

PUBLIER SES DONNÉES SUR LE WEB

Les fondamentaux de la création de sites web

PLAN DU COURS

Définitions

Histoire du Web

Communication sur le Web

HTML, CSS & Javascript

Content Management Systems

Publier un corpus avec Heurist

OBJECTIFS DU COURS

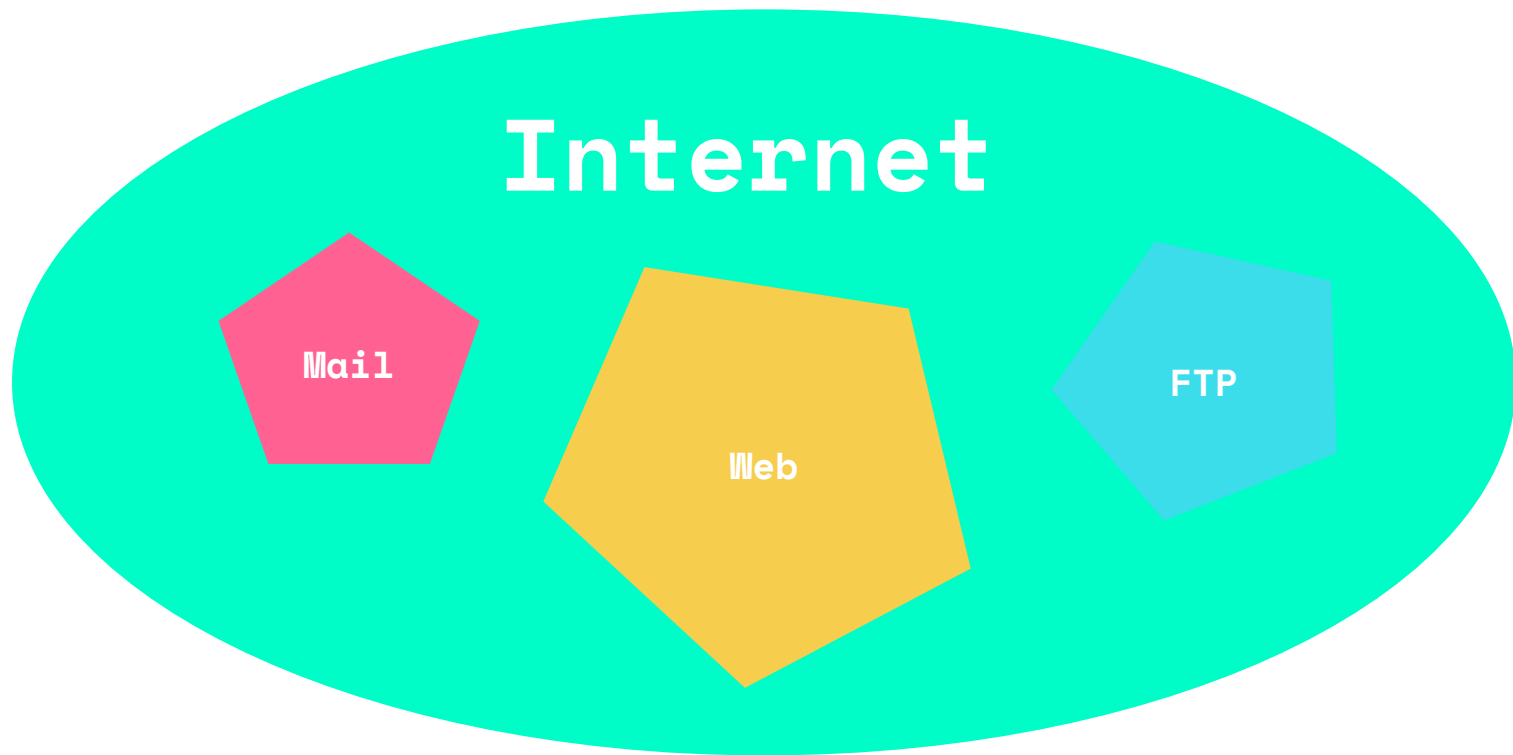
- Découvrir comment fonctionne le web
- Connaître les principes fondamentaux de la création de site web
- Donner des outils pour publier ses travaux en ligne

Dans quel but ?

- Fournir une compréhension générale des enjeux liés au Web
- Être capable de communiquer vos besoins aux ingénieurs
- Offrir des moyens de diffusion d'un corpus numérique

DÉFINITIONS

WEB \neq INTERNET



WEB ≠ INTERNET

Internet

- réseau d'infrastructures
- ensemble des machines qui communiquent entre elles
- reliées entre elles par des liens physiques

Web

- réseau d'information
- ensemble de documents dispersés sur ces machines
- reliés entre eux par des liens hypertextes

