

# Développement Web - HTML -

**Groupe des étudiants : CIR1**

Maxime Lorentz

remerciement Ayoub KARINE



# Qui suis-je ?

## Parcours professionnel

**2012 - Actuellement**

Développeur front indépendant  
UX/UI designer  
Mes clients : TPE - Multinationale

**2005 - 2011**

Concepteur multimédia (Hippocampe) à Brest

**2004 - 2005**

Concepteur multimédia à Luxembourg

**2001 - 2004**

Concepteur multimédia à Baden-Baden (Allemagne)

# Informations pratiques

- Matières à enseigner à l'ISEN
  - CIR1 : Développement web
- Volume horaire : 60h
  - 30h pour HTML/XML → Semestre 1
  - 30h pour JS → Semestre 2
  - 2h/semaine
- DS et examen :
  - 1 DS + 1 examen en Janvier → Semestre 1
  - 1 DS + 1 examen en Juin → Semestre 2
  - Quelques TPs seront aussi notés et intégrés dans la note du DS
  - Note finale =  $\frac{1}{2}$  DS +  $\frac{1}{2}$  examens

# Quelques conseils

- Posez le maximum de questions
  - oralement
  - via messagerie
- Refaites les exercices à la maison



- Travaillez en groupe

# Plan

- Généralités
  - Internet
  - Web
  - Front-end VS Back-end VS Full-stack
- HTML
  - Développement Front-end
  - Définition et versions
  - Balises HTML

# Plan

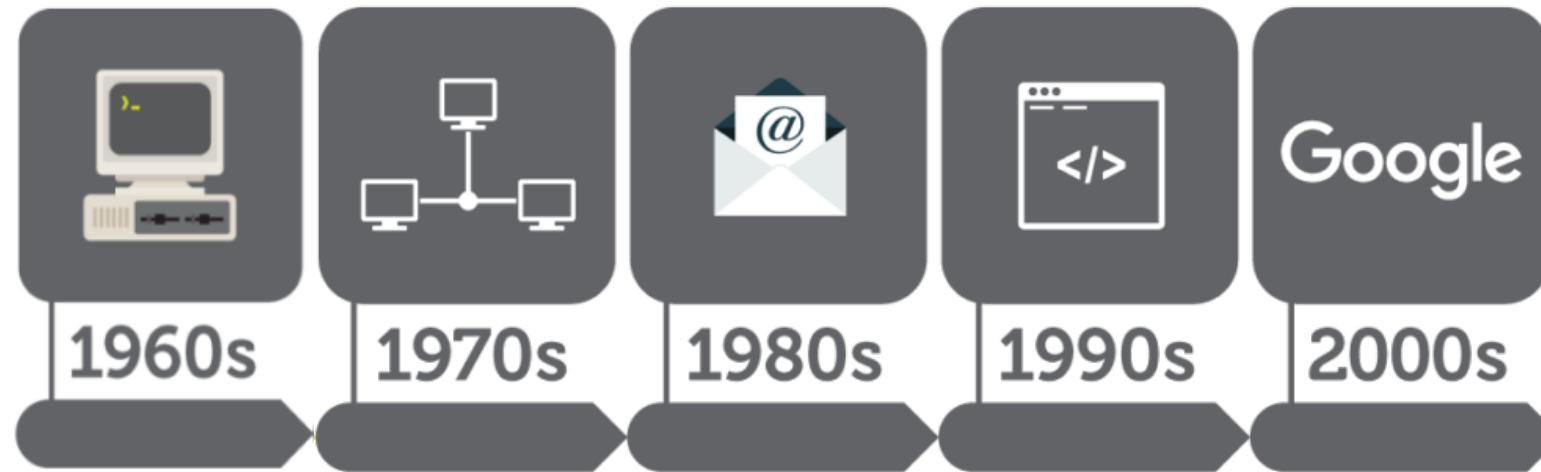
- Généralités
  - Internet
  - Web
  - Front-end VS Back-end VS Full-stack
- HTML
  - Développement Front-end
  - Définition et versions
  - Balises HTML

## Internet : Définition

- internetting
- C'est un réseau de réseaux
- Il s'agit d'un grand réseau informatique mondial accessible au public et qui permet de connecter plusieurs équipements
- Internet n'appartient à aucun organisme, c'est un travail collaboratif gouverné par des standards (W3C) et des règles



# Internet : Histoire



## Internet aujourd'hui :

- Un milliard de sites en ligne
- Trois milliards d'internaute



# Internet VS Intranet VS Extranet

## ■ Internet

- Un réseau mondial accessible au public

## ■ Intranet

- Un réseau privé interne accessible aux membres d'une organisation

## ■ Extranet

- Un réseau semi-privé accessible aux membres d'une organisation et à des utilisateurs spécifiques de l'extérieur

## Internet ≠ Web

- Les machines communiquent via internet à travers des protocoles de transfert de données standard
- Internet a plusieurs applications :
  - courrier électronique
  - messagerie instantanée
  - téléchargement de fichiers
  - **World Wide Web (WWW) ou WEB tout court**
  - ...
- On se connecte au web via un accès internet qui peut être obtenu par :
  - réseau filaire
  - réseau sans fil
  - ...

# Web

- Le **Web** permet de consulter des pages accessibles sur des sites. C'est une application répartie dans plusieurs appareils:
  - un éditeur web pour **créer** la page web → tâche réalisée par le programmeur
  - un serveur web pour **stocker** la page web → tâche réalisée par le programmeur
  - un navigateur web pour **consulter** la page web → tâche réalisée par l'internaute

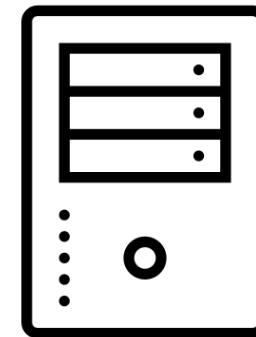
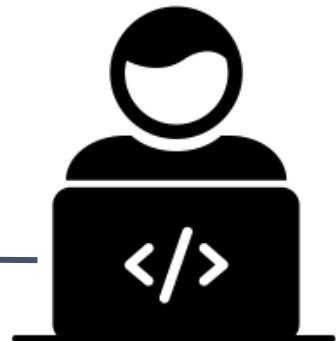
Internaute



Navigateurs



Programmeur

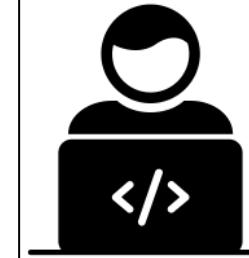


Serveur  
web

# Web : mode de fonctionnement

1. Le client envoie une requête au serveur web via internet
2. Le serveur cherche la page demandée
3. Le navigateur interprète la réponse reçue et affiche la page HTML

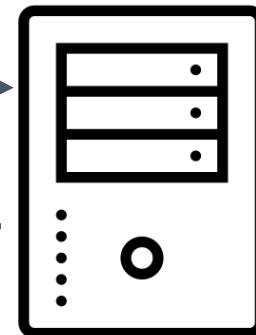
Internaute/Client



```
bonjour.html
<html>
  <body>
    <b> bonjour</b>
  </body>
</html>
```

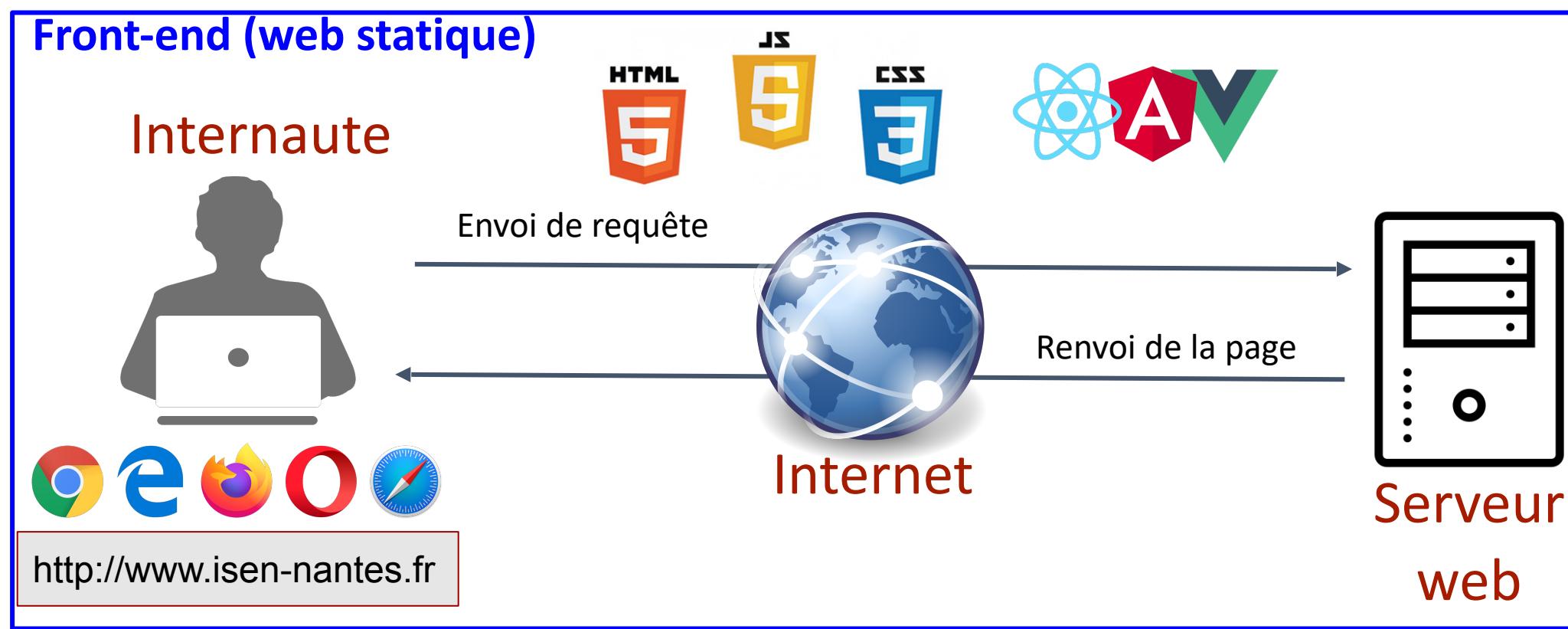


Internet

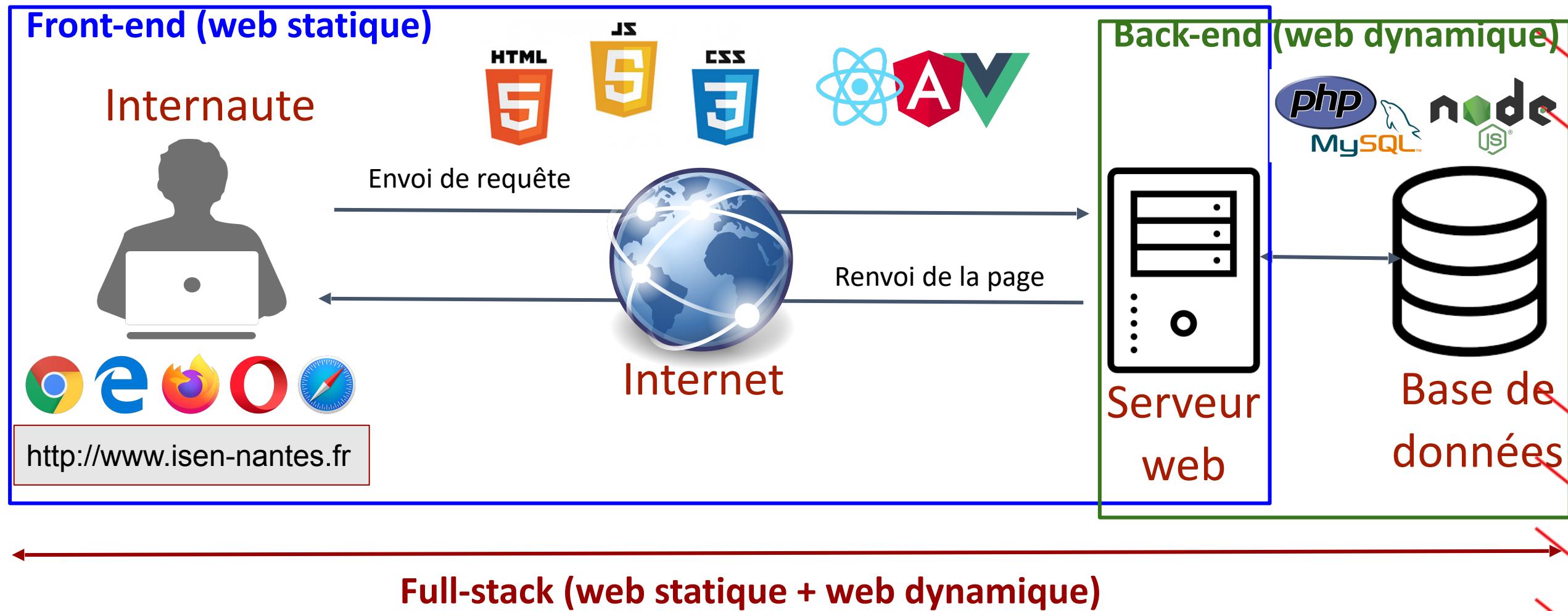


Serveur  
web

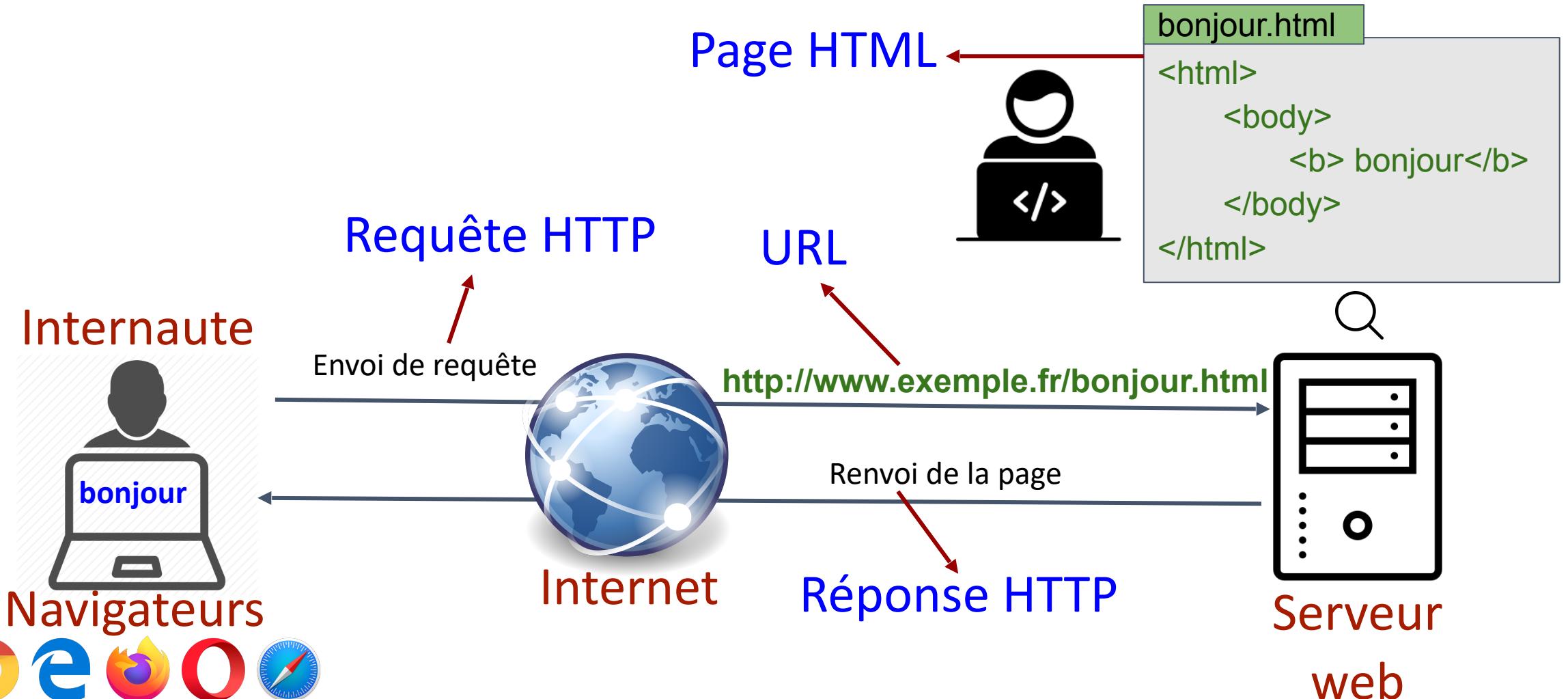
# Web statique VS Web dynamique



# Web statique VS Web dynamique



# Web : mode de fonctionnement



## Web : HTTP

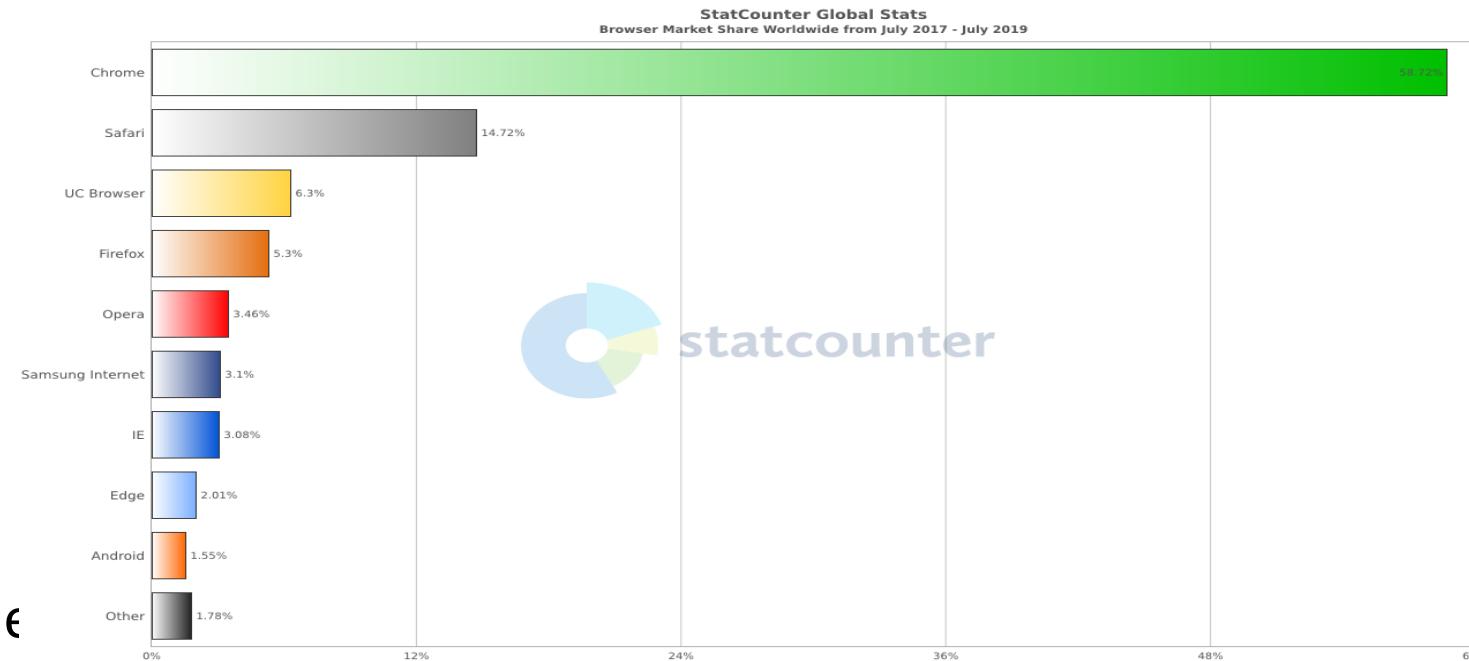
- HTTP (Hypertext Transfer Protocol) est un protocole de communication entre un client et un serveur
