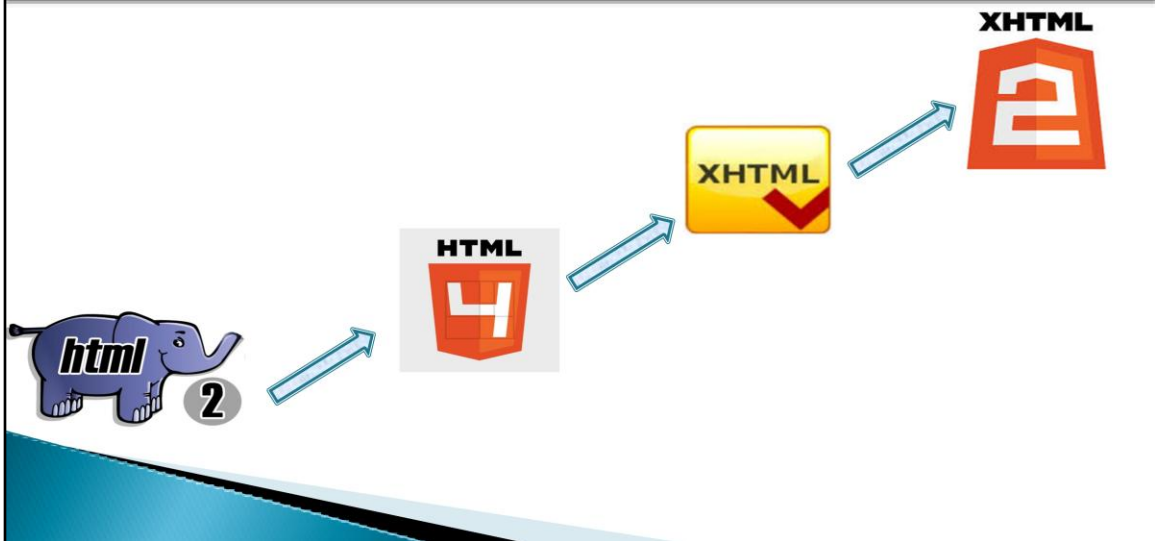




# Rappels

## Les versions HTML



C'est le World Wide Web Consortium (W3C) qui est chargé de l'évolution et de la normalisation du langage HTML. Depuis 1992, pas moins de cinq normes se sont succédées :

- La première version de HTML date de 1992. Bien loin de notre HTML actuel, elle ne comprenait que quelques balises de mise en forme, les balises de liens et quelques autres aujourd'hui obsolètes.
- Le HTML+, puis le HTML 2.0, sa version officielle, arrivent en 1994 et apportent deux grandes nouveautés : les images et les formulaires.
- La version suivante, de 1996, porte le numéro 3.2. C'est la version la plus utilisée aujourd'hui. Elle comprend la majorité des balises utilisées actuellement et offre, entre autres, les tables, la mise en forme de texte autour d'images ainsi que le support des applets Java.
- La version 4.0, officialisée en décembre 1997, a été corrigée en avril 1998 et décembre 1999. La grande nouveauté de cette version 4 est le support des frames et des feuilles de styles.
- En janvier 2000, le W3C publie les spécifications de XHTML 1.0, le futur langage pour la création de pages Web. Le XHTML, eXtended HyperTextMarkup Language, permettra de définir ses propres balises, comme c'est déjà possible avec le XML, un autre langage de description de documents très puissant.

# Rappels

## Les versions HTML



Pour le W3C, le HTML était fini depuis la version 4. Il commença à développer le XHTML 2, avec le XML pour décor. Une rébellion se dessina au sein du W3C.

Les mécontents formèrent leur propre groupe : le Web Hypertext Application Technology Working Group, ou WHATWG pour faire court.

Au début, le gros travail du WHATWG portait sur deux spécifications : Web Forms 2.0 et Web Apps 1.0. Ces deux spécifications étaient destinées à compléter le HTML. Avec le temps, elles furent regroupées dans une seule spécification, simplement appelée HTML5.

Pendant que le WHATWG développait le HTML5, le W3C continuait à travailler sur le XHTML2.

En 2009, le W3C annonça que la charte du XHTML2 ne serait pas renouvelée.

Le HTML5 est la dernière itération en date.

# Rappels

## Les DTD

HTML4.01 strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

XHTML 1.0 strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

HTML5

```
<!DOCTYPE html>
```

Une déclaration de type de document, ou doctype, est traditionnellement utilisée pour spécifier le type de balisage du document.

Le doctype de l'HTML 4.01 ressemble à ceci :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Ci-dessous, le doctype du XHTML 1.0 :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Ces déclarations sont difficilement compréhensibles, mais elles ne font que dire à leur façon ce document est écrit en HTML 4.01 ou ce document est écrit en XHTML 1.0.