SERVEUR / CLIENT

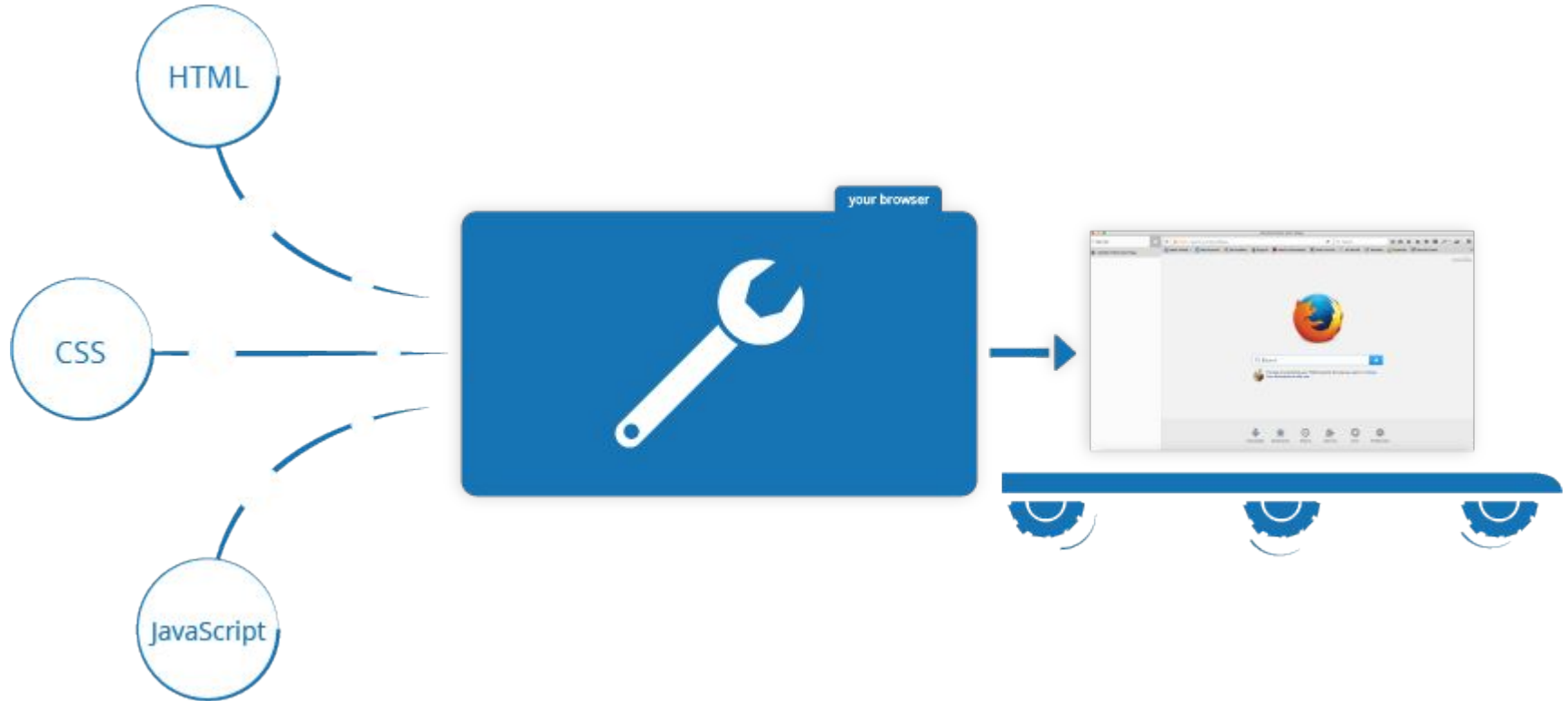
Client

- envoie des requêtes
- machine sur laquelle est exécuté le logiciel
- ordinateur personnel avec un navigateur

Serveur

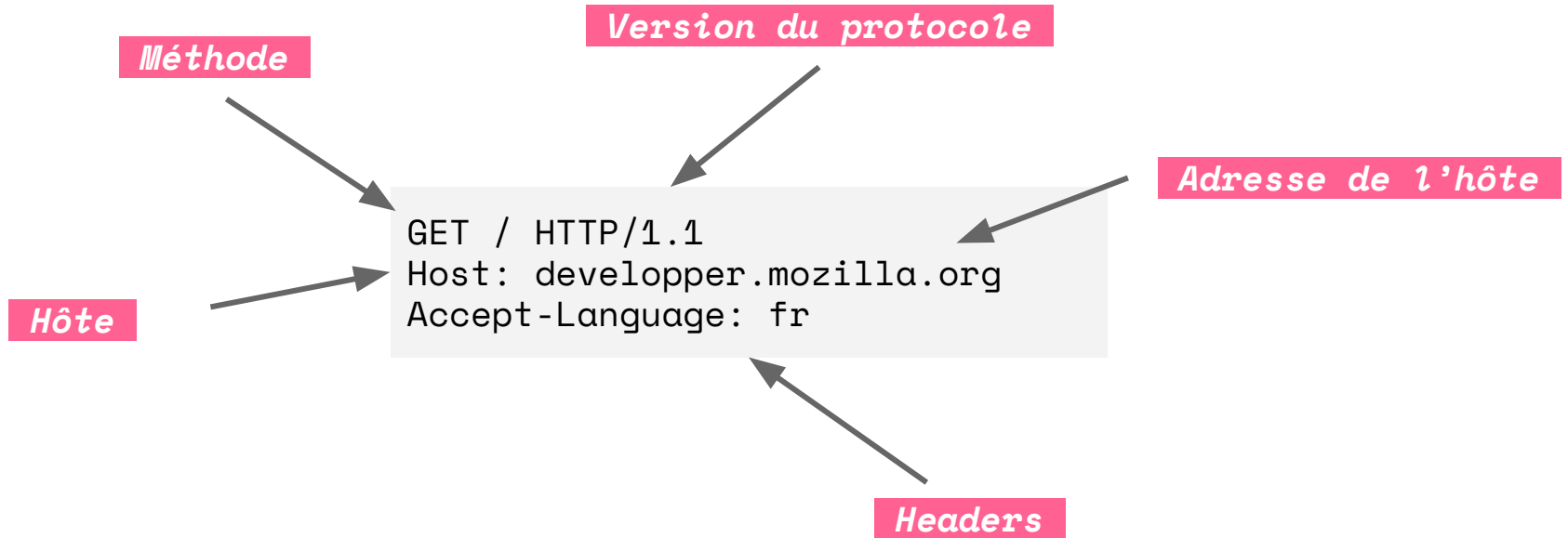
- écoute les requêtes et y répond
- machine dotée de puissance de calcul supérieur
- data center où est stocké du code

