

# Technologies web

- HTML5
- CSS3
- JavaScript

**HTML**



# Le langage HTML

- Historique & versions
- Service web
- Outils de développement
- Structure d'une page web
- Compatibilité : Validation W3C
- Les styles css : séparer le fond de la forme
- Davantage de balises
- Pour aller plus loin : les frameworks

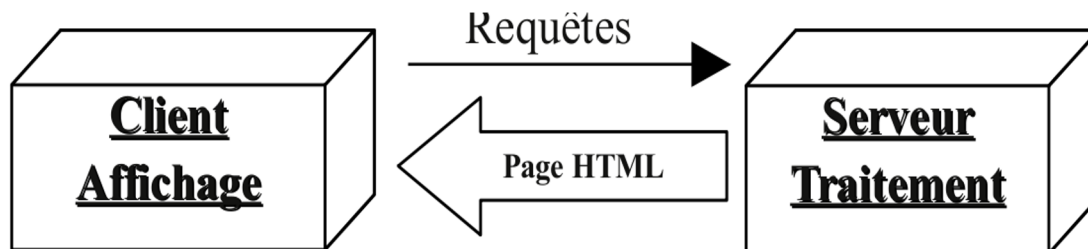


# Historique

- **HTML 1** : c'est la toute première version créée en 1991.
- **HTML 2** : apparaît en 1994 et prend fin en 1996. C'est cette version qui posera en fait les bases des versions suivantes du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C (tandis que la première version a été créée par un seul homme).
- **HTML 3** : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, les applets, les scripts, le positionnement du texte autour des images, etc.
- **HTML 4** : il s'agit de la version la plus répandue du HTML (plus précisément, il s'agit de HTML 4.01). Elle apparaît pour la première fois en 1998 et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires, etc. Mais surtout, cette version permet pour la première fois d'exploiter des feuilles de style.
- **HTML 5** : c'est LA dernière version. Elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc.

# Le service web

- **Serveur Web (RFC HTTP 1945)**
  - Programme applicatif acceptant des connexions dans le but de traiter des requêtes HTTP en délivrant une réponse HTTP.
  - Peut être considéré comme serveur de fichiers ou de documents.
- **Client Web (RFC HTTP 1945)**
  - Programme applicatif qui permet d'émettre des requêtes HTTP et interpréter les réponses HTTP afin d'extraire les données reçues (HTML, Javascript, etc ...). Le plus souvent, l'applicatif client est un navigateur (en anglais browser).
  - Les plus utilisés sont : Firefox, Internet Explorer, Safari, Chrome, Opéra, Lynx, Konqueror, ...
- **Architecture Client/Serveur**
  - Topologie la plus courante actuellement. Le serveur est en attente des demandes en provenance du client



# Outils de développement

## Création des pages web

1. Editeur de texte basique (ex : bloc-notes)
2. Editeur de texte pour langages informatiques (ex : NotePad++, SublimeText...) :
  - Reconnaissance d'éléments de langage
  - Coloration syntaxique
  - Modules additionnels (FTP)
3. Editeurs WYSIWYG (ex : DreamWeaver, Komposer...) :
  - Evitent l'utilisation du langage HTML,
  - Facilité d'emploi,
  - Mauvaise qualité du code généré par l'outil
4. CMS : Content Management System (ex : wordpress)

# Outils de développement

## Test des pages web

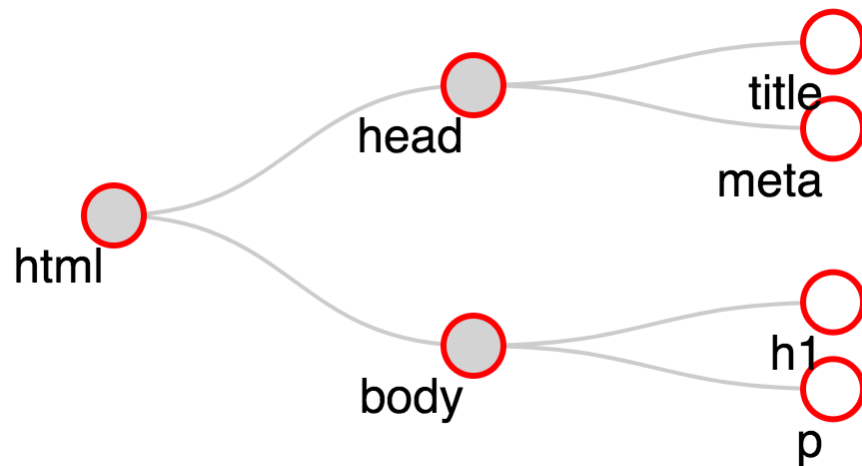
1. Ouverture du fichier html par l'explorateur de fichier  
(adresse de type <file:///C:/Users/.....>)  
Ne fonctionne qu'avec le HTML / CSS / Javascript
2. Installation d'un serveur HTTP sur son poste (ex : Apache via Xampp)  
(adresse de type <http://127.0.0.1/.....>)  
Bonne pratique pour :
  - Préparer le déploiement sur un serveur sur le net
  - Tester sa page dans différents navigateurs
  - Evoluer vers des pages dynamiques (PHP, javascript...)
3. Utilisation d'un NAS (Network Access Storage)
4. Utilisation d'un IDE en ligne  
[https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_default](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default)  
<http://codepen.io>, <http://jsbin.com>

Voir un comparatif des navigateurs : <https://www.w3counter.com/globalstats.php>

# Structure d'une page web

## Balises imbriquées

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Your first HTML page</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1>My home page</h1>
    <p>Hi! Welcome to my Home Page! </p>
  </body>
</html>
```



- A une balise ouvrante (tag) correspond généralement une balise fermante (close tag)
- Certaines balises sont « auto-fermantes » (self-closing tag)
- Certaines balises contiennent des attributs

# Structure d'une page web

## Ajout de contenu

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Ma première page</title>
  </head>

  <body>
    <h1>Bonjour tout le monde</h1>
    <p>Voici ma première page<br/></p>
    <hr/>
    <p>2° paragraphe</p>
    
  </body>
</html>
```



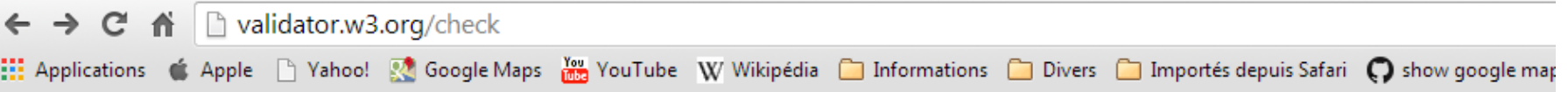
# Davantage de balises


## Mise en page

- `<ul> .. </ul>` Liste non triée, liste à puces
- `<ol> .. </ol>` Liste triée, liste à numéros
- `<li> .. </li>` Élément de la liste
- `<table> .. </table>` Définit un tableau
- `<tr> .. </tr>` Définit une ligne d'un tableau
- `<td> .. </td>` Définit une cellule de données

# Compatibilité entre navigateurs

## Validation W3C



 **Markup Validation Service**  
Check the markup (HTML, XHTML, ...) of Web documents

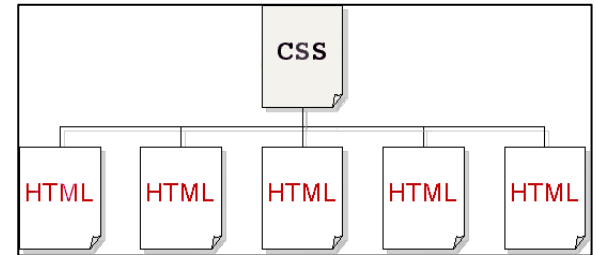
**Jump To:** [Notes and Potential Issues](#) [Validation Output](#)