- HTTPS (avec **S** pour secured, soit « sécurisé ») est la variante sécurisée du HTTP. Il utilise le protocole TLS (Transport Layer Security)
- Version actuelle : HTTP 2.0



- Principe :
  - a. Le client HTTP (navigateur, ...) envoie une requête (GET, POST, ...)
  - b. Le serveur (apache, ...) renvoie plusieurs informations au client HTTP. Parmi elles, le document demandé

# Navigateur web

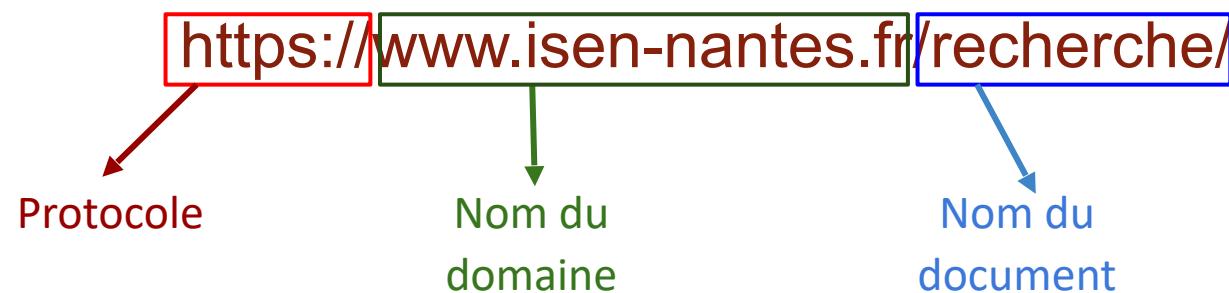
- C'est un client HTML qui permet de consulter et d'afficher le world wide web. Quel navigateur doit-on utiliser ?



- Chrome possède un outil
- Exemple : utilisation de de [isen-nantes.fr](http://isen-nantes.fr)

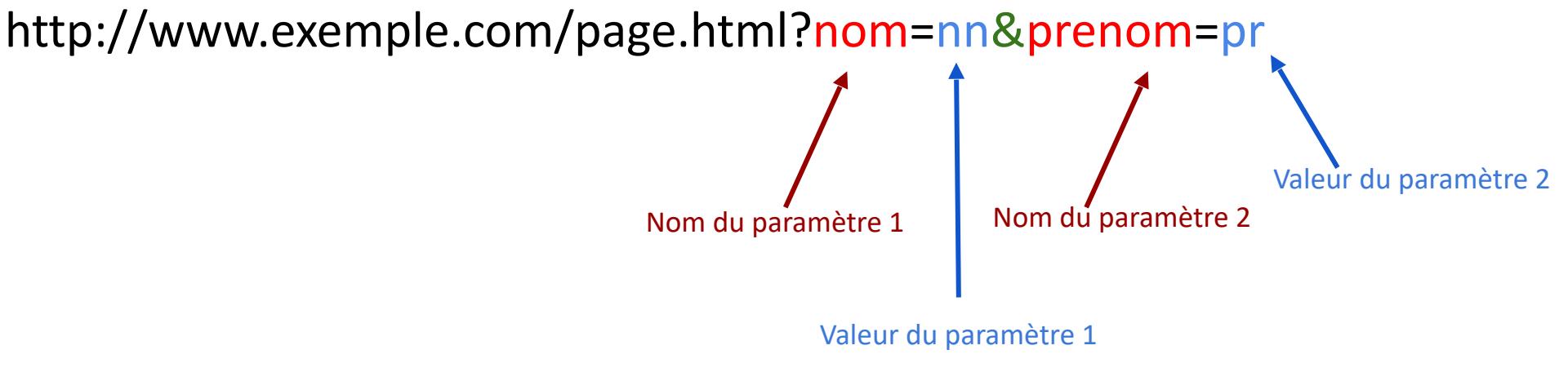
# Web : URL

- Souvent appelé **adresse Web**
- Les requêtes HTTP concernent, généralement, une ressource identifiée par son URL ( Uniform Resource Locator = localisateur uniforme de ressource)
- URL ↗ URI (Uniform Resource Identifier = identifiant uniforme de ressource)
- Contenu basique d'une URL : le protocole (comment), le nom du domaine (où) et le nom du document (quoi)



## Web : URL

- Les navigateurs ajoutent automatiquement "http://" ou "https://" avant le site demandé
- Dans chaque serveur web, un fichier par défaut est programmé (index.html, default.htm, index.php, ...)
- Il est possible de transmettre des données à travers une URL



The diagram shows a URL in a red-bordered box: `http://www.exemple.com/page.html?nom=nn&prenom=pr`. Red arrows point from the text "Nom du paramètre 1" to the word "nom" and from "Nom du paramètre 2" to the word "prenom". Blue arrows point from the text "Valeur du paramètre 1" to the value "nn" and from "Valeur du paramètre 2" to the value "pr".

http://www.exemple.com/page.html?nom=nn&prenom=pr

Nom du paramètre 1

Nom du paramètre 2

Valeur du paramètre 1

Valeur du paramètre 2

## Web : URL

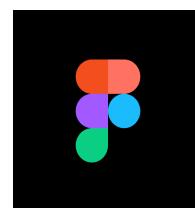
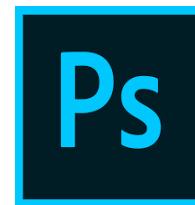
- Une URL est réellement une adresse IP (Internet Protocol)
- Le protocole qui se charge de traduire les noms de domaine en adresse IP est DNS (Domain Name System)
- Exemple :
  - utilisation de la commande tracert en windows ou traceroute en mac
  - l'extension **Network and Internet tools** sur Google Chrome

# Plan

- Généralités
  - Internet
  - Web
  - Front-end VS Back-end VS Full-stack
- HTML
  - Développement Front-end
  - Définition et versions
  - Balises HTML

# Développeur Front-end

- Un développeur Front-end s'occupe de la programmation côté client =  
Il s'intéresse aux éléments que l'on voit à l'écran et avec lesquels on peut interagir
- Ses champs de compétences sont :
  - Le web design (Photoshop, Figma, Adobe XD, Sketch, ...)



- La programmation web

Langages



Frameworks  
(React, Angular, Vue, Bootstrap, ...)



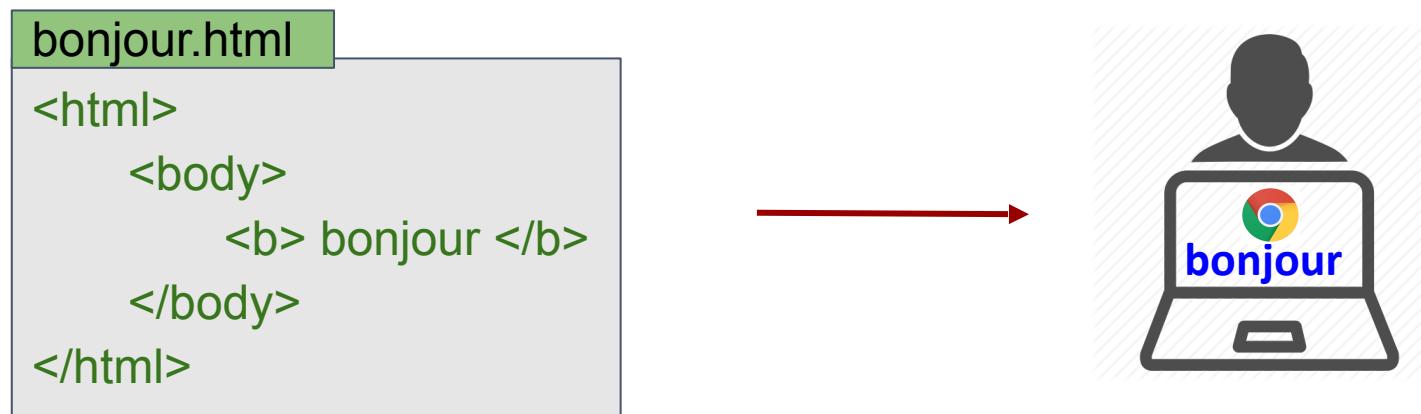
# Développeur Front-end

- Outils/sites incontournables pour un développeur



# Définition

- HTML (HyperText Markup Language) est un langage de balisage qui sert à représenter les pages web. Ce n'est pas un langage de programmation
- HTML mets en forme les documents hypertextes (ensemble d'informations liées entre elles par des hyperliens)
- HTML est inspiré du Standard Generalized Markup Language (SGML)
- Une page web = document (en extension .html) contenant du code HTML = balises + texte

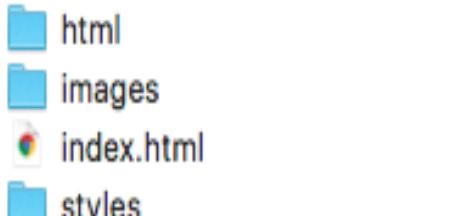


# Versions

- 1995 : HTML 2.0
- Début 1997 : HTML 3.2
- Fin 1997 : HTML 4.0
- Fin 1999 : HTML 4.0.1
- 2014 : HTML 5
- 2016 : HTML 5.1
- **2017 : HTML 5.2**
  
- Comment connaître la version HTML utilisée dans un site web ?
  - Ouvrir un site dans google chrome. Par exemple : <https://isen-nantes.fr/>
  - Afficher le code HTML via le devtools (clique droit + inspecter) et consulter l'entête : <!DOCTYPE html> → version actuelle : HTML 5.2



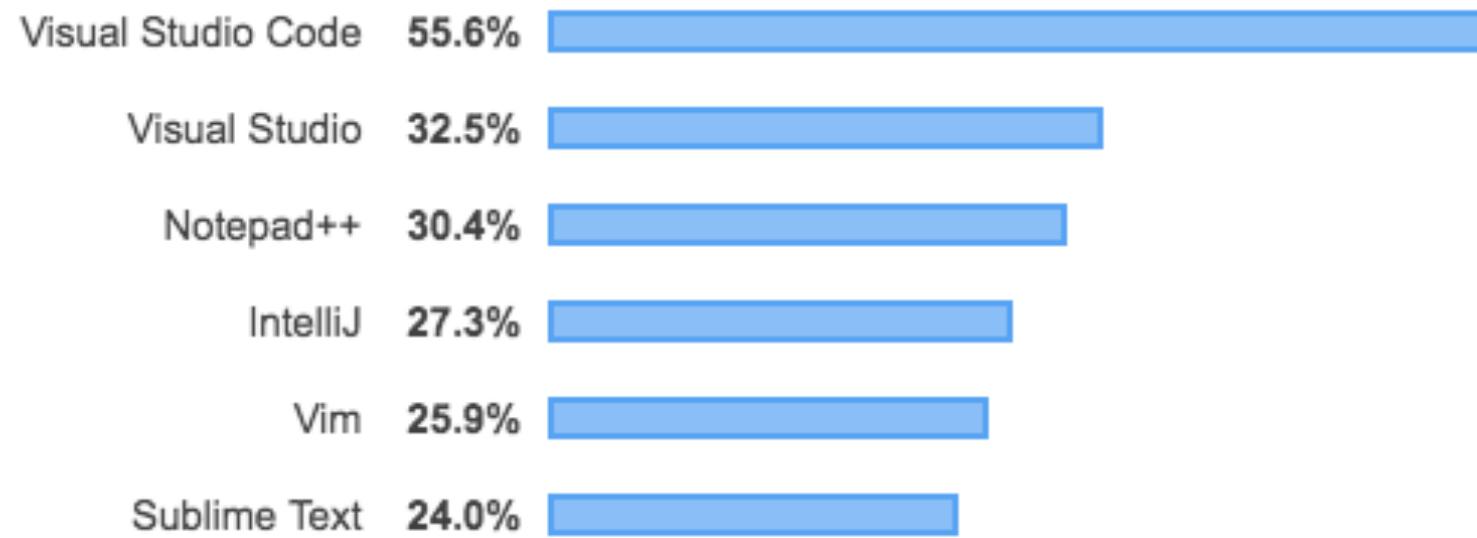
# Où écrire du code HTML : Éditeur de texte **VS** Editeur de code ou IDE<sup>1</sup>

	Editeur de texte	Editeur de code ou IDE
Exemple	Notepad, TextEdit, ...	Sublimetext, Visual studio code, Brackets, ...
Structure (indentation automatique) + couleur	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;meta charset="utf-8"&gt; &lt;title&gt; Titre &lt;/title&gt; &lt;/head&gt; &lt;body&gt; Contenu de la page &lt;/body&gt; &lt;/html&gt;</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;html lang="fr"&gt;   &lt;head&gt;     &lt;meta charset="utf-8"&gt;     &lt;title&gt; index &lt;/title&gt;   &lt;/head&gt;   &lt;body&gt;     texte   &lt;/body&gt; &lt;/html&gt;</pre>
Gestion des fichiers	 <ul style="list-style-type: none"> <li>▶ html</li> <li>▶ images</li> <li>▶ index.html</li> <li>▶ styles</li> </ul>	<ul style="list-style-type: none"> <li>◀ codes_cours</li> <li>▶ html</li> <li>▶ images</li> <li>▶ styles</li> <li>◀ index.html</li> </ul>

<sup>1</sup> Integrated Development Environment

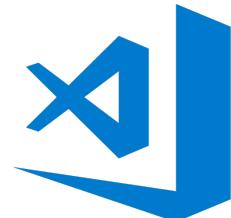
# Quel IDE choisir en programmation Front-end?

- Classement du site StackOverflow (<https://insights.stackoverflow.com/survey/2019#technology>)



# Visual Studio Code

- Editeur de code développé le 29/04/2015 par Microsoft
- Open source (<https://github.com/Microsoft/vscode>) + gratuit
- Programmé en TypeScript, JavaScript et CSS
- Disponible en plusieurs systèmes d'exploitation : Windows, Linux et Mac OS
- Différents packages et extensions disponible
- Outils de débogage
- Mises à jours fréquentes
- Auto-complétion puissante
- Plusieurs extensions



<https://code.visualstudio.com/>

## Choix des noms de fichiers : bonnes pratiques

- L'extension doit être appropriée avec le type de fichier
- Éviter les espaces. Utilisez plutôt "\_" ou "-"
- Ne pas utiliser les caractères spéciaux (?, %, #, /, :, ;, •,) et internationaux (å, €, ...)
- Utiliser que des minuscules
- Adopter des noms courts et significatifs

## Ce que le navigateur ignore dans un fichier HTML

- Les espaces vides et les tabulations
  - `isen, brest nantes devient isen, brest, nantes`
- Les retours à la ligne
- Les éléments HTML non connus
- Les commentaires HTML

# Balises HTML

- Une balise est un nom d'un élément de la page web (souvent son abréviation)
- Une balise est un mot clé entre "<" et ">"
  - Généralement, une balise de début (ouvrante) et de fin (fermante)
  - Certaines balises sont seules (vides, elles n'ont pas de contenus). Exemple : <br>

Balise ouvrante

<nomelement>

Contenu (texte ou  
autres éléments HTML)

contenu ici </nomelement>

Balise fermante  
(commençant par /)

- Exemple :

<p> contenu ici </p>

# Types d'élément

## En ligne (display:inline)

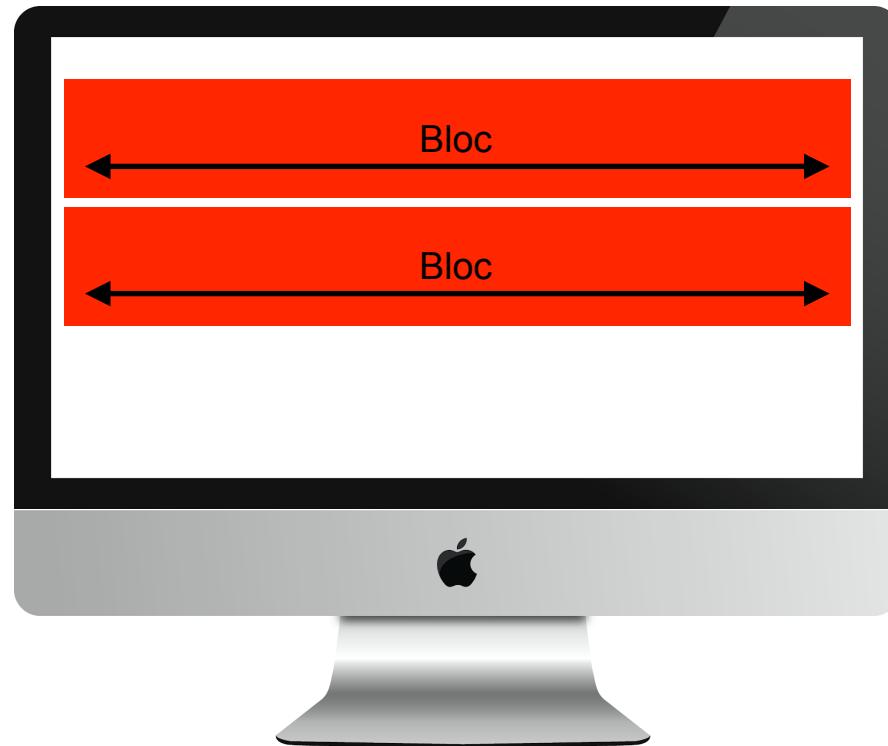
- Il s'insère dans la ligne courante (actuelle) et prend la largeur qui leur ai nécessaire

## En bloc (display:block)

- Les éléments de rendu CSS block se placent toujours l'un en dessous de l'autre par défaut (comme un retour chariot) et prennent toute la largeur de l'écran.