NAVIGATEUR : LOGICIEL DU WEB



PROTOCOLE HTTP

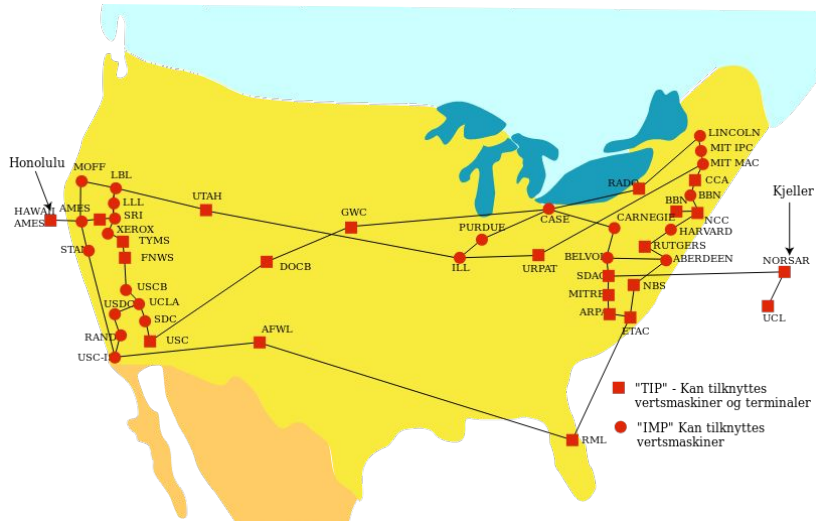
Format standardisé de requête pour communiquer sur le Web
Différents types : GET / POST / PUT / DELETE



HISTOIRE DU WEB

EN BREF

Au début des années 1970, la DARPA crée le réseau **Arpanet** dans le but de relier les différentes universités aux États-Unis.



Au début des années 80, le système de **noms de domaines** (DNS) permet de s'abstraire des adresses physiques des machines :

Les machines acquièrent un nom unique, une **URL**

EN BREF

En 1984, le CERN s'inspire d'Arpanet pour organiser son système d'information distribué entre plusieurs laboratoires.

C'est **Tim Berners-Lee** qui s'occupe de créer un protocole de communication (**HTTP : Hypertext Transfert Protocol**) pour organiser l'information de manière abordable en composant des pages reliées entre elles par des liens hypertextes.

Peu à peu, les accès deviennent publiques et la première “toile internet” est créée en **1989**.

EN BREF

Les premières versions de HTML voient le jour au début des années 1990.

En 1993, **Mosaic** devient le premier navigateur pour grand public. C'est le premier navigateur à pouvoir afficher des images et des formulaires !

Le navigateur Netscape permet quant à lui de changer la police du texte, de le faire clignoter, de l'aligner à gauche, au centre, à droite... Les avis divergent quant au rôle de HTML.

Cela mène en 1994 à la création du **W3C** par Tim Berners-Lee.



WORLD WIDE WEB CONSORTIUM

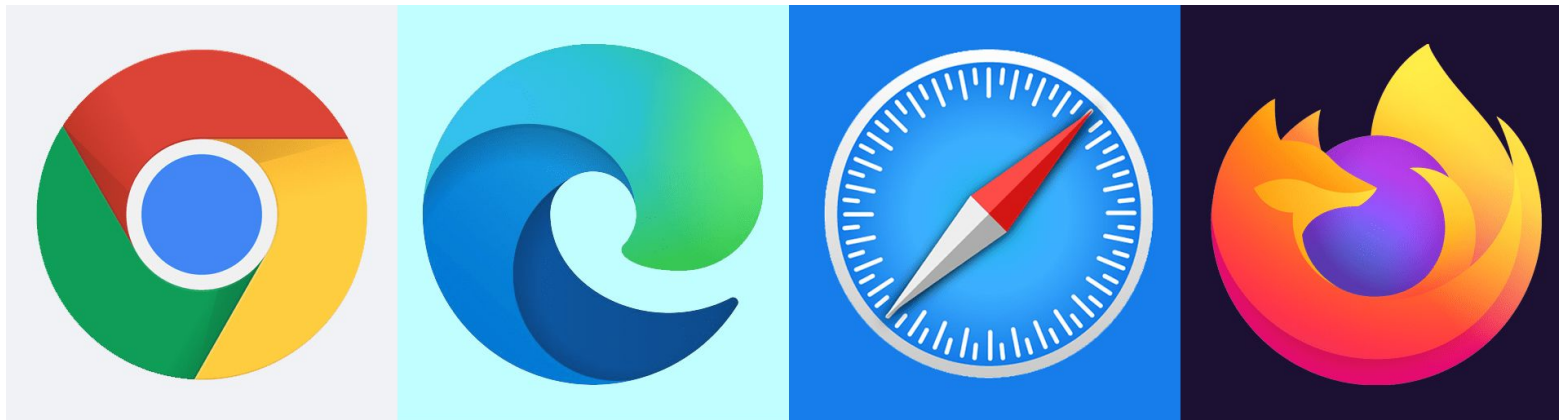
Le **W3C** est administré par trois entités : le MIT, l'université Keio, l'ERCIM en Europe.

Composé d'environ 400 organisations, il s'occupe de promouvoir l'utilisation de technologies standardisées du web, comme HTML, SVG, PNG, RDF, ...

Les standards suivent une maturation *très* rigoureuse qui passe par plusieurs étapes pour devenir une **W3C Recommendation** !

INTERPRÉTATION DU HTML

La norme HTML évolue (HTML5 en 2017) plus vite que les standards et les fonctionnalités sont adoptées par les navigateurs à des rythmes différents ce qui explique des différences de comportements d'un navigateur à l'autre

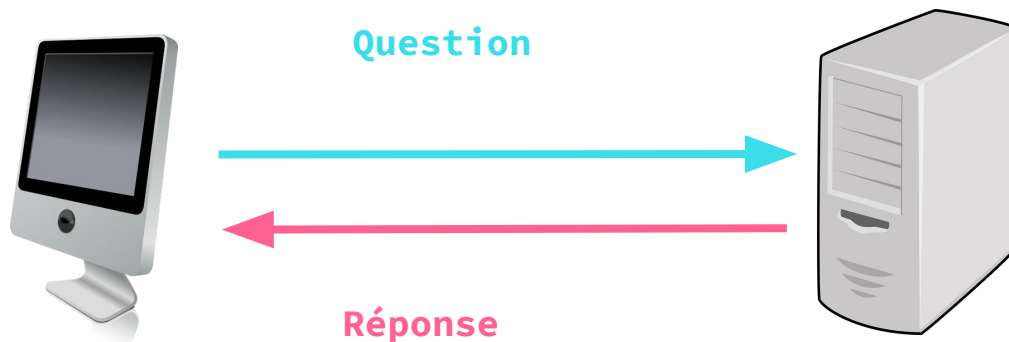


COMMUNICATION SUR LE WEB

TYPLOGIE DES SITES WEB

- Sites web **statiques** : le contenu des pages est écrit “en dur” dans les fichiers qui composent le site
- Sites web **dynamiques** : le contenu peut évoluer dynamiquement sans que leur propriétaire ait besoin de modifier les documents manuellement
- Site **web** lorsqu’il est mis à disposition sur le web
- Site **local** et il n’est accessible que sur l’ordinateur où se trouvent les documents qui le composent