**Error found while checking this document as HTML5!**

<b>Result:</b>	1 Error, 2 warning(s)	
<b>Source :</b>	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;meta http-equiv="Content-Type" content="text/html; charset=utf-8" /&gt;     &lt;link rel="stylesheet" href="style.css" /&gt;     &lt;title&gt;Ma première page&lt;/title&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;h1&gt;Bonjour tout le monde&lt;/h1&gt;     &lt;p&gt;Voici ma première page&lt;br/&gt;&lt;/p&gt;     &lt;hr/&gt;</pre>	
<b>Encoding :</b>	utf-8	<input type="text" value="(detect automatically)"/>
<b>Doctype :</b>	HTML5	<input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html	



# Les styles CSS

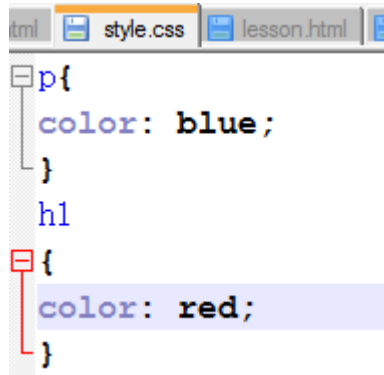


- Choix de la police, taille, couleur du texte
- Choix des bordures, fond d'écran
- Mise en page du site, largeur, hauteur, position des différentes zones de la page
- Versions :
  - **CSS 1** : dès 1996, elle pose les bases de ce langage qui permet de présenter sa page web, comme les couleurs, les marges, les polices de caractères, etc.
  - **CSS 2** : apparue en 1999 puis complétée par CSS 2.1, cette nouvelle version de CSS rajoute de nombreuses options. On peut désormais utiliser des techniques de positionnement très précises, qui nous permettent d'afficher des éléments où on le souhaite sur la page.
  - **CSS 3** : c'est la dernière version, qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, etc.

# Les styles css

## Mise en oeuvre

```
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  <link rel="stylesheet" href="style.css" />  
  <title>Ma première page</title>  
</head>
```



```
style.css  
p{  
  color: blue;  
}  
h1  
{  
  color: red;  
}
```

Modification des couleurs  
de titre et de paragraphe

# Les styles css

## class et formatage du texte

```
<body>
  <h1>Bonjour tout le monde</h1>
  <p class="paragraphe1">Voici ma première page<br/></p>
```

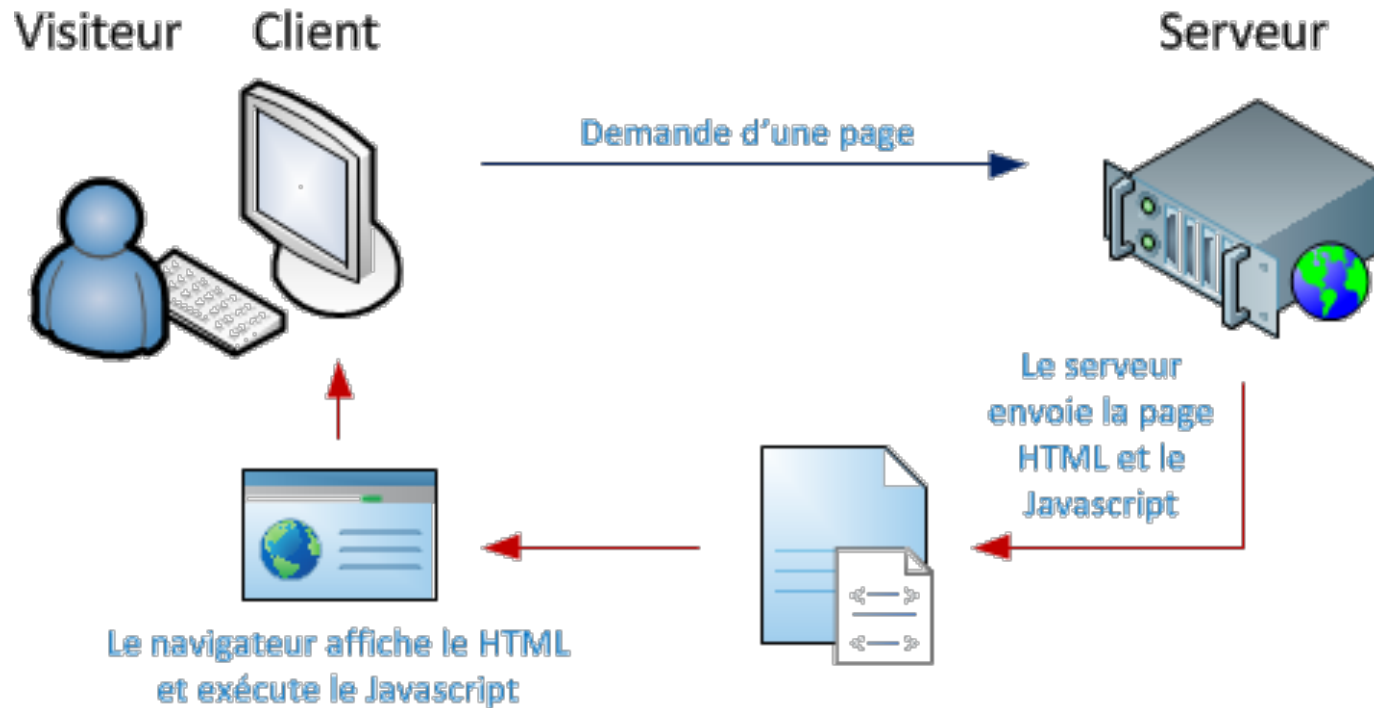
```
.paragraphe1
{
  font-size: x-large;
  font-family: Arial;
}
```

# Techniques modernes : Utilisation de frameworks

The screenshot shows the Layoutit website builder interface. The browser address bar displays [www.layoutit.com/build](http://www.layoutit.com/build). The interface includes a top navigation bar with buttons for 'Edit', 'Developer', 'Preview', 'Download', 'Share or Save', and 'Clear'. On the right side of the top bar are links for 'Register', 'Login', 'Feedback', and 'Donate!'. A left sidebar contains a 'GRID SYSTEM' section with options for 12, 8.6, 8.4, 4.4.4, and 2.6.4 columns, along with sections for 'BASE CSS', 'COMPONENTS', 'JAVASCRIPT', and 'COMMUNITY'. The main workspace shows a 'Container' with a 'Row' containing a 'Column'. The column contains a navigation bar with 'Brand', 'Link', 'Link', 'Dropdown', a search box, and a 'Submit' button. Below the navigation bar is a large hero unit with the text 'Hello, world!' and a 'Learn more' button. The hero unit text reads: 'This is a template for a simple marketing or informational website. It includes a large callout called the hero unit and three supporting pieces of content. Use it as a starting point to create something more unique.'

Exemple avec layoutit.com & Bootstrap3

# Pages statiques / Pages dynamiques



# Pages statiques / Pages dynamiques





# Le langage JavaScript

- Caractéristiques
- La console JavaScript
- Relations avec le HTML
- Gestion des évènements
- Le modèle DOM
- AJAX
- Outils de développement
- Bibliothèques externes



Voir un comparatif des langages de programmation : <https://www.tiobe.com/tiobe-index/>

# JavaScript : caractéristiques

- Prévu pour une exécution dans le navigateur
- Mais peut aussi s'exécuter sans navigateur (**Node.js**)
- Langage orienté objet mais faiblement typé :
  - number : **10**, **-34.5**, **1e-4** ...
  - boolean : **true** / **false**
  - string : '**Bonjour**', '**C'est cool**'
  - object : 

```
var car = {type:"Fiat", model:"500", color:"white"};
```
- Variables déclarées avec **var** ou **let** et constantes avec **const**  
(privilégier la notation camelCase : *nombreDeParamètres*, *jourDeLaSemaine*)
- Syntaxe proche de C/C++ et du Java
- S'insère dans le code html entre les balises **<script>...</script>**  
(on met directement le code JavaScript [entre les balises](#)  
ou alors dans un fichier avec l'extension **.js** et [on insère un lien](#))
- Permet d'interagir avec le html et le css