## Un élément de type bloc

Un élément de type bloc occupe automatiquement, par défaut, toute la largeur disponible dans son conteneur.



Paragraphe : **< p > ... </ p >** (*paragraph*)

Saut de ligne : **< br >** (*breaking row*)

Ligne de séparation : **< hr >** (*horizontal row*)

Division : **< div > ... </ div >**

Tableau : **< table > ... </ table >** etc.

## Autres éléments de type bloc

Un élément de type bloc occupe automatiquement, par défaut, toute la largeur disponible dans son conteneur.

Titres :      **<h1> </h1>**  
                 **<h2> </h2>**

...

**<h6> </h6>**

Adresse :     **<address> </address>**

Citations :    **<blockquote>**  
                  **<p>**permet de regrouper plusieurs blocs**</p>**  
                 **</blockquote>**

## Autres éléments de type bloc

### Listes

```
<ul>  
    <li></li>  
</ul>
```

{

Liste à puces (unordered)

==

- élément 1
- élément 2
- élément 3

```
<ol>  
    <li></li>  
</ol>
```

{

Liste numéroté (ordered)

==

1. élément 1
2. élément 2
3. élément 3

```
<dl>  
    <dt></dt>  
    <dd></dd>  
</dl>
```

{

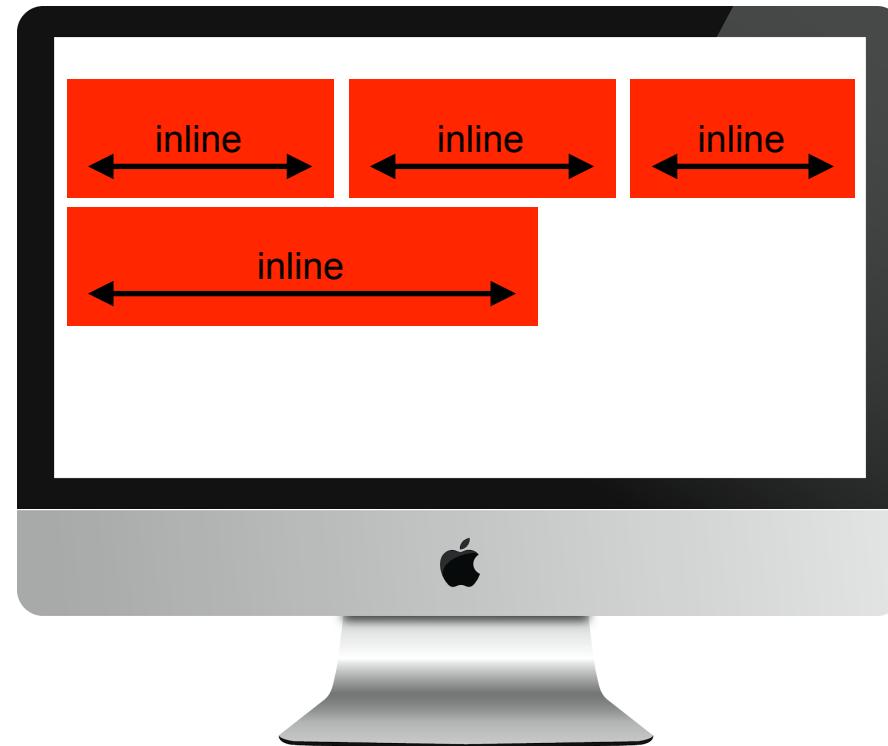
Liste de définition  
dl : conteneur  
dt : terme  
dd : définition

==

Terme 1  
    définition terme 1  
Terme 2  
    définition terme 2

## Un élément de type inline

Les éléments inline se placent toujours l'un à côté de l'autre afin de rester dans le texte. Par défaut, leur taille va être déterminée par le texte ou l'élément qu'ils contiennent.



Emphasis : **<em>...</em>**

Strong : **<strong>...</strong>**

Indice : **<sub>...</sub>**

Exposant : **<sup>...</sup>**

etc.

# Commentaires

- Les commentaires en HTML sont entre `<!--` et `-->`. Autrement dit, le contenu entre `<!--` et `-->` ne s'affiche pas par le navigateur

`<!-- Commentaire dans une ligne -->`

`<!-- Commentaire  
multi-line-->`

# Attributs

- Un élément peut avoir aucun ou plusieurs attributs
- Un attribut clarifie ou modifie l'élément
- Syntaxe :

Balise ouvrante

```
<nomelement nomattribut="valeur"> contenu ici </nomelement>
```

Attribut

Contenu (text ou autres éléments HTML)

Balise fermante  
(commençant par /)

```
  
<p id="1"> contenu ici </p>
```

- La valeur d'un attribut est dans certains cas fixe (booléen, texte, ...)

# Structure basique d'un document HTML

<!DOCTYPE>

Renseigne le navigateur sur le langage utilisé par votre page web.

<html>

<head>

Donne des informations à votre navigateur pour traiter correctement les informations se trouvant dans le «body»

</head>

<body>

La partie visible de votre page qui sera affichée par le navigateur

</body>

</html>

# Déclaration de type de document

- Le type de document informe le navigateur de la manière avec laquelle il va traiter le document
- Exemple
  - XHTML 1.0 :  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
  - HTML 4.01  
`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
  - HTML 5  
`<!DOCTYPE html>`

# Structure basique d'un document HTML

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title> Titre </title>
  </head>
  <body>
    Contenu de la page
  </body>
</html>
```

Encodage



Tester le bout de code à gauche dans le site suivant :  
[https://www.w3schools.com/tags/tryit.asp?filename=tryhtml\\_doctype](https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_doctype)

# Validation d'un document HTML

- Un document HTML valide est un document qui respecte la syntaxe correcte de la version HTML utilisée
- L'outil fiable pour vérifier la validation d'un document HTML est :  
<http://validator.w3.org/>
  - Insertion de l'URL vers le document
  - Utilisation d'un fichier html
  - Insertion du code HTML

# HTML vs xHTML

- xHTML respecte strictement le modèle XML (eXtensible Markup Language)  
→ plus strict que HTML
- En gros, 6 règles sont à respecter :
  - noms de balises et attributs en minuscule : <b> et non <B>
  - balises de fin obligatoires pour balises non vides : <p>text</p> et non <p>Hi hi
  - balises vides terminées <br/> et non <br >
  - balises correctement imbriquées <b><i>Hi</i></b> et non <b><i>Hi</b></i>
  - valeurs d'attributs entre guillemets <body bgcolor="white"> et non <body  
bgcolor=white>
  - 6 attributs avec valeur obligatoire checked="checked" et non ckecked

# Caractères spéciaux

- Certains caractères ne sont pas acceptés en HTML. Ils peuvent être remplacés par des entités
  - **Syntaxe :**
    - &numero\_entite;
    - &nom\_entite;
  - **Exemple :**

&	&amp;	&#38;
"	&quot;	&#34;
'	&apos;	&#39;
€	&euro;	&#8364;

# Opérations sur le texte

Balise	Description
<code>&lt;p&gt;...&lt;/p&gt;</code>	paragraphe
<code>&lt;h1&gt;...&lt;/h1&gt;</code> <code>&lt;h2&gt;...&lt;/h2&gt;</code> <code>&lt;h3&gt;...&lt;/h3&gt;</code> <code>&lt;h4&gt;...&lt;/h4&gt;</code> <code>&lt;h5&gt;...&lt;/h5&gt;</code> <code>&lt;h6&gt;...&lt;/h6&gt;</code>	six niveaux de titre
<code>&lt;hr&gt;</code>	séparation de deux contenus
<code>&lt;pre&gt;...&lt;/pre&gt;</code>	texte préformaté

## Opérations sur le texte

Balise	Description
<code>&lt;div&gt;...&lt;/div&gt;</code>	organisation de contenu
<code>&lt;br&gt;</code>	retour à la ligne
<code>&lt;b&gt;...&lt;/b&gt;</code>	texte en gras
<code>&lt;u&gt;...&lt;/u&gt;</code>	texte souligné
<code>&lt;i&gt;...&lt;/i&gt;</code>	texte en italique

# Listes

Balise	Description
<pre>&lt;ul&gt;     &lt;li&gt;...&lt;/li&gt; &lt;/ul&gt;</pre>	Liste non-ordonnée
<pre>&lt;ol&gt;     &lt;li&gt;...&lt;/li&gt; &lt;/ol&gt;</pre>	Liste ordonnée
<pre>&lt;dl&gt; &lt;!-- description list--&gt;     &lt;dt&gt;...&lt;/dt&gt; &lt;!-- defines terms--&gt;     &lt;dd&gt;...&lt;/dd&gt; &lt;!-- defines definition --&gt; &lt;/dl&gt;</pre>	Liste descriptive

# Formulaire

Balise	Description
<code>&lt;form method=.." action=.."&gt; ... &lt;/form&gt;</code>	Début et fin du formulaire
<code>&lt;input type="..."&gt;</code>	Contrôle interactif (texte, password, radio, ...)
<code>&lt;select&gt;   &lt;option value="texte"&gt;...&lt;/option&gt;   ... &lt;/select&gt;</code>	Liste d'options
<code>&lt;textarea rows="10" cols="30"&gt; ...&lt;/textarea&gt;</code>	Zone de texte
<code>&lt;button type="button" value=""&gt;...&lt;/button&gt;</code>	Bouton

# Tableaux

Balise	Description
<table> ... </table>	Début et fin du tableau
<tr> ...</tr>	Déclaration du ligne (table row)
<th> ...</th>	Déclaration d'un en-tête (table header)
<td> ...</td>	Déclaration d'une cellule (table cell data)
<caption> ... </caption>	Titre du tableau

```
<table>
  <tr> <th>Menu item</th> <th>Calories</th> <th>Fat</th> </tr>
  <tr> <td>Chicken noodle
       soup</td> <td>120</td> <td>2</td> </tr>
  <tr> <td>Caesar salad</td> <td>400</td> <td>26</td> </tr>
</table>
```

Source : Learning Web Design, 5th Edition,  
Jennifer Niederst Robbins

# Tableaux

Attributs	Description
<code>&lt;td colspan="<b>n</b>"&gt; ... &lt;/td&gt;</code> <code>&lt;th colspan="<b>n</b>"&gt; ... &lt;/th&gt;</code>	Fusion de " <b>n</b> " colonnes
<code>&lt;td rowspan="<b>m</b>"&gt; ...&lt;/tr&gt;</code> <code>&lt;th colspan="<b>m</b>"&gt; ... &lt;/th&gt;</code>	Fusion de " <b>m</b> " lignes

- Il est pratique de regrouper :
  - plusieurs `<th>` dans `<thead>`
  - plusieurs `<tr>` dans `<tbody>`
- La balise `<tfoot>` sert à insérer un *footer* au tableau (comme un résumé par exemple)

# Tableaux

- Avant la version HTML5, un attribut border dans la balise table permet de générer les bordures : <html border="5">

L'attribut border ainsi que d'autres attributs (bgcolor, align, ...) ne sont plus autorisés en HTML5

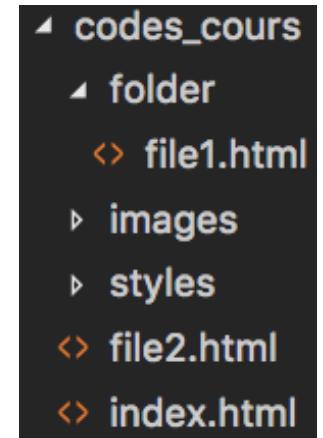
- **Solution :** utilisation d'un style CSS (Cascading Style Sheets) pour les tableaux (à insérer dans le <head>)



```
<style>
    table, td, th {border: 1px solid}
</style>
```

# Liens

- **Syntaxe :** `<a href="url">Contenu du lien</a>`
- **Lien vers une page dans le web :**
  - Exemple : `<a href="https://isen-nantes.fr/recherche">ISEN-Nantes-rech</a>`
- **Lien relatif (URL relative au chemin courant) vers une page web dans le serveur :**
  - Exemple :
    - Accès à file1.html depuis index.html  
`<a href="folder/file1.html">lien</a>`
    - Accès à index.html depuis file2.html  
`<a href="index.html">lien</a>`
    - Accès à index.html depuis file1.html  
`<a href="../index.html">lien</a>`
- **Lien root-relatif (URL relative à partir de la racine) vers une page web dans le serveur :**
  - Exemple : `<a href="/code_cours/folder/file1.html">linked content</a>`



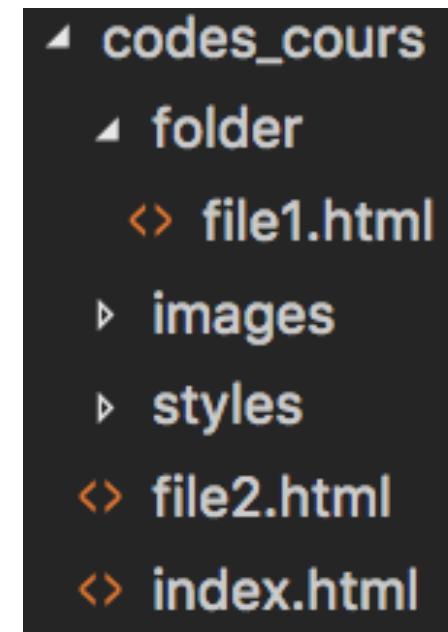
# Liens

## ■ Lien vers un élément dans le même document HTML

- Étape 1 : Identification de la destination
  - Exemple : `<h2 id="id2">texte</h2>`
- Étape 2 : Lien vers la destination
  - Exemple : `<a href="#id2">lien</a>`

## ■ Lien vers un élément dans un autre document HTML

- Étape 1 : Identification de la destination dans file2.html
  - Exemple : `<h2 id="id2">H</h2>`
- Étape 2 : Lien vers la destination depuis index.html
  - Exemple : `<a href="file2.html#id2">lien</a>`



# Liens

- Ouverture d'un lien dans un nouvel onglet dans le navigateur
  - **Syntaxe :** `<a href="url" target="_blank">lien</a>`
- Lien vers un mail
  - **Syntaxe :** `<a href="mailto:name@example.com">Mail</a>`
- Lien vers un téléphone
  - **Syntaxe :** `<a href="tel:+01-800-555-1212">Tel</a>`

# Médias

- Un fichier multimédia stocké dans le serveur web possède :

- **Un type MIME** (Multipurpose Internet Mail Extensions)<sup>1</sup>
- **Un suffixe** (une extension) ou plusieurs

Exemple : SVG (Scalable Vector Graphics)

- **Type MIME** : images/svg+xml
- **Suffixes** : .svg ou .svgz

- L'emplacement des informations MIME diffère d'un serveur à un autre.

Par exemple dans le cas du serveur Apache, ces informations sont stockées dans un fichier localisé dans un dossier racine appelé htaccess.

---

1 <https://htmlstrip.com/mime-file-type-checker>

# Médias : images

**Syntaxe : **

**src** = source où se trouve l'image, l'url peut être absolue ou relative (obligatoire)

**alt** = description de l'image pour l'accessibilité(et le référencement) ou si l'image ne s'affiche pas (obligatoire)

**title** = pour infobulle

**width** = largeur de l'image en pixel

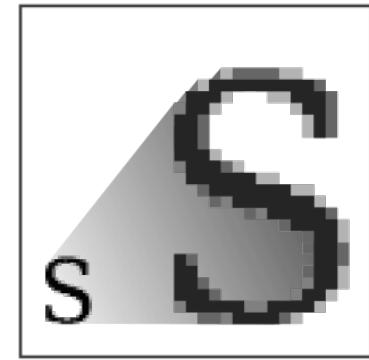
**height** = hauteur de l'image en pixel

Bonne pratique : regrouper toutes les images dans un répertoire (nommé **images** par exemple)

Le navigateur, au chargement de l'image, il l'enregistre dans la mémoire cache (mémoire temporaire)

# Médias : images

- On peut définir une largeur et une hauteur d'une image en pixel en utilisant les attributs **width** et **height** :
  - Exemple : 
- Formats des images souvent utilisés sur le web :
  - Images matricielles (bitmap)
    - PNG (Portable Network Graphics)
    - JPEG (Joint Photographic Experts Group)
    - GIF (Graphics Interchange Format)
  - Images vectorielles
    - SVG (Scalable Vector Graphics)
  - ...



Matriciel  
.jpeg .gif .png



Vectoriel  
.svg

# Médias : images

- .svg (.svg) est le format d'image recommandé en web qui peut s'insérer :

- **avec la balise <img>**

- Exemple :

- ```

```

- Avantages :

- ?] Utile pour les illustrations statiques

- ?] Appel à des fichiers externes

- ?] Supporté par la plupart des navigateurs sauf Internet Explorer 8 et Android Browser 3

- Inconvénients :

- ?] Impossible d'appliquer un style externe (.css)

- ?] Impossible de manipuler l'image en utilisant JavaScript (.js)

# Médias : images

- .svg (.svg) est le format d'image recommandé en web qui peut s'insérer :

- avec la balise `<img>`
- **avec la balise `<svg> ... </svg>`**

- Exemple :

```
<svg width="100" height="100">  
    <circle cx="50" cy="50" r="40" stroke="black" stroke-width="4" fill="gray" />  
</svg>
```

- Inconvénients :

- Le code d'une image .svg peut être généré MAIS il est souvent long

- Avantages :

- Interaction facile avec les fichiers de style ".css" et les fichiers contenant du code JavaScript ".js"
  - Supporté par la plupart des navigateurs

# Médias : images

- .svg (.svg) est le format d'image recommandé en web qui peut s'insérer :

- avec la balise `<img>`
  - avec la balise `<svg> ... </svg>`
  - **avec la balise `<object> ... </object>`**

- Exemple :

```
<object type="image/svg+xml" data="logo-isen.svg">
    
</object>
```

- Inconvénients :

- Impossible d'appliquer un style externe ".css"

- Avantages :

- Appel à des fichiers externes en cas d'erreur



# Médias : images adaptatives

- Un site web **responsive (adaptatif)** est un site qui est conçu et développé de façon à pouvoir s'adapter à toutes les résolutions d'écran (PC, laptop, tablette, phablette, ...)
- Un site pour qu'il soit *responsive*, il faut que, entre autres, ses images soient responsive aussi



# Médias : images adaptatives

## ■ Résolution dynamique (commutation de résolution) :

- la même image est stockée dans le serveur en différentes tailles
- le navigateur choisit le format adéquat en se basant sur la résolution du périphérique chargé de l'affichage

```

```



**px** : pixel

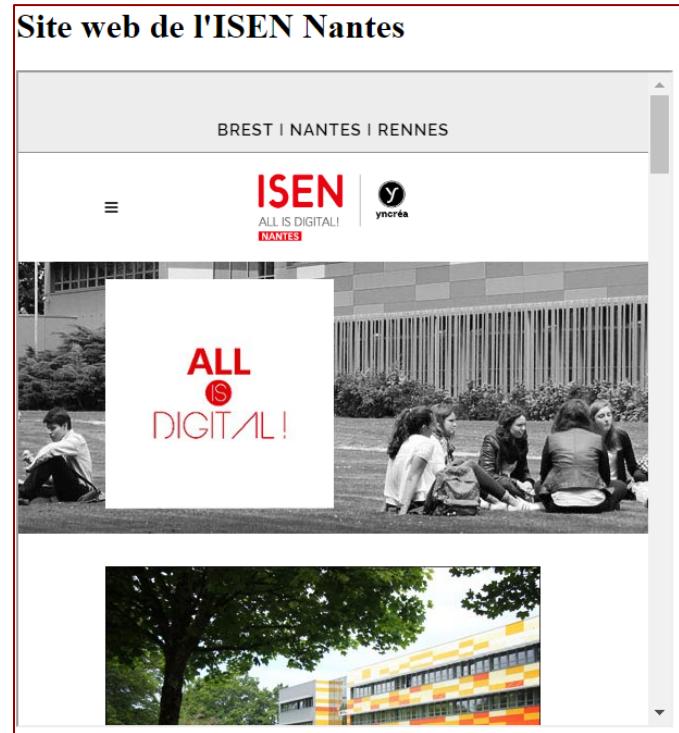
**vw** : viewport = surface de la fenêtre  
du navigateur

# Médias : iframe

- La balise **iframe** permet l'affichage d'une page web à l'intérieur d'une autre page web
- **Syntaxe :** <iframe src="URL" width="200" height="200" name="nom"></iframe>

- URL : lien absolu ou relatif vers le contenu
- width : largeur
- height : hauteur
- name : nom de l'élément (utile pour les liens avec l'attribut **target**)

## Site web de l'ISEN Nantes



# Médias : Vidéo

- Aucun consensus existe entre les navigateurs sur l'extension à utiliser pour les fichiers vidéos (MP4, WebM, Ogg, ...)
- Syntaxe :

- **controls** : affichage de play/pause ..
- **src** : URL relative ou absolue
- **width** : largeur
- **height** : hauteur
- **type** : MIME

```
<video width="320" height="240" controls>
    <source src="movie.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
</video>
```

# Médias : Audio

- Pareil que les vidéos, aucun consensus existe entre les navigateurs sur l'extension à utiliser pour les fichiers audio (MP3, WAV, Ogg, ...)
- Syntaxe :

- **controls** : affichage de play/pause ..
- **src** : URL relative ou absolue
- **type** : MIME

```
<audio controls>
  <source src="audio.mp3" type="audio/mp3">
  <source src="audio.ogg" type="audio/ogg">
</audio>
```

# Médias



**Vous devez avoir le droit d'utilisation pour  
chaque média (image, vidéo, audio, ...)  
utilisé dans votre site web**