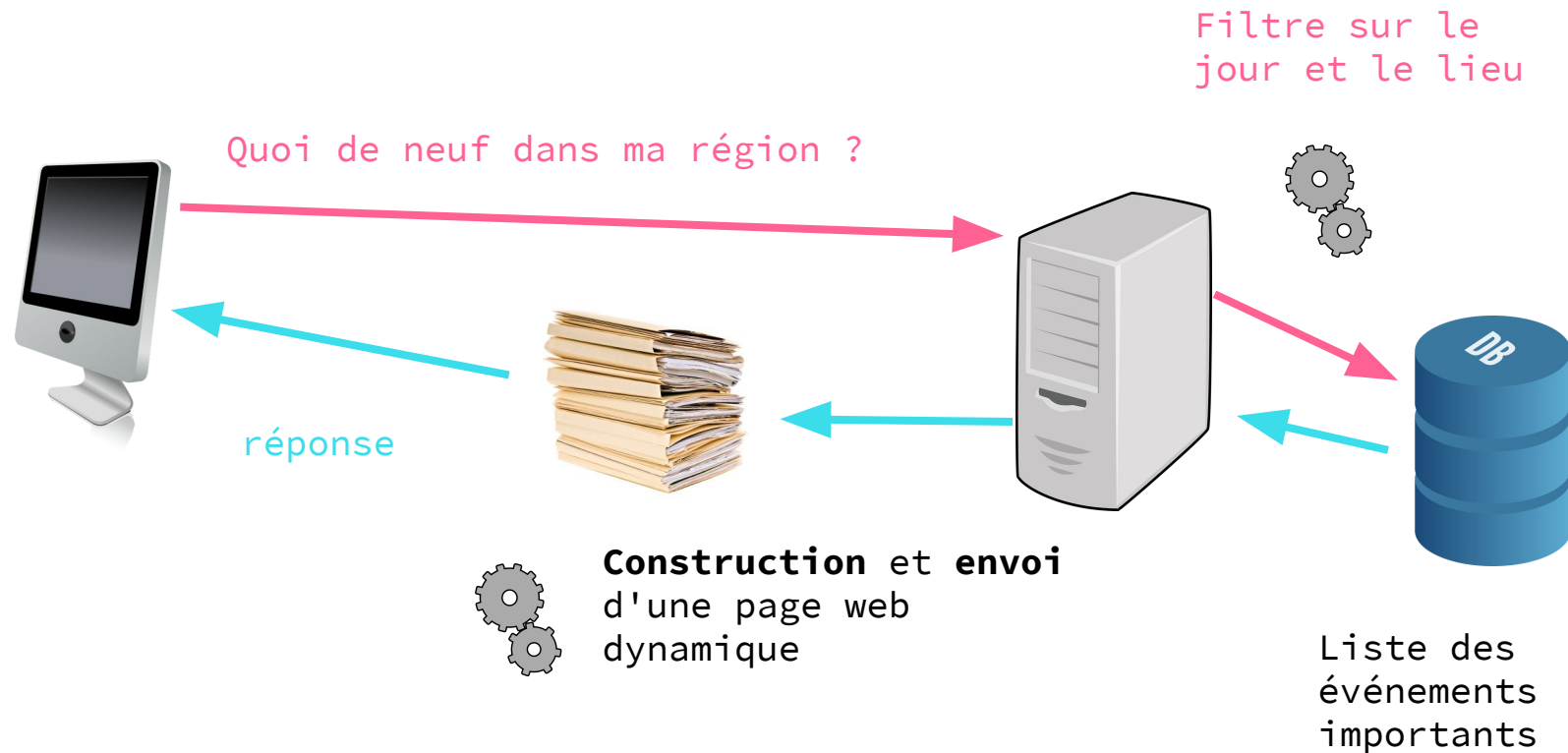
CONSULTATION DE PAGE WEB STATIQUES



Les pages statiques sont stockées dans de simples fichiers

ex: page d'accueil, à propos, contact...

CONSULTATION DE PAGE WEB DYNAMIQUES



PRODUCTION / RÉCEPTION DES DONNÉES

Il faut distinguer deux rôles dans le fonctionnement d'un site web :

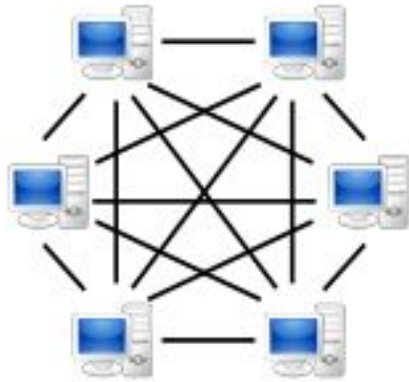
- Le **fournisseur** de données
- Le **consommateur** de données

Lorsque vous consultez une page web, vous consommez la donnée fournie par un tiers : vous êtes **consommateur** de la donnée

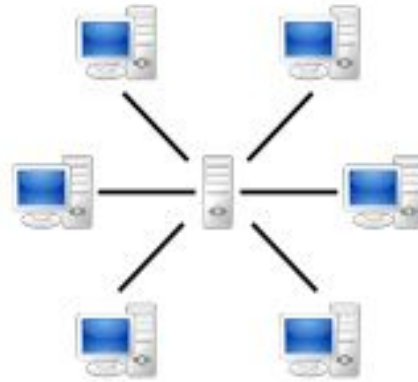
Lorsque vous remplissez un formulaire ou cliquez sur un bouton “j’aime”, vous envoyez des données sur internet : vous êtes **fournisseur** de données

ARCHITECTURE CLIENT/SERVEUR

Cette architecture à deux rôles est connue sous le nom de **“client/serveur”**



Pair à pair



Client/server

CHEMINEMENT DE LA DONNÉE

C'est le navigateur web qui se charge de contacter le serveur web. Il envoie les **requêtes** et **interprète** les réponses.

Par exemple, vous tentez de vous rendre sur le site

http://www.google.com

1. Le navigateur demande à votre Fournisseur d'Accès à Internet à quel serveur correspond l'adresse **www.google.com**
2. Le FAI répond en donnant l'adresse physique (**adresse IP** : 104.20.168.68)
3. Le navigateur envoie une requête HTTP formulant "je veux la page google"
4. Le serveur web prend connaissance de la requête et répond en HTTP
5. Le navigateur interprète la réponse comme étant une page web et l'affiche

QUEL TYPE DE DOCUMENT COMPOSENT LE WEB ?

HTML



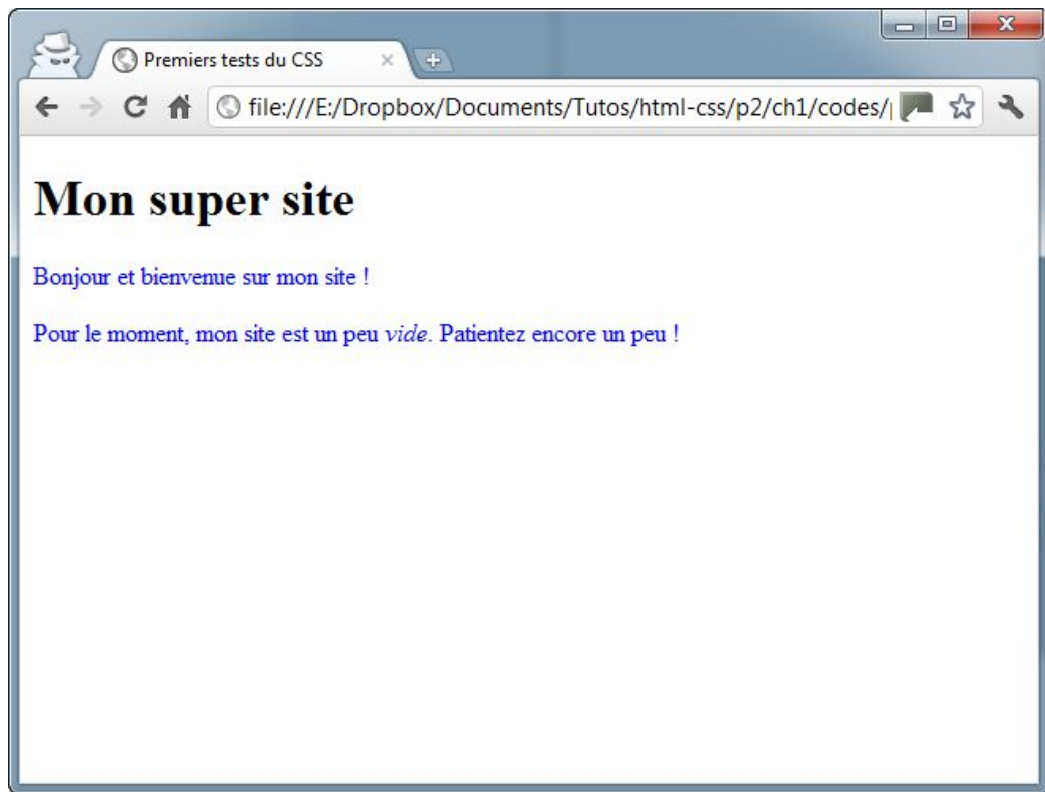
JS



CSS



OUVRIR UN FICHER HTML



HTML, CSS & JAVASCRIPT

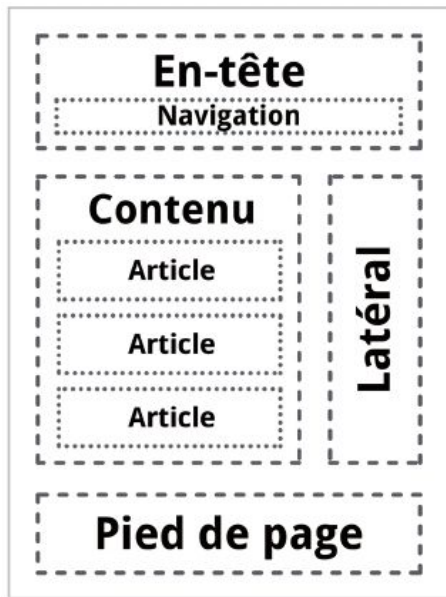
HTML

HyperText Markup Language

Constituer un contenu

HTML - CONSTITUTION D'UNE PAGE WEB

Le langage HTML décrit la structure de la page



Il décrit les **relations contenant/contenu** entre les différentes sections de la page

Il ne décrit pas le **style** (couleurs du texte, animations, polices de caractère, taille et position des blocs, etc.)

Il ne décrit pas les **logiques d'interactivité** (popup, barre de chargement, défilement automatique de photos)

HYPERTEXT MARKUP LANGUAGE

Un document HTML est un avant tout *un fichier texte*

Le HTML est langage pour *catégoriser l'information* textuelle

La structure HTML correspond à une arborescence de *noeuds imbriqués*

L'information y est entourée de *balises*

HTML : UN FICHER TEXTE

[illegible]

CATÉGORISER L'INFORMATION

→ *Structure du contenu*

blocs `<div>`, `<section>`, `<header>`, `<footer>`, ``, etc.

→ *Hiérarchie de l'information*

titre de différents niveaux `<h1>` `<h2>`, paragraphe `<p>`, etc.

→ *Ajout de sémantisme au texte*

emphase `<emph>`, mise en évidence ``, `<article>`, `<summary>`

→ *Mise en forme du texte*

italique `<i>`, gras ``, etc. ~ Usage déprécié

→ Référence externes

images ``, liens ``, etc.

→ *Ajout de métadonnées*

info de la page `<head>`, description `alt="..."` / `title="..."`

LE DOCUMENT OBJECT MODEL : IMBRICATION DE NŒUDS

```
<html>
  <head>
    <title>Nous aider </title>
    <meta charset="utf-8"/>
    <link rel="stylesheet" href="style3.css"/>
  </head>
  <header>  </img> </header>

  <nav>
    <ul class="navbar">
      <li><a href="association.html"> L'association </a>
      <li><a href="nosactions.html"> Nos actions </a>
      <li><a href="Nousaider.html"> Nous aider </a>
      <li><a href="Lestemoignages.html"> Les témoignages </a>
      <li><a href="liens.html"> Page de crédit </a>
    </ul>
  </nav>
  <body>
    <section>
      <article>

        <h1> Nous aider </h1>

        <h2> Devenir bénévole </h2>

        <p> Ludopital recherche toujours des BÉNÉVOLES : </p>
```

```
<text>
  <body>
    <p><chi rend="bold">Materials</hi></p>
    <p>Tony Harrison's poem 'An Old Score', in <chi rend="italic">Tony Harrison: Selected Poems</hi>. Harmondsworth: Penguin, 1987. Photocopy; printout of plain text.</p>
    <p>Files distributed via email: 'An Old Score' (An Old Score.txt), 'XML and TEI' (talk01.pdf), 'Create an XML file' (exercise01.pdf). The latter two from DHOXSS course 'Introduction To TEI.'</p>
    <p><chi rend="bold">Where is the text? What is encoding / markup?</hi></p>
    <list type="ordered">
      <item>Annotate the photocopy of the page as you would if you were preparing to analyse it for a seminar. Think about what features you annotate (or markup). </item>
      <item>Now look at the plain text. Note the differences between the two. Annotate the plain text with some instructions to make it look like the published version.</item>
      <item>"Only that which is explicit can be reliably processed." What are the similarities <chi rend="italic">and</hi> differences between the two sets of annotations, or what we will from now on call 'markup'? Try to look at it the way a machine might. Which are feature of presentation and which are features of meaning? Where do these overlap?</item>
      <item>Electronic markup / encoding. Brief demo of HTML.</item>
    </list>
    <p><chi rend="bold">Create an XML document</hi></p>
    <list type="ordered">
      <item>We then will follow 'exercise01.pdf' from the Oxford TEI worksheets. NB. Instead of using Wilfred Owen's 'Strange Meeting' we will be using Tony Harrison's poem 'An Old Score' (.txt and .pdf files distributed via email).</item>
      <item>Notice how structured the encoding is: how elements such as &lt;l>, &lt;lg>, &lt;h1>, &lt;h2> are nested within each-other. When I demonstrate more complex encoding we will see how these structural features can be given additional meaning.</item>
    </list>
  </body>
</text>
TEI>
```


BALISES HTML

Métadonnées
de la page

Contenu
visible de
la page

```
<!DOCTYPE html>

<html lang="fr">

  {
    <head>
      <title>Titre affiché dans le navigateur</title>
    </head>

    {
      <body>
        ... Contenu de la page proprement dit ...
      </body>
    }
  }

</html>
```


ACCÉDER AU CODE BRUT D'UNE PAGE



WIKIPÉDIA
L'encyclopédie libre

Rechercher dans Wikipédia



Seglinglin



Discussion

Brouillon

Préférences

Bêta

Liste de suivi

Contributions

Se déconnecter

Accueil

Portails thématiques

Article au hasard

Contact

Basculer vers l'ancien
habillage

Contribuer

Débuter sur Wikipédia

Aide

Communauté

Modifications récentes

Faire un don

Outils

Pages liées

Suivi des pages liées

Téléverser un fichier

Pages spéciales

Lien permanent

Informations sur la page

Citer cette page

Élément Wikidata

Imprimer / exporter

Créer un livre

Télécharger comme PDF

Article Discussion

Lire

Modifier

Modifier le code

Voir l'historique



Plus

Hypertext Markup Language

Le **Hypertext Markup Language**, généralement abrégé **HTML** ou dans sa dernière version **HTML5**, est le langage de balisage conçu pour représenter les **pages web**.

Ce langage permet :

- d'écrire de l'**hypertexte**, d'où son nom,
- de structurer **sémantiquement** la page,
- de mettre en forme le contenu,
- de créer des formulaires de saisie,
- d'inclure des **ressources multimédias** dont des **images**, des **vidéos** informatiques,
- de créer des documents **interopérables** avec des équipements traitant aux exigences de l'**accessibilité du web**.

Il est souvent utilisé conjointement avec le langage de programmation **style en cascade** (CSS). HTML est inspiré du *Standard Generalized Markup Language* d'un format ouvert.

Sommaire [masquer]

- 1 Dénominations
- 2 Évolution du langage
 - 2.1 1989-1992 : Origine

HTML
HyperText Markup Language

HTML

Caractéristiques

`.html`

`/html`

Wide Web Consortium & W3C

Langage de balisage

Standard Generalized Markup Language

`page`

Norme ISO/IEC 15445

Retour

Avancer

Actualiser

Enregistrer sous...

Imprimer...

Caster...

Envoyer sur l'appareil : e5290-alfa1

Traduire en français

AdBlock — le meilleur bloqueur de pubs

AdBlocker Ultimate

Antidote

Enregistrer dans Zotero

Afficher le code source de la page

Inspecter

Sélection

Suppression

Modification

Ajout

MANIPULER DU HTML EN DIRECT

Démo CodePen

DIV est un conteneur générique de type **bloc** : son contenu occupe tout l'espace horizontal

P (pour paragraphe) est aussi un conteneur de type bloc mais il est destiné à ne contenir que du phrasé et plus particulièrement des paragraphes, c'est pourquoi les navigateurs choisissent d'ajouter des espaces au dessus et en dessous.

Même écrits sur une seule ligne, les conteneurs de types blocs occupent tout l'espace horizontal et s'empilent donc verticalement. Pour preuve !

SPAN est un conteneur de type **inline** générique. Contrairement aux blocs qui s'étendent sur toute la ligne, les conteneurs inline **n'occupent que la place nécessaire à leur contenu**. Des éléments de ce type se suivent donc sur la même ligne.

Exemple de blocs imbriqués:

A
B
C

D E F: par défaut, la plupart des balises ne préservent pas les espaces en début et fin de bloc, comme ici.

PRE est une balise de type bloc qui permet de préserver

```
HTML
1 <!doctype html>
2 <html lang="fr">
3   <head>
4     <title>Titre de la page</title>
5     <meta charset='utf-8'>
6     <meta name="description" content="Toturiel HTML">
7     <meta name="keywords" content="HTML, CSS, JavaScript">
8     <meta name="author" content="Ségolène Albouy">
9     <link rel='stylesheet' href=''>
10    <script src=''></script>
11  </head>
12  <body>
13    <div>DIV est un conteneur générique de type
14      <strong>bloc</strong> : son contenu occupe tout l'espace
        horizontal</div>
15    <p>P (pour paragraphe) est aussi un conteneur de type
```

CSS

Cascading Style Sheets

Embellir le HTML

LES FEUILLES DE STYLE CSS : CASCADING STYLE SHEET

L'apparence d'un site web est définie par des *règles de style* formulées dans le langage CSS.

Les règles CSS sont réunies :

- Soit dans des *fichiers externes* auxquels on fait référence
- Soit dans le *code source de la page HTML* des éléments <style>

Cascading Style Sheets car les styles s'appliquent en cascadant à travers la hiérarchie du document HTML : un style appliqué à une balise s'appliquera à toutes les autres balises contenues à l'intérieur

LES SÉLECTEURS & LES PROPRIÉTÉS

```
strong, em, div, p {  
    color: blue;  
}  
p {  
    background-color: red;  
}  
.class {  
    font-family: Impact;  
}  
#id {  
    margin: 10px;  
    font-style: italic;  
}
```

MODIFIER L'APPARENCE D'UNE PAGE

Contact

Basculer vers l'ancien habillage

Contribuer

Développer sur Wikipédia

Aide

Communauté

Modifications récentes

Faire un don

Outils

Pages liées

Suivi des pages liées

Téléverser un fichier

Pages spéciales

Lien permanent

Informations sur la page

Citer cette page

Éléments Wikidata

Imprimer / exporter

Charger une liste

Hypertext Markup Language

Le ***Hypertext Markup Language***, généralement abrégé **HTML** ou dans sa dernière version **HTML5**, est le langage de balisage conçu pour représenter les pages web.

Ce langage permet :

- d'écrire de l'*hypertexte*, d'où son nom,
- de structurer sémantiquement la page,
- de mettre en forme le contenu,
- de créer des formulaires de saisie,
- d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques,
- et de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (*CSS*).
HTML est inspiré du *Standard Generalized Markup Language* (*SGML*). Il s'agit d'un format ouvert.

Sommaire [masquer]

1 Dénominations

2 Évolution du langage

2.1 1989-1998 : Origine

HTML

Caractéristiques

| | |
|-------------------|--|
| Extensions | .html .htm |
| Type MIME | text/html |
| Développé par | World Wide Web Consortium & WHATWG |
| Versions initiale | 1998 |
| Type de format | Langage de balisage |
| Basé sur | Standard Generalized Markup Language |
| Origine de | XHTML |
| Norme | ISO/IEC 15445 |

Éléments

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

AdBlock

Adblock Plus

Ad-Blocker

Augury

```
<!DOCTYPE html>
<html class="client-js vector-animations-ready ve-available" lang="fr" dir="ltr">
<head></head>
<body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject mw-editable page-Hypertext_Markup_Language rootpage-Hypertext_Markup_Language skin-vector action-view skin-vector-max-width skin-vector-search-header" style="" => $0
<div class="mw-page-container">
  <a class="mw-jump-link" href="#content">Aller au contenu</a>
  <div class="mw-page-container-inner">
    <input type="checkbox" id="mw-sidebar-checkbox" class="mw-checkbox-hack-checkbox" checked>
    <header class="mw-header"></header>
    <div class="mw-workspace-container">
      <div id="mw-panel" class="mw-sidebar"></div>
      <div id="mw-navigation"></div>
      <div class="mw-content-container">
```

Styles Computed Layout Event Listeners >>

Filter :how .cl

font-family: sans-serif; background-color: yellow;

body { background-color: #f0f0f0; color: #202122; overflow-y: scroll; font-family: 'Brush Script MT', cursive; background-color: #burlywood important;

html, body { load.php?la=...

MANIPULER LA CSS EN DIRECT

Démo CodePen

DIV est un conteneur générique de type **bloc** : son contenu occupe tout l'espace horizontal

P (pour paragraphe) est aussi un conteneur de type bloc mais il est destiné à ne contenir que du phrasé et plus particulièrement des paragraphes, c'est pourquoi les navigateurs choisissent d'ajouter des espaces au dessus et en dessous.

Même écrits sur une seule ligne, les conteneurs de types blocs occupent tout l'espace horizontal et s'empilent donc verticalement. Pour preuve !

SPAN est un conteneur de type **inline** générique. Contrairement aux blocs qui s'étendent sur toute la ligne, les conteneurs inline **n'occupent que la place nécessaire à leur contenu**. Des éléments de ce type se suivent donc sur la même ligne.

Exemple de blocs imbriqués:

A
B
C

D E F: par défaut, la plupart des balises ne préservent pas les espaces en début et fin de bloc, comme ici.

PRE est une balise de type bloc qui permet de préserver

```
HTML
1 <!doctype html>
2 <html lang="fr">
3   <head>
4     <title>Titre de la page</title>
5     <meta charset='utf-8'>
6     <meta name="description" content="Toturiel HTML">
7     <meta name="keywords" content="HTML, CSS, JavaScript">
8     <meta name="author" content="Ségolène Albouy">
9     <link rel='stylesheet' href=''>
10    <script src=''></script>
11  </head>
12  <body>
13    <div>DIV est un conteneur générique de type
      <strong>bloc</strong> : son contenu occupe tout l'espace
      horizontal</div>
14    <p>P (pour paragraphe) est aussi un conteneur de type
```

```
CSS
JS
```

JAVASCRIPT

Langage de programmation

Interagir avec la page

LANGAGE DE PROGRAMMATION

Comme Python, c'est un **langage impératif** : il définit une suite d'instruction.