# JavaScript : la console

- Permet d'exécuter du code à la volée
- Utile pour déboguer les pages web :
  - Signalement des [erreurs](#)
  - Inspection des variables
  - Mode pas à pas
- Donne accès aux éléments du DOM et au css

# JavaScript : les objets simples

```
var stylo = {  
  type: "bille",  
  couleur: "bleu",  
  marque: "Bic"  
};  
  
console.log("Mon stylo à " + stylo.type + " " + stylo.marque + " écrit en " +  
stylo.couleur);  
  
stylo.couleur = "rouge"; // Modifie la couleur de l'encre du stylo  
  
console.log("Mon stylo à " + stylo.type + " " + stylo.marque + " écrit en " +  
stylo.couleur);
```

- Que va afficher la console ?

# JavaScript : les fonctions

```
// Fonction calculant le discriminant d'une équation
// du second degré : ax2 + bx + c = 0
function delta(a, b, c) {
    return (b**2 - 4*a*c)
}
```

```
// Fonction calculant les racines d'une équation
// du second degré : ax2 + bx + c = 0
function racines(a, b, c) {
    var d = delta(a, b, c);
    var solutions = [];
    if (d > 0) {
        solutions.push((-b-Math.sqrt(d))/(2*a));
        solutions.push((-b+Math.sqrt(d))/(2*a));
    }
    else if (d == 0) {
        solutions.push(-b/(2*a));
    }
    return solutions;
}
```

```
⋮ Console What's New
top
> delta(1,5,4)
< 9
> racines(1,5,4)
< ▶ (2) [-4, -1]
> delta(4,4,1)
< 0
> racines(4,4,1)
< ▶ [-0.5]
> delta(5,3,1)
< -11
> racines(5,3,1)
< ▶ []
>
```

A voir : <https://codepen.io/anon/pen/mqMoPE>

# JavaScript : relation avec le HTML

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Today's Date</title>
  </head>
  <body>
  </body>
</html>
```

```
>> let d = new Date();
>> document.body.innerHTML = "<h1>Today's date is " + d + "</h1>"
```

Output

```
"<h1>Today's date is Sat Jun 24 2017 12:16:14 GMT-0400 (EDT)</h1>"
```

# JavaScript : Utilisation du DOM

- DOM = Document Object Model
- Arbre d'objets HTML hiérarchisés
- Visible dans les outils de développement des navigateurs
- Utilisation de la balise **<id>**

# JavaScript : gestion des évènements

```
<span id="clickme">Cliquez-moi !</span>

<script>

  var element = document.getElementById('clickme');

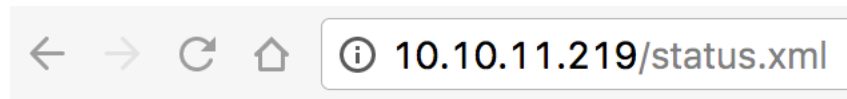
  element.onclick = function() {
    alert("Vous m'avez cliqué !");
  };

</script>
```



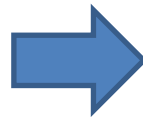
# JavaScript : AJAX

- AJAX = Asynchronous JavaScript And XML
- Ce n'est pas un langage, mais une technique de récupération de données à travers le web :



This XML file does not appear to have any style

```
▼<status>
  <!-- Energie par poste -->
  ▼<w11>
    <C1>985063</C1>
    <C2>314709</C2>
    <C3>178582</C3>
  </w11>
</status>
```



Actualiser	
Poste	Consommation
Climatisation	985063 Wh
Eclairage	314709 Wh
Prises	178582 Wh

# JavaScript : AJAX

## L'objet *XMLHttpRequest*

```
<script>
function loadXML() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      var xmlDoc = this.responseXML;
      var c1 = xmlDoc.getElementsByTagName("C1")[0].childNodes[0].nodeValue;
      var c2 = xmlDoc.getElementsByTagName("C2")[0].childNodes[0].nodeValue;
      var c3 = xmlDoc.getElementsByTagName("C3")[0].childNodes[0].nodeValue;
      document.getElementById("C1").innerHTML = c1 + " Wh";
      document.getElementById("C2").innerHTML = c2 + " Wh";
      document.getElementById("C3").innerHTML = c3 + " Wh";
    }
  };
  xhttp.open("GET", "http://10.10.11.219/status.xml", true);
  xhttp.send();
}
</script>
```

```
▼ <status>
  <!-- Energie par poste -->
  ▼ <W11>
    <C1>985063</C1>
    <C2>314709</C2>
    <C3>178582</C3>
  </W11>
</status>
```

# JavaScript : AJAX

## L'objet *XMLHttpRequest*

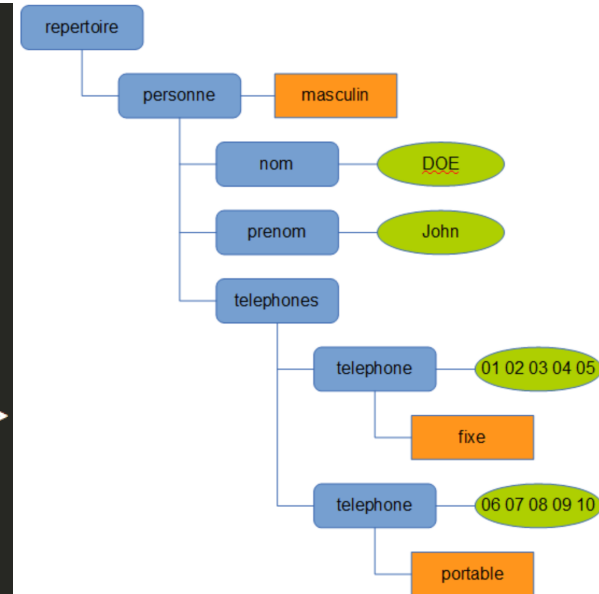
Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the <a href="#">Http Messages Reference</a>
statusText	Returns the status-text (e.g. "OK" or "Not Found")

Source : [https://www.w3schools.com/js/js\\_ajax\\_http\\_response.asp](https://www.w3schools.com/js/js_ajax_http_response.asp)

# JavaScript : AJAX

## La propriété `responseXML`

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<repertoire>
  <!-- John DOE -->
  <personne sexe="masculin">
    <nom>DOE</nom>
    <prenom>John</prenom>
    <telephones>
      <telephone type="fixe">01 02 03 04 05</telephone>
      <telephone type="portable">06 07 08 09 10</telephone>
    </telephones>
  </personne>
</repertoire>
```



élément

`xmlDoc.getElementsByTagName("nom")[0].childNodes[0].nodeValue;`

attribut

`xmlDoc` - Variable affectée avec la propriété `responseXML`

`getElementsByTagName("nom")[0]` - retourne le 1<sup>o</sup> élément `<nom>`

`childNodes[0]` - Le 1<sup>o</sup> enfant de l'élément `<nom>` (text node)

valeur

`nodeValue` - la valeur du nœud (le texte « DOE »)

# JavaScript : AJAX

## Le format JSON

```
var courses = {  
  "fruits": [  
    { "kiwis": 3,  
      "mangues": 4,  
      "pommes": null  
    },  
    { "panier": true }  
  ],  
  "legumes":  
    { "patates": "amandine",  
      "figues": "de barbarie",  
      "poireaux": false  
    }  
};
```



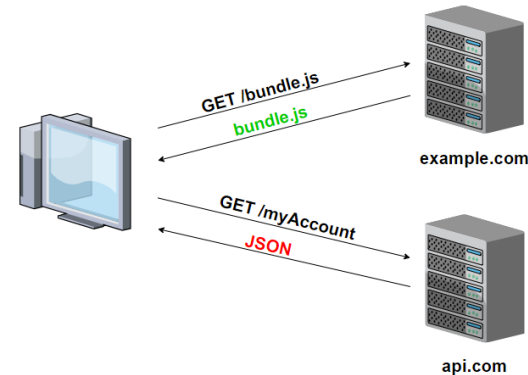
```
<?xml version="1.0" ?>  
<root>  
  <fruits>  
    <item>  
      <kiwis>3</kiwis>  
      <mangues>4</mangues>  
      <pommes></pommes>  
    </item>  
    <item>  
      <panier>>true</panier>  
    </item>  
  </fruits>  
  <legumes>  
    <patates>amandine</patates>  
    <figues>de barbarie</figues>  
    <poireaux>>false</poireaux>  
  </legumes>  
</root>
```

# JavaScript : AJAX

## La sécurité avec CORS

### Cross-Origin Resource Sharing

est un dispositif du navigateur qui permet d'obtenir des ressources provenant de multiple origines, ce qui est normalement bloqué par le navigateur.

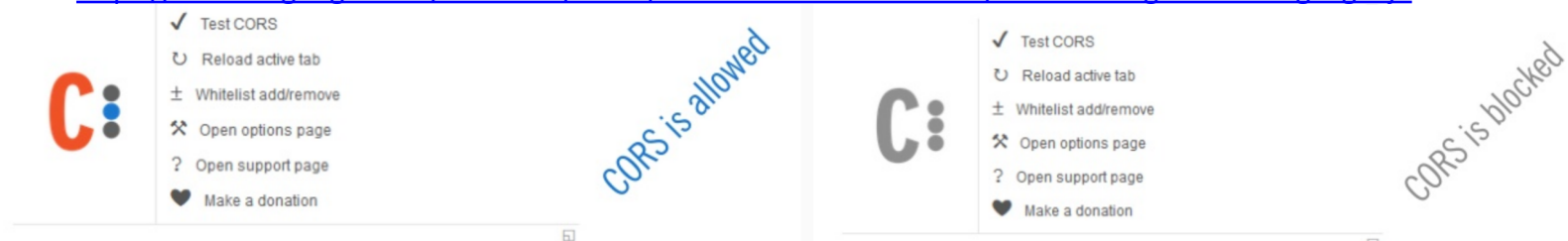


Ce blocage se produit quand une page html n'affiche pas les données en provenance d'un fichier XML, ce qui peut se voir avec la console du navigateur :

```
✖ Failed to load https://example.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'https://anfo.pl' is therefore not allowed access. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.
```

La plupart des navigateurs proposent un plugin pour activer CORS.

<https://chrome.google.com/webstore/detail/allow-cors-access-control/lhobafahddgcelffkeicbaginieeejlf>

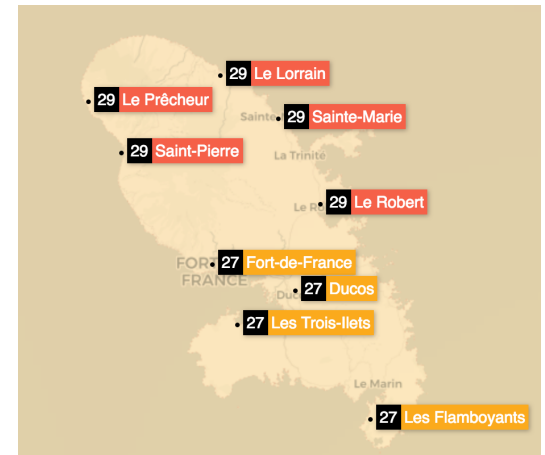


Source : <https://developer.mozilla.org/fr/docs/Web/HTTP/CORS>

# JavaScript : AJAX

## Quelques exemples d'API

- <https://openweathermap.org/api>



- <http://fixer.io/>

```
GET https://api.fixer.io/latest?symbols=USD,GBP
{
  "base": "EUR",
  "date": "2017-12-08",
  "rates": {
    "GBP": 0.87525,
    "USD": 1.1742
  }
}
```

# Quelques liens

Pour les RFC :

<http://abcdrfc.free.fr/>

<http://www.ietf.org/rfc.html>

Memento pour les balises

<https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-des-balises-html>

<http://css.mammothland.net/balises-html.php>

Memento pour le CSS

<https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-des-proprietes-css>

<http://www.loribel.com/info/memento/css.html#incl>

Pour les couleurs

<http://www.code-couleur.com/>

<http://html-color-codes.info/Codes-couleur-HTML/>