```
console.log('Test');
```

- Ajouter un titre et une balise META description pour le référencement



**formation.tech**

# Contenu d'une page web (balise body)

# Commentaires HTML



- ▶ Pour commenter le HTML
  - Permet de décrire les balises pour un meilleure relecture par les développeurs
  - Permet de désactiver les balises, sans les supprimer
- ▶ Exemple

```
<!-- Le menu principal du site -->
<nav id="menu">
    <ul>
        <li><a href="index.html">Accueil</a></li>
        <li><a href="présentation.html">Présentation</a></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
</nav>
<!-- <p>Un paragraphe désactivé</p> -->
```

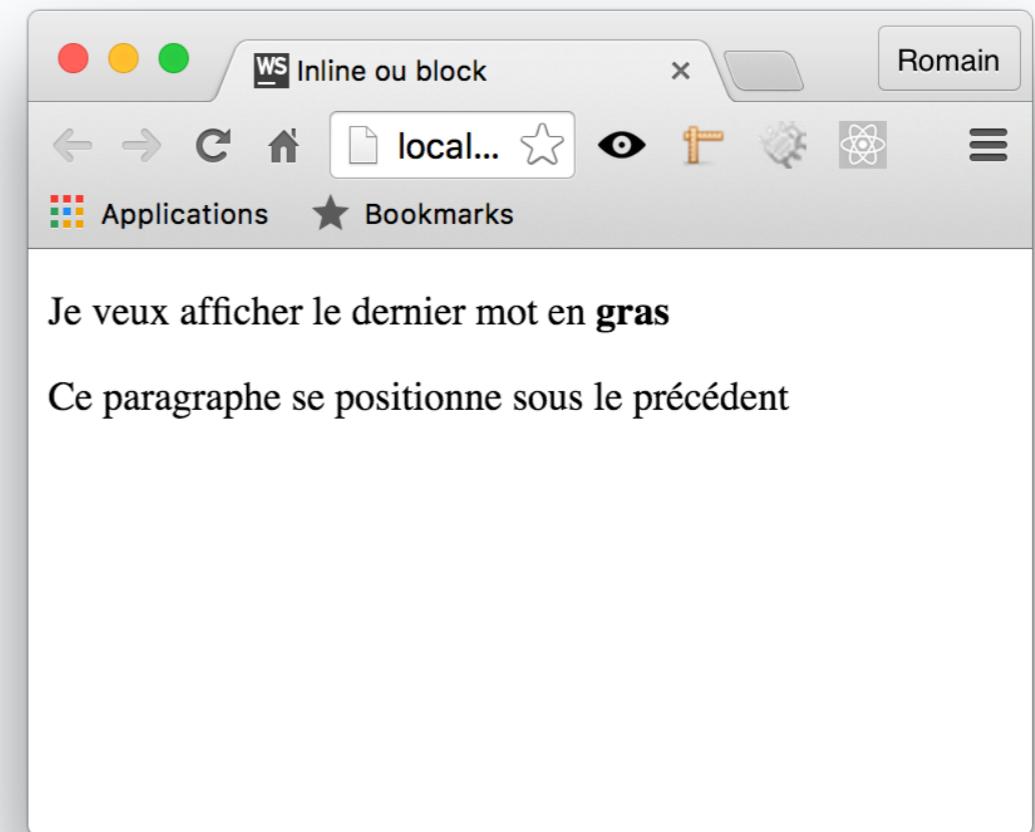


# Balises block ou inline

- En HTML il existe 2 grands types de balises :
  - Balises blocs (block), qui provoquent par défaut un retour à la ligne
  - Balises en ligne/à la file (inline), qui permettent se positionnent au sein d'un texte
- Exemple

```
<p>
    Je veux afficher le
    dernier mot en <b>gras</b>
</p>

<p>
    Ce paragraphe se
    positionne sous le précédent
</p>
```



# Balises block



- Principales balises block :

```
<address>, <article>, <aside>, <blockquote>, <canvas>, <dd>, <div>, <dl>,  
<fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1>, <h2>, <h3>,  
<h4>, <h5>, <h6>, <header>, <hgroup>, <hr>, <li>, <main>, <nav>,  
<noscript>, <ol>, <output>, <p>, <pre>, <section>, <table>, <tfoot>,  
<ul>, <video>
```

- Principales balises inline :

```
<b>, <big>, <i>, <small>, <tt>, <abbr>, <acronym>, <cite>, <code>, <dfn>,  
<em>, <kbd>, <strong>, <samp>, <time>, <var>, <a>, <bdo>, <br>, <img>,  
<map>, <object>, <q>, <span>, <sub>, <sup>, <button>, <input>, <label>,  
<select>, <textarea>
```

# Les balises de regroupement



- Pour faire des regroupements sur une page :
  - <**address**> : coordonnées de contact
  - <**article**> : un article (de blog par exemple)
  - <**aside**> : du contenu associé au contenu principal (colonne de droite, encart au milieu d'un article de presse)
  - <**div**> : regroupement pour la mise en forme (couleur de fond, positionnement)
  - <**header**> : entête (de page, d'article, d'une section quelconque)
  - <**main**> : partie principale (de page, d'article, d'une section quelconque)
  - <**footer**> : pied (de page, d'article, d'une section quelconque)
  - <**nav**> : éléments de navigation
  - <**section**> : regroupement par catégorie (exemple : introduction, présentation...)



# Autres balises bloc courantes

- ▶ Les titres : h1, h2, h3, h4, h5, h6

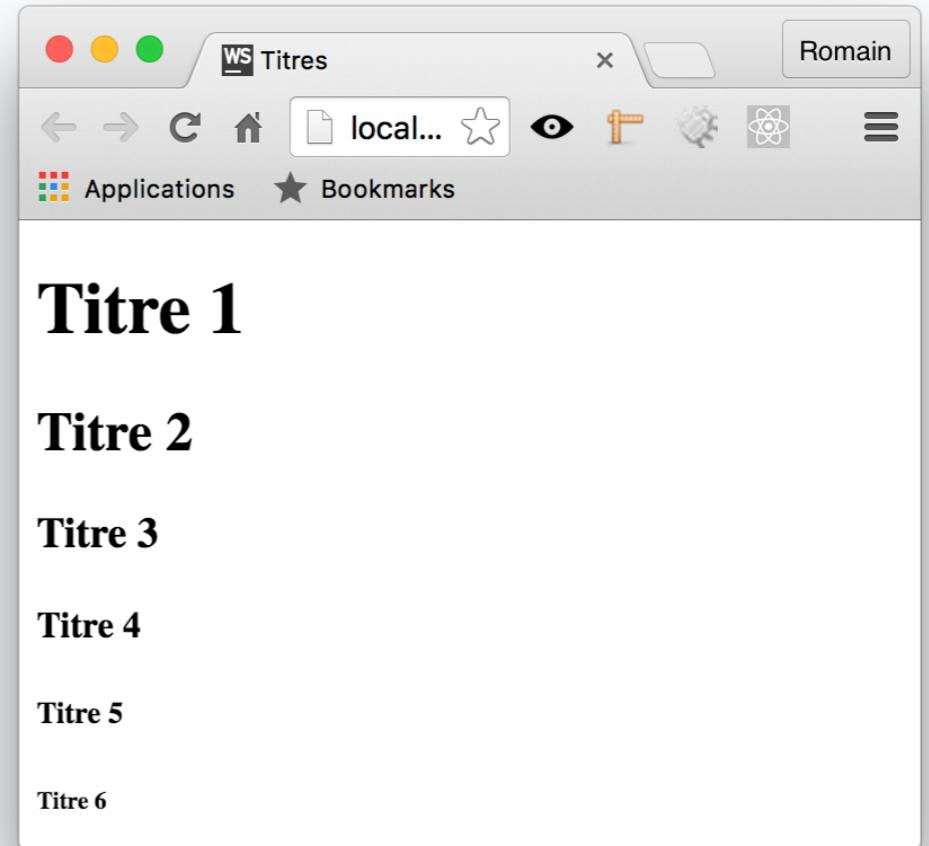
```
<h1>Titre 1</h1>
<h2>Titre 2</h2>
<h3>Titre 3</h3>
<h4>Titre 4</h4>
<h5>Titre 5</h5>
<h6>Titre 6</h6>
```

- ▶ Les paragraphes : p

```
<p>Un paragraphe</p>
```

- ▶ Un bloc de citation

```
<blockquote>Une citation</blockquote>
```





# Les listes

## ▶ Liste non ordonnées

```
<ul>
  <li>Google</li>
  <li>Microsoft</li>
  <li>Apple</li>
</ul>
```

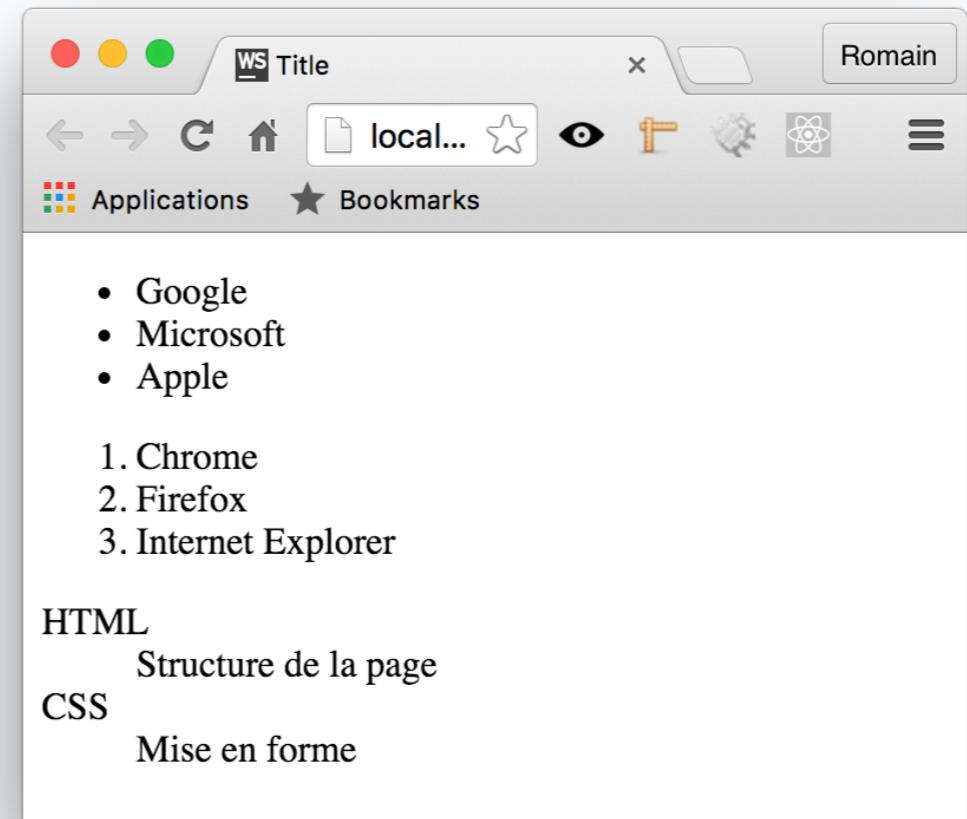
## ▶ Liste ordonnées

```
<ol>
  <li>Chrome</li>
  <li>Firefox</li>
  <li>Internet Explorer</li>
</ol>
```

## ▶ Liste de définitions

```
<dl>
  <dt>HTML</dt>
  <dd>Structure de la page</dd>

  <dt>CSS</dt>
  <dd>Mise en forme</dd>
</dl>
```

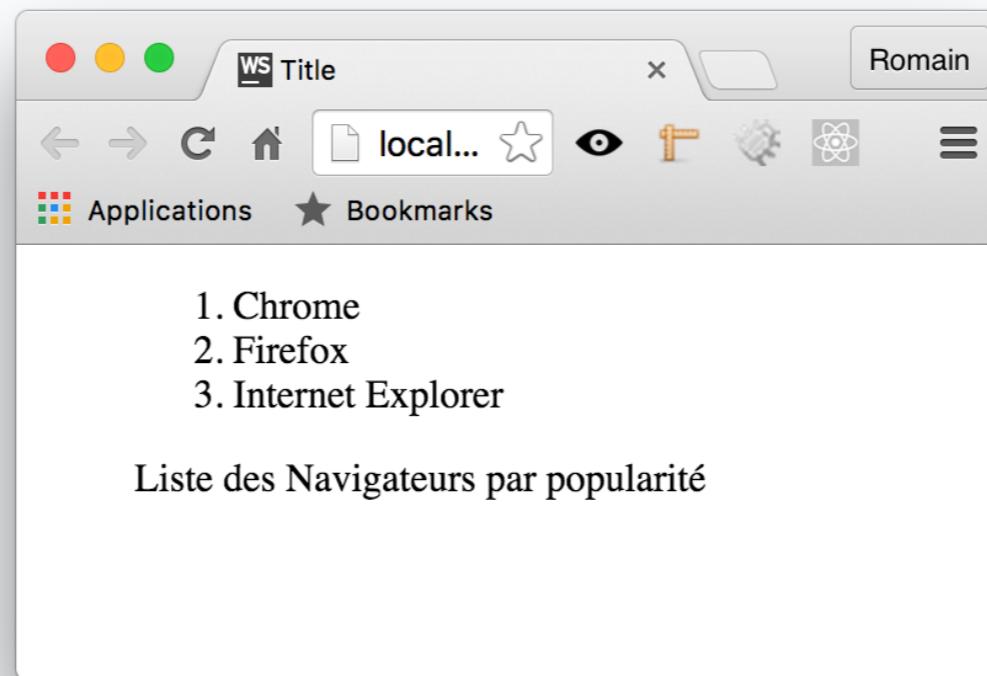




# Légende

- Mettre une légende à un élément (liste, tableau, image, vidéo...)

```
<figure>
  <ol>
    <li>Chrome</li>
    <li>Firefox</li>
    <li>Internet Explorer</li>
  </ol>
  <figcaption>Liste des Navigateurs par popularité</figcaption>
</figure>
```



# Principales balises inline



## ▶ Balises inlines

Ne provoque pas de retour à la ligne. Permettent de délimiter des mots.

b : gras, texte sans importance particulière, visuellement différent

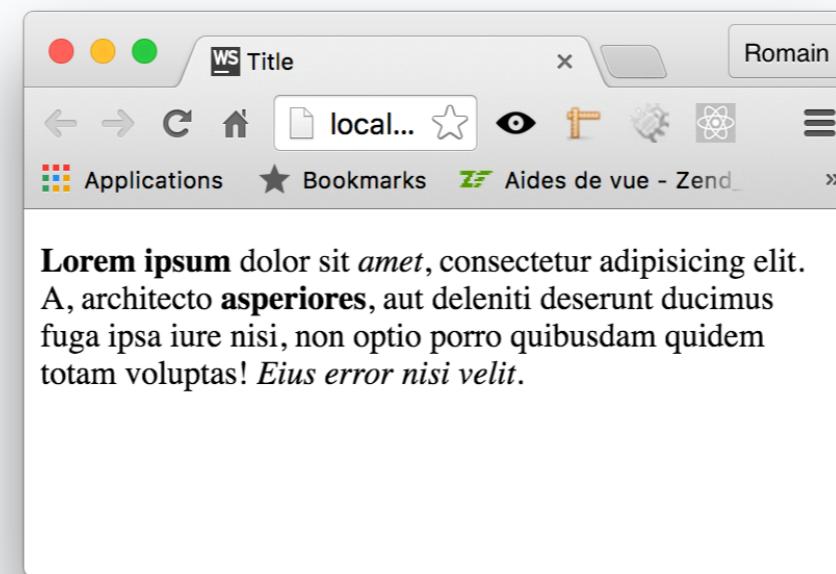
i : italique, ton différent, parole prise par un autre

em : emphase légère (importance plus grande que le reste du texte)

strong : emphase forte

span : délimitation neutre (pour le CSS ou le JS)

```
<p>
  <b>Lorem ipsum</b> dolor sit <i>amet</i>, consectetur adipisicing elit.
  A, architecto <strong>asperiores</strong>, aut deleniti deserunt ducimus
  fuga ipsa iure nisi, <span>non optio</span> porro quibusdam quidem
  totam voluptas! <em>Eius error nisi velit</em>.
</p>
```





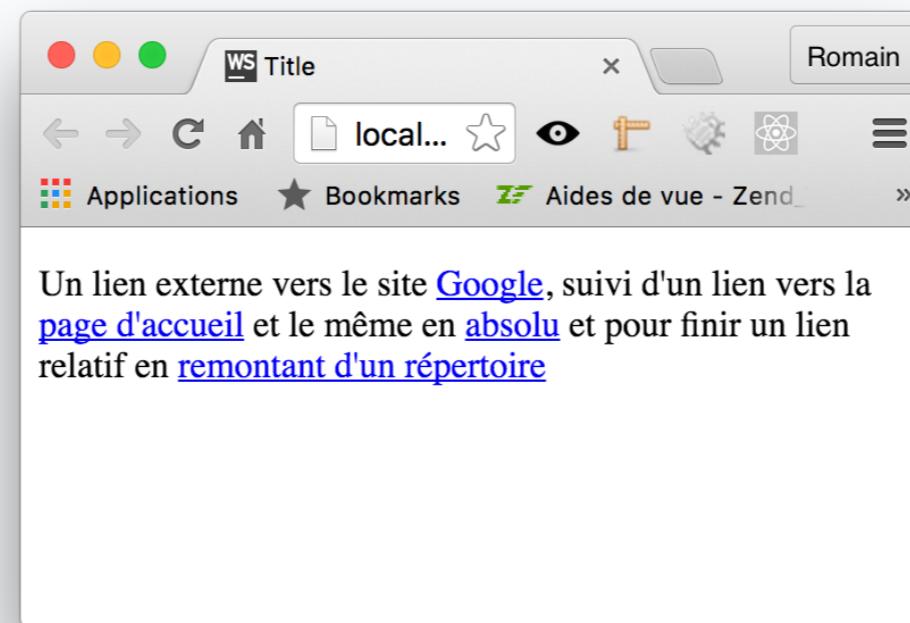
## ▶ Liens

Balise inline, la balise délimite les éléments sur lesquels porte le lien (la zone cliquable).

L'attribut href permet de définir la cible du lien. Le lien peut être :

- Externe : commence par http:// ou un autre protocole, ou // (dans le protocole courant)
- Relatif : dépend d'où se trouve le fichier (../ pour remonter d'un dossier)
- Absolu : depuis la racine du serveur (après http://domaine.ext)

```
<p>
    Un lien externe vers le site <a href="http://www.google.fr/">Google</a>, suivi d'un lien vers
    la <a href="index.html">page d'accueil</a> et le même en <a href="/mondossier/index.html">absolu</a>
    et pour finir un lien relatif en <a href="../page.html">remontant d'un répertoire</a>
</p>
```





# Exercices

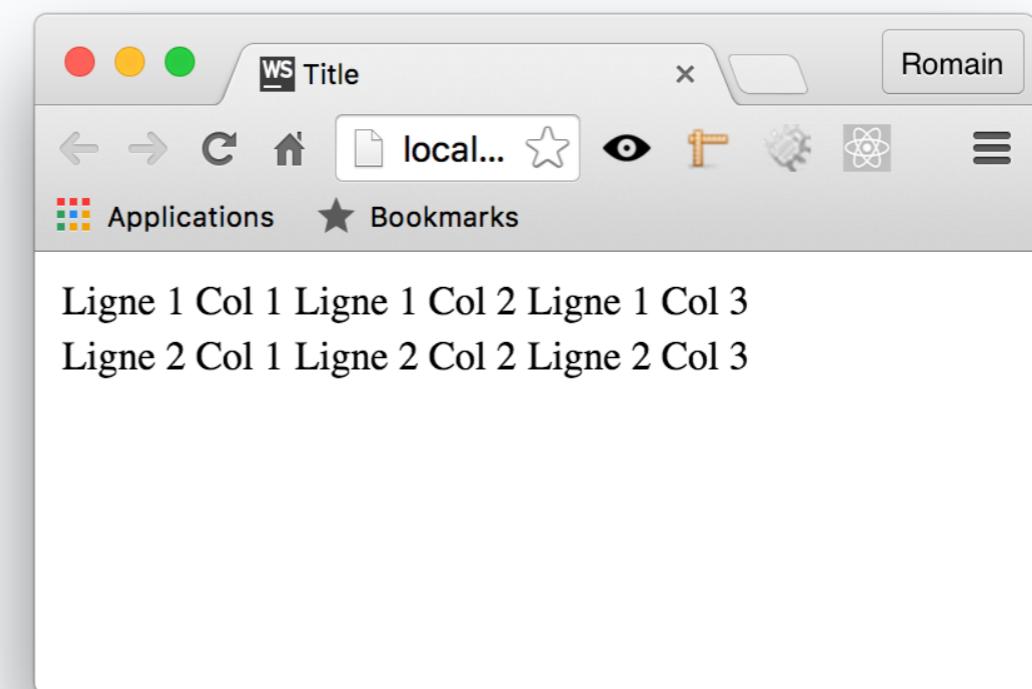
- ▶ Exercices
  - Créer une page cv.html contenant, 3 blocs :
    - Le titre « Description » suivi d'un paragraphe vous présentant
    - Le titre « Photo » suivi d'une photo
    - Le titre « Loisirs » suivi d'une liste de loisirs
  - Sur la page index.html ajouter un lien vers la page cv.html
  - Et inversement un lien depuis cv vers index



# Tableaux

- Les tableaux sont utilisés pour présenter des données par lignes ou colonnes (penser tableau Excel)
- 3 balises principales, table pour l'ensemble du tableau, tr pour les lignes et td pour les cellules

```
<table>
  <tbody>
    <tr>
      <td>Ligne 1 Col 1</td>
      <td>Ligne 1 Col 2</td>
      <td>Ligne 1 Col 3</td>
    </tr>
    <tr>
      <td>Ligne 2 Col 1</td>
      <td>Ligne 2 Col 2</td>
      <td>Ligne 2 Col 3</td>
    </tr>
  </tbody>
</table>
```





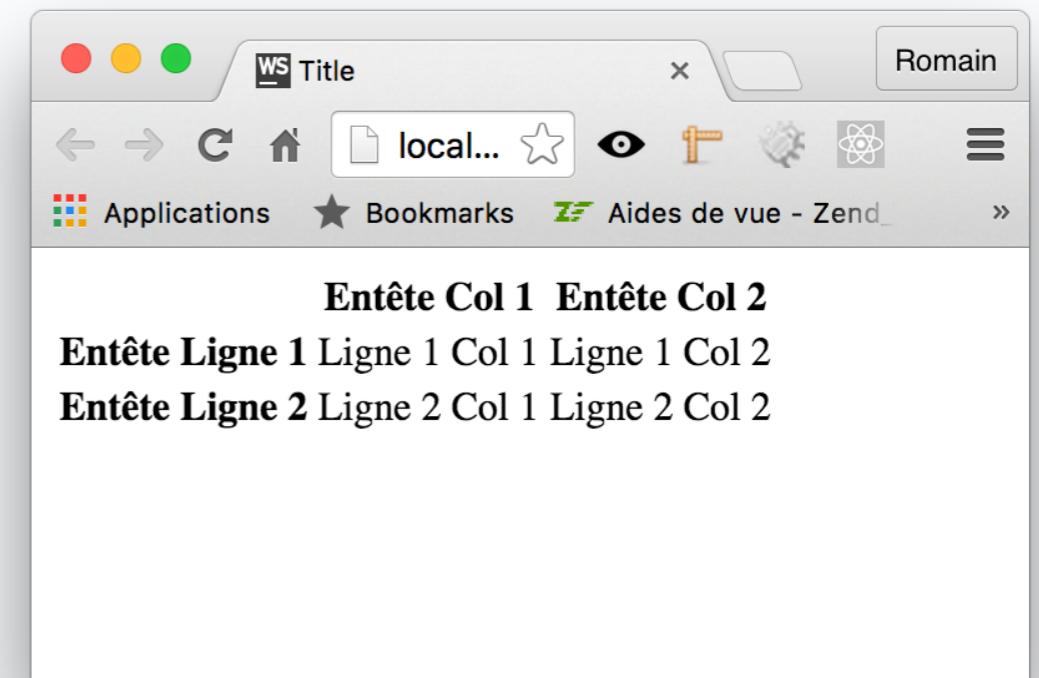
# Tableaux

## ▶ Les entêtes

Il est possible de définir des cellules d'entêtes avec la balise th

L'attribut scope permet via les valeurs « col » ou « row » de spécifier s'il s'agit respectivement d'un entête de ligne ou de colonne.

```
<table>
  <tr>
    <th></th>
    <th scope="col">Entête Col 1</th>
    <th scope="col">Entête Col 2</th>
  </tr>
  <tr>
    <th scope="row">Entête Ligne 1</th>
    <td>Ligne 1 Col 1</td>
    <td>Ligne 1 Col 2</td>
  </tr>
  <tr>
    <th scope="row">Entête Ligne 2</th>
    <td>Ligne 2 Col 1</td>
    <td>Ligne 2 Col 2</td>
  </tr>
</table>
```





# Tableaux

- Détailer un tableau

- thead, tbody, tfoot : pour regrouper les lignes d'entêtes, de corps ou de pied.
- summary et caption : légende (seule caption s'affiche)

```
<table summary="Un tableau avec des sections">
  <caption>Affiche la légende</caption>
  <thead>
    <tr>
      <th></th>
      <th scope="col">Entête Col 1</th>
      <th scope="col">Entête Col 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Entête Ligne 1</th>
      <td>Ligne 1 Col 1</td>
      <td>Ligne 1 Col 2</td>
    </tr>
    <tr>
      <th scope="row">Entête Ligne 2</th>
      <td>Ligne 2 Col 1</td>
      <td>Ligne 2 Col 2</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th scope="row">Entête Ligne 3</th>
      <td>Ligne 3 Col 1</td>
      <td>Ligne 3 Col 2</td>
    </tr>
  </tfoot>
</table>
```

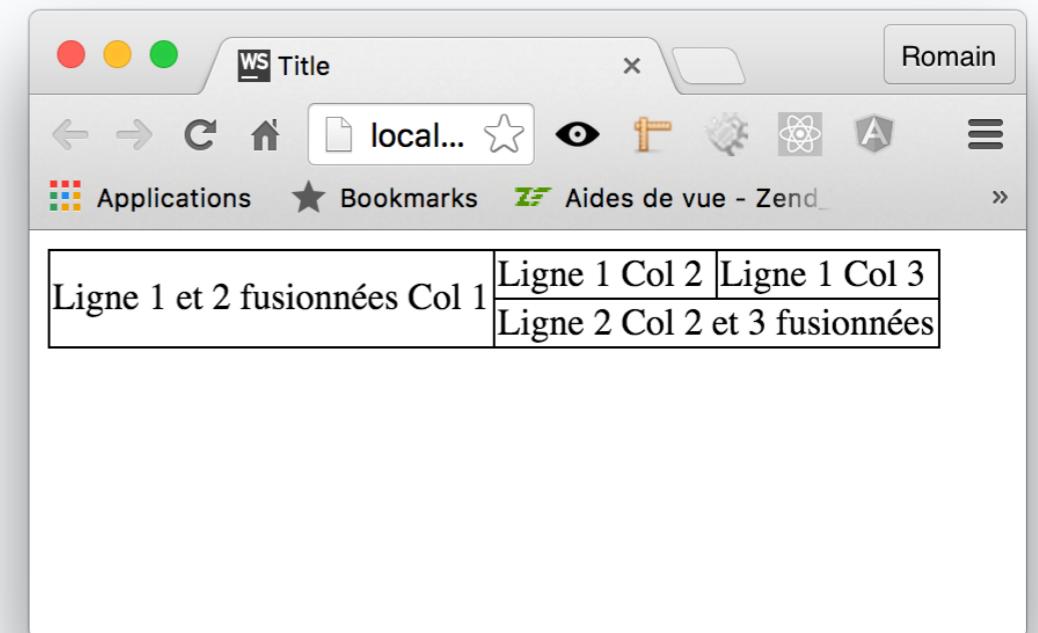




# Tableaux

- ▶ Fusionner des cellules
  - l'attribut rowspan permet de fusionner avec les cellules du dessous (le nombre indique le nombre total de cellules à fusionner)
  - l'attribut colspan permet la fusion de colonnes

```
<table>
  <tr>
    <td rowspan="2">Ligne 1 et 2
fusionnées Col 1</td>
    <td>Ligne 1 Col 2</td>
    <td>Ligne 1 Col 3</td>
  </tr>
  <tr>
    <td colspan="2">Ligne 2 Col 2
et 3 fusionnées</td>
  </tr>
</table>
```





# Exercices

- ▶ Exercices
  - Créer une page timetable.html contenant le tableau suivant :

Time Table					
Hours	Mon	Tue	Wed	Thu	Fri
	Science	Maths	Science	Maths	Arts
	Social	History	English	Social	Sports
	Lunch				
	Science	Maths	Science	Maths	Project
	Social	History	English	Social	



- ▶ Images

```

```

- ▶ Audio

```
<audio controls>
    <source src="fichier.ogg" type="audio/ogg">
    <source src="fichier.mp3" type="audio/mpeg">
        Balise audio non-supportée
</audio>
```

- ▶ Vidéos

```
<video controls>
    <source src="fichier.ogg" type="video/ogg">
    <source src="fichier.mp4" type="video/mpeg">
        Balise video non-supportée
</video>
```

[https://developer.mozilla.org/fr/docs/Web/HTML/format\\_media\\_support](https://developer.mozilla.org/fr/docs/Web/HTML/format_media_support)



# Formulaires

- Formulaires

```
<form action="page.php" method="post">
    <div>
        Prénom : <input type="text" name="prenom">
    </div>
    <div>
        <button type="submit">Valider</button>
    </div>
</form>
```

- Action : URL de la page à qui seront envoyé les données

- Method : Type de requête

GET : lorsqu'il y a un intérêt à ce que la page appelée soit mis en favoris (ex : formulaires de recherche)

POST : les données ne sont pas dans l'URL donc pas en favoris (inscription, envoi de messages, login/password...)

- Name : sur chaque champ devant être envoyé au serveur



# Formulaires

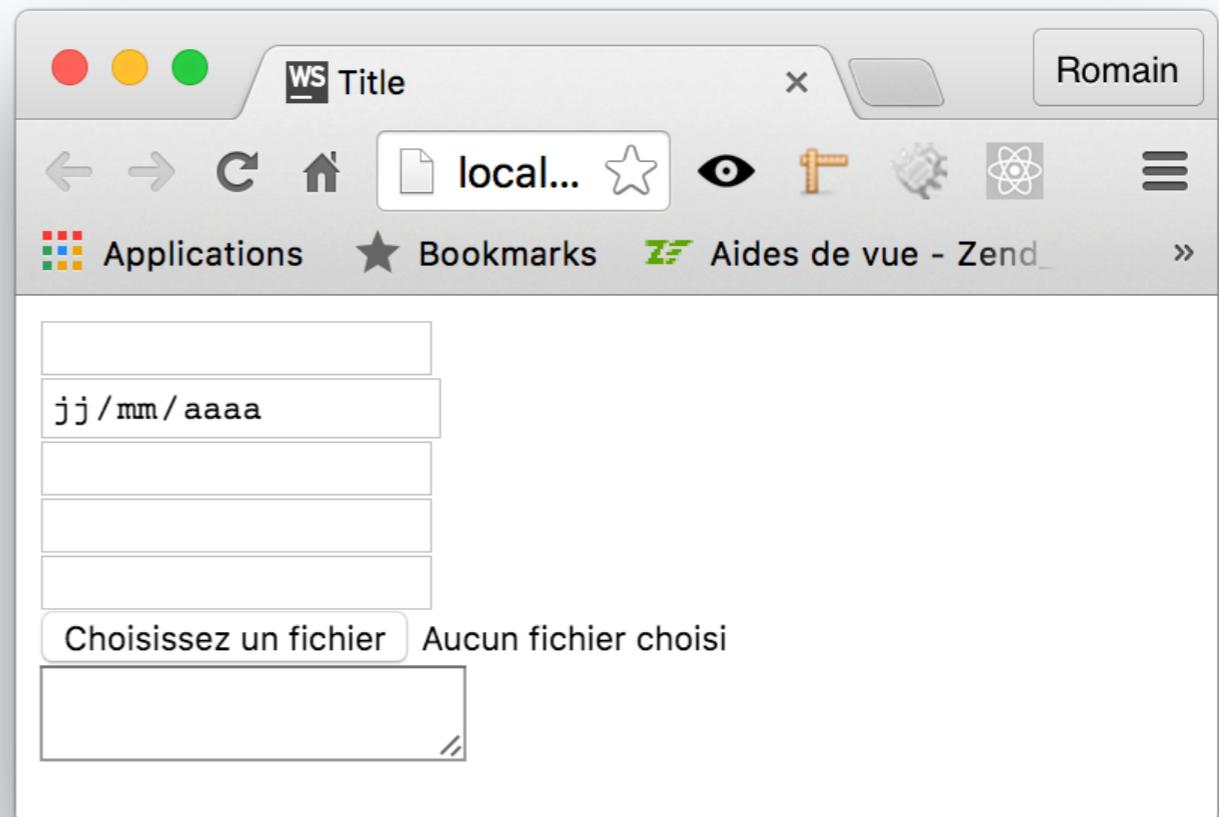
- Type de champs

```
<input type="text">
<input type="date">
<input type="email">
<input type="tel">
<input type="password">

<input type="hidden">

<input type="file">

<textarea></textarea>
```