C'est un langage **exécuté sur l'ordinateur du client** (*front end*)

Le **navigateur** constitue l'environnement d'exécution du JavaScript

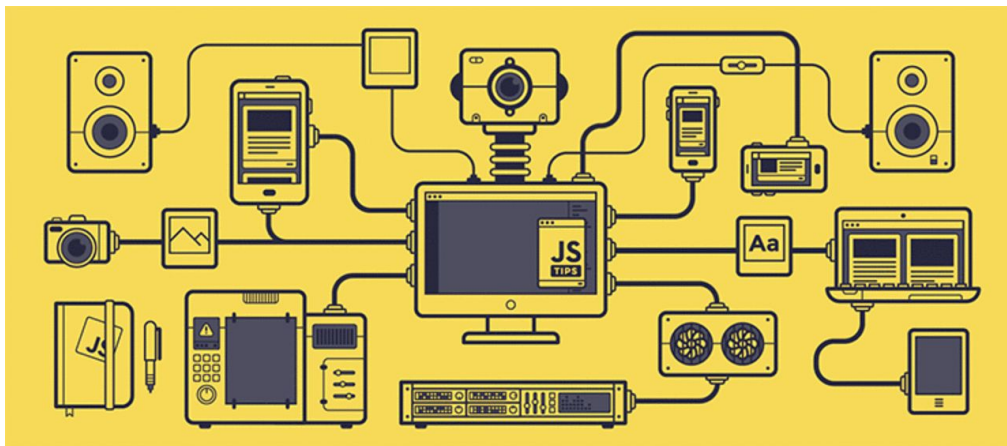
Comme pour de nombreux langages de programmation, il permet de déclarer des **variables**, des **fonctions**, de définir des **conditions**, des **boucles**.

Il permet l'interaction entre l'utilisateur et la page Web : la notion d'**événement** est au centre de JavaScript (click, réception de données, soumission de formulaires, etc.)

AGIR SUR LA PAGE PROGRAMMATIQUEMENT

Faire des opérations arithmétiques

Afficher des messages



Manipuler le DOM

Capter les flux de données

CODER EN JAVASCRIPT EN DIRECT

Démo CodePen

DIV est un conteneur générique de type **bloc** : son contenu occupe tout l'espace horizontal

P (pour paragraphe) est aussi un conteneur de type bloc mais il est destiné à ne contenir que du phrasé et plus particulièrement des paragraphes, c'est pourquoi les navigateurs choisissent d'ajouter des espaces au dessus et en dessous.

Même écrits sur une seule ligne, les conteneurs de types blocs occupent tout l'espace horizontal et s'empilent donc verticalement. Pour preuve !

SPAN est un conteneur de type **inline** générique. Contrairement aux blocs qui s'étendent sur toute la ligne, les conteneurs inline **n'occupent que la place nécessaire à leur contenu**. Des éléments de ce type se suivent donc sur la même ligne.

Exemple de blocs imbriqués:

A
B
C

D E F: par défaut, la plupart des balises ne préservent pas les espaces en début et fin de bloc, comme ici.

PRE est une balise de type bloc qui permet de préserver

```
HTML
1 <!doctype html>
2 <html lang="fr">
3   <head>
4     <title>Titre de la page</title>
5     <meta charset='utf-8'>
6     <meta name="description" content="Toturiel HTML">
7     <meta name="keywords" content="HTML, CSS, JavaScript">
8     <meta name="author" content="Ségolène Albouy">
9     <link rel='stylesheet' href=''>
10    <script src=''></script>
11  </head>
12  <body>
13    <div>DIV est un conteneur générique de type
      <strong>bloc</strong> : son contenu occupe tout l'espace
      horizontal</div>
14    <p>P (pour paragraphe) est aussi un conteneur de type
```

```
CSS
JS
```

CONTENT MANAGEMENT SYSTEMS

AIDER À LA CRÉATION DE SITES DYNAMIQUES



WORDPRESS

WORDPRESS



CMS le plus utilisé

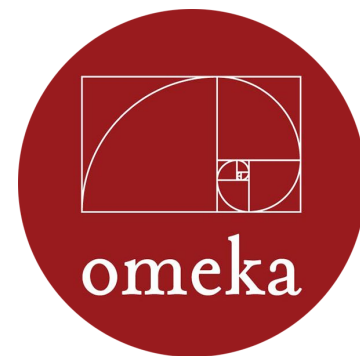
- libre, gratuit, open source
- multiples extensions, plugins, thèmes
- ~30% du Web

Polyvalent

- adapté à de nombreux types de site Web
- site vitrine, blog, vente en ligne, etc.
- wordpress.org / wordpress.com (blogging)

OMEKA

OMEKA



Open source

- développé par Center for History and New Media
- même initiative que Zotero ou Tropy
- environnement Sandbox disponible

Destiné au patrimoine

- Omeka S (recherche) et Omeka Classic (bibliothèques)
- Orienté gestion de collection
- Exposition virtuelle, bibliothèque numérique, etc.
- exemples

(APPARTÉ)

GITHUB PAGES

METTRE EN LIGNE DU HTML SIMPLEMENT

1. Créer un *repository* nommé exactement comme votre *username*
2. Ajouter un fichier `index.html`
3. (ajouter des fichiers *html*, *css* et *javascript*)
4. Commit
5. C'est en ligne

→ **`https://username.github.io.`**



PUBLICATION WEB
AVEC HEURIST

ALLER PLUS LOIN

Créer une page web en direct :

codepen.io/pen

Découvrir le HTML :

developer.mozilla.org/fr/docs/Web/HTML

Sur la sémantique des éléments :

diveinto.html5doctor.com/semantics.html

Communications sur internet :

interstices.info/auteur/laurent-viennot