On pourrait penser que le doctype qui déclare ce document est écrit en HTML5 comporte le numéro cinq quelque part, mais ce n'est pas le cas. Le doctype de l'HTML5 ressemble à cela :

```
<!DOCTYPE html>
```

# Rappels

## Les différents composants d'un document HTML

Les bases d'un code HTML5 ressemblent beaucoup à celles rédigées à l'aide des versions précédentes HTML4 et XHTML. On y retrouve un *doctype* suivi des éléments les plus courants :

<html>, <head>, <title>, <meta>, <link>, <script> et <body>.

Un document HTML se compose de trois parties :

- une ligne contenant les **informations de version HTML**;
- une section en-tête déclarative (enclose par l'élément **HEAD**) ;
- un corps, qui emporte le contenu effectif du document. Le corps peut être implémenté par l'élément **BODY** .

Voici un exemple de document HTML simple :

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de la page</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
</head>
<body>
  ...
  <!-- Le reste du contenu -->
  ...
</body>
</html>
```

# Rappels

## Les différents composants d'un document HTML

L'élément racine `<html>` va recueillir les deux principaux éléments de la hiérarchie :

`<head>` et `<body>`.



### Les fameux `<head>` et `<body>`

À ce niveau, le code HTML peut être divisé en deux parties : l'en-tête située entre les balises `<head>` et `</head>`, qui regroupe toutes les méta-informations, c'est-à-dire les données qui ne sont pas représentées directement à l'écran dans le rendu du document, mais qui lui sont tout de même liées : le titre dans `<title>` (extrêmement recommandé), les autres méta-informations variées (facultatives : mots-clés, description générale de la page) dans zéro ou plusieurs `<meta>`, et éventuellement `<link>`, `<script>` ou `<style>`.

La deuxième partie comprend le corps à proprement parler, c'est-à-dire `<body>`. On y retrouvera tout le reste du contenu HTML, structuré par des balises variées - selon ce que l'on aura à y placer - et dont l'apparence sera affectée par les styles CSS chargés dans `<link>` et/ou `<style>`.

# Rappels

## Les commentaires

Un **commentaire** en HTML est un texte qui sert simplement de mémo. Il n'est pas affiché, il n'est pas lu par l'ordinateur, cela ne change rien à l'affichage de la page.

```
1 <!-- Ceci est un commentaire -->
```

Cela sert à *vous* et aux personnes qui liront le code source de votre page. Vous pouvez utiliser les commentaires pour laisser des indications sur le fonctionnement de votre page.

Quel intérêt ? Cela vous permettra de vous rappeler comment fonctionne votre page si vous revenez sur votre code source après un long moment d'absence.

Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page, mais vous pouvez vous en servir pour vous aider à vous repérer dans votre code source.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- En-tête de la page -->
5     <meta charset="utf-8" />
6     <title>Titre</title>
7   </head>
8
9   <body>
10    <!-- Corps de la page -->
11  </body>
12 </html>
```

# Rappels

## La structure composite d'un document HTML

Le HTML peut utiliser, dans un même document, plusieurs vocabulaires de façon indépendante.





# Rappels

## Le modèle de document (DOM)

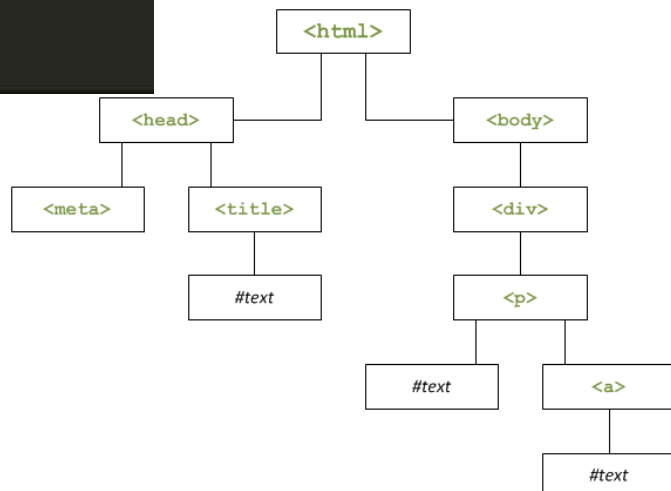
Le *Document Object Model* est une structure de données qui représente un document HTML comme une arborescence.

La racine de cet arbre est un nœud nommé document qui correspond grossièrement à la balise `<html>`.

Les balises HTML définies dans la page Web correspondent aux nœuds de l'arbre DOM et en constituent la structure.