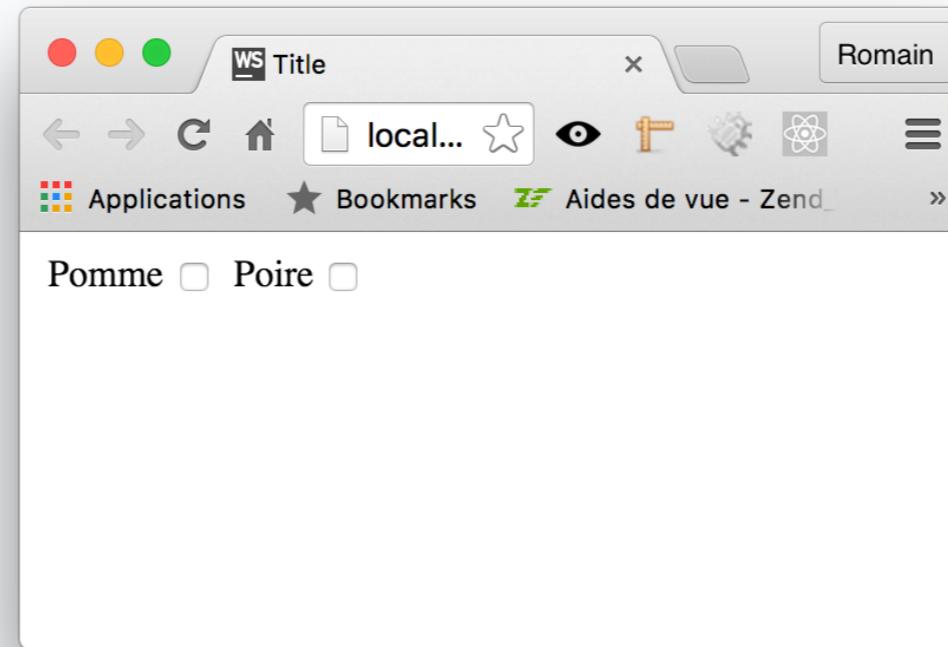


# Formulaires

- ▶ Cases à cocher

La balise label permet de rendre toute la zone cliquable

```
<label>
    Pomme
    <input type="checkbox" value="pomme" name="fruits">
</label>
<label>
    Poire
    <input type="checkbox" value="poire" name="fruits">
</label>
```

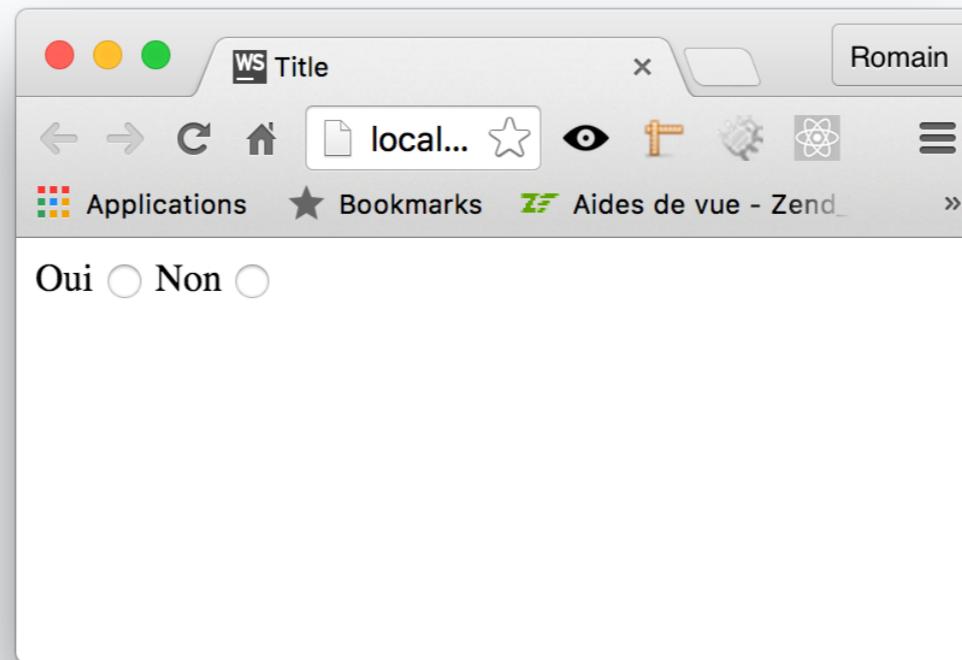




# Formulaires

- ▶ Boutons Radio

```
<label>
    Oui
    <input type="radio" value="Oui" name="OuiNon">
</label>
<label>
    Non
    <input type="radio" value="Non" name="OuiNon">
</label>
```

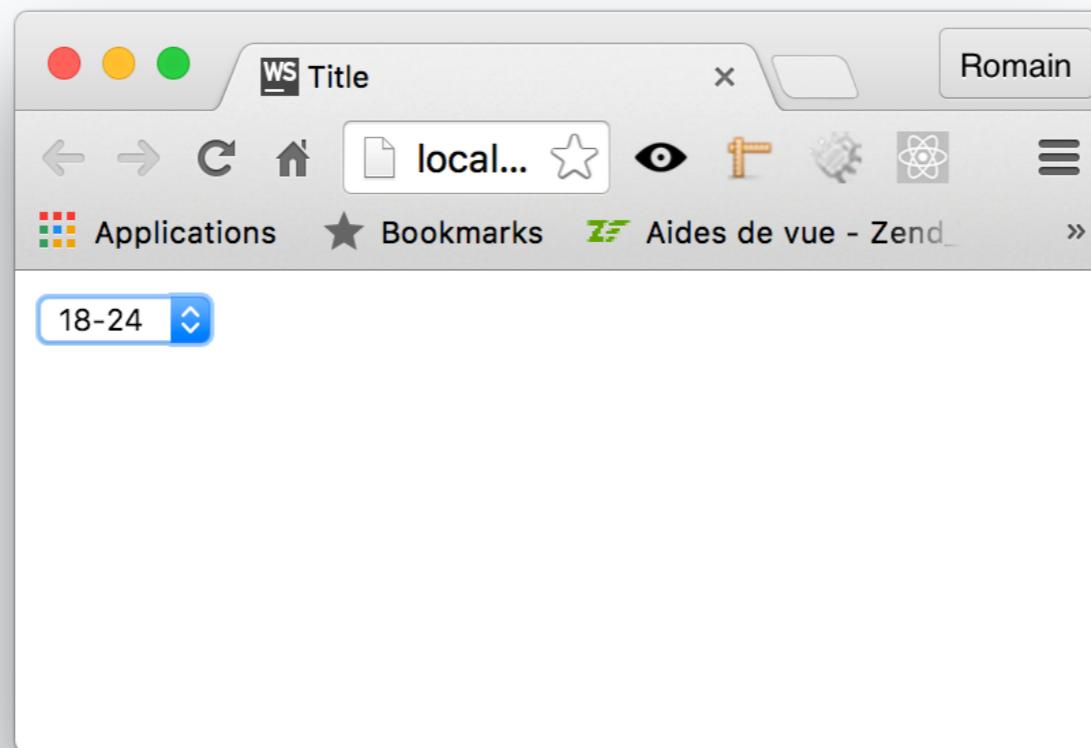




# Formulaires

- ▶ Liste déroulante

```
<select>
    <option>18-24</option>
    <option>25-29</option>
    <option>30-35</option>
</select>
```





# Attributs courants

## ▶ Identifiants

- id

L'attribut id permet d'identifier de manière unique un élément sur la page (utile pour le CSS et JS)

- class

L'attribut class permet d'identifier un groupe d'élément sur la page

```
<ul id="liste-prenoms">
    <li id="moi" class="prenom français">Romain</li>
    <li class="prenom anglais">William</li>
    <li class="prenom espagnol">Roberto</li>
</ul>
```



# Exercices

**Nom**

Prénom Nom

**Choisissez votre nom d'utilisateur**

@gmail.com

[Je préfère utiliser mon adresse e-mail actuelle](#)

**Créez un mot de passe**

**Confirmez votre mot de passe**

**Date de naissance**

Jour Mois Année

**Sexe**

Je suis...



**formation.tech**

# Accessibilité Web



# Définitions

- Accessibilité web :

L'accessibilité du web est la problématique de l'accès aux services et contenus en ligne non seulement pour les personnes handicapées et les seniors, mais aussi de manière plus générale pour tous les utilisateurs qui ne disposent pas du confort offert par un ordinateur de bureau situé dans une pièce tranquille.

Ex : navigateurs en braille, écrans de petites tailles...



# Définitions

- Normes
  - WCAG 2.0  
Norme du W3C (internationale)  
<https://www.w3.org/Translations/WCAG20-fr/>
  - Référentiel AccessiWeb 2.2  
Norme française  
[http://www.accessiweb.org/index.php/accessiweb\\_2.2\\_liste\\_generale.html](http://www.accessiweb.org/index.php/accessiweb_2.2_liste_generale.html)



# Attributs accesskey et tabindex

- Attribut accesskey  
Permet de définir une touche raccourci pour accéder à un lien, à un champs
- Attribut tabindex  
Permet de réordonner l'ordre dans lequel se fait le passage d'un lien ou d'un champ à un autre lorsqu'on appuie sur la touche TAB

```
<p><a href="lien1.htm" title="mon premier lien" tabindex="1" >lien 1 </a></p>
<p><a href="lien2.htm" title="mon deuxième lien" tabindex="2">lien 2 </a></p>
<p><a href="lien1.htm" title="mon troisième lien" tabindex="3">lien 3 </a></p>
```



# Questions à se poser

- Le langage est-il facile à comprendre ?
- La page a-t-elle un sens sans la feuille de style ?
- Les liens de la page sont-ils logiques ?
- Utilisez-vous le texte alternatif ?
- Proposez-vous une navigation au clavier efficace ?



# Exercices

- Exercices
  - Installer l'extension Opquast dans Firefox
  - Vérifier que vos pages web sont accessibles



**formation.tech**

# Bonnes Pratiques

# Bonnes Pratiques - Guides



- Dev front :
  - <https://github.com/thedaviddias/Front-End-Checklist>
  - <https://github.com/kamranahmedse/developer-roadmap>
  - <https://github.com/FrontendMasters/front-end-handbook-2018>
  - <https://developer.mozilla.org/en-US/docs/Learn>