Le langage jQuery est en mesure d'interroger le DOM pour connaître les caractéristiques (attributs et valeurs HTML, propriétés et valeurs CSS) des balises qui constituent un document HTML, mais aussi de modifier ces éléments pour changer l'allure et/ou le contenu du document.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Hello World!</title>
5   </head>
6
7   <body>
8
9     <script>
10
11       alert('Hello world!');
12
13     </script>
14
15   </body>
16 </html>
```



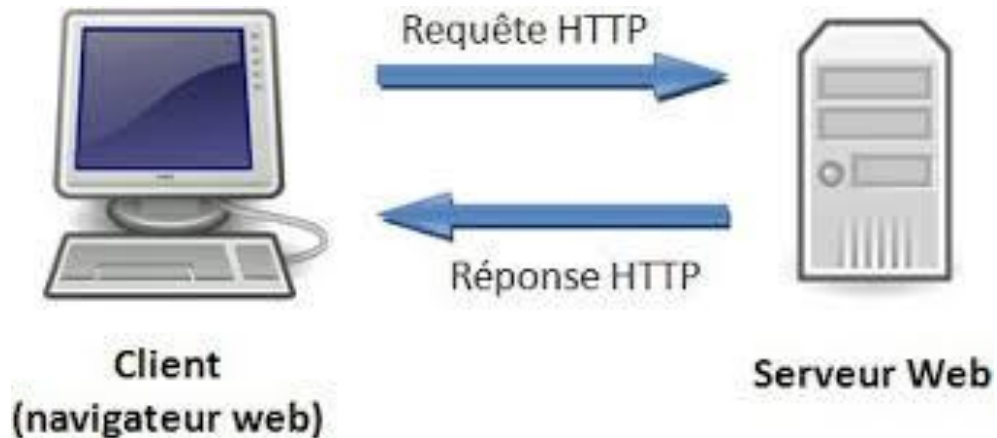


# Rappels

## Le protocole HTTP

Le Hypertext Transfer Protocol, plus connu sous l'abréviation HTTP, le « protocole de transfert hypertexte », est un protocole de communication client-serveur développé pour le World Wide Web.

Il est utilisé pour échanger toute sorte de données entre client HTTP et serveur HTTP.



# Rappels

## Interaction client / serveur HTTP

Un ensemble de lignes de commandes est envoyé au serveur par le biais du navigateur afin d'effectuer une requête HTTP.

La ligne de commande comprend la commande (HEAD, GET, POST...), l'URL (l'adresse de la page sur le serveur) et la version de protocole (la version de HTTP utilisée).

```
GET /fichier.ext HTTP/1.1
Host: www.site.com
Connection: Close
<nouvelle ligne>
```

### Exemple de réponse :

HTTP/1.1 200 OK

Date: Thu, 11 Jan 2007 14:00:36 GMT

Server: Apache/2.0.54 (Debian GNU/Linux) DAV/2 SVN/1.1.4

Connection: close

Transfer-Encoding: chunked

Content-Type: text/html; charset=ISO-8859-1

# Rappels

## Interprétation du HTML par le navigateur

Le navigateur est le programme qui nous permet de voir les sites web. Le travail du navigateur est de lire le code HTML et CSS pour afficher un résultat visuel à l'écran. Si votre code CSS dit « Les titres sont en rouge », alors le navigateur affichera les titres en rouge. Le rôle du navigateur est donc essentiel !

Chaque navigateur interprète le code à sa manière donc il faudra vous y faire et prendre l'habitude de vérifier régulièrement que votre site fonctionne correctement sur la plupart des navigateurs.

Tous les navigateurs, même moderne, ne sont pas en phase pour prendre en compte les nouveautés HTML5.

Tester les différents navigateurs sur [fmbip.com](http://fmbip.com) ou [html5test.com](http://html5test.com) ou [canuise.com](http://canuise.com) afin de s'apercevoir de la prise en charge ou non des nouveautés HTML5.

findmebyIP

browser support revealed

AC033

your IP address

{ 82.234.101.121 }

A free tool from [iWeb and Browser Web Design Agency](#) - Deep Blue Sky

[7 Like](#)

[454](#)

[17 Tweet](#)

[10](#)

[Follow](#)

[3-1](#)

What's my IP?

top

Browser

CSS3

HTML5

Forms 2.0

CSS3 Selectors

Script

IP & Location

Your Browser

What is a Browser?

A "browser" is a piece of software used to access & display the Internet on your computer. Examples include Internet Explorer, Mozilla Firefox, Opera, Safari and

Browser

Browser Version

Operating System



Mozilla Firefox

32



HTML5 TEST how well does your browser support html5?

your browser other browsers compare news device lab about the test

YOUR BROWSER SCORES **471** OUT OF 555 POINTS

HTTP Sniffer  
PROFESSIONAL  
HTTPS  
SNIFFER  
DOWNLOAD  
FREE

You are using Firefox 32.0 on Windows 8.1

Copy ✓ X

Save results Compare to Share Donate

semantics

**Parsing rules** 10

<!DOCTYPE html> triggers standards mode Yes ✓  
 HTML5 tokenizer Yes ✓  
 HTML5 tree building Yes ✓

HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. Support for SVG and MathML is not required though, so

multimedia

**Video** 33/35

<video> element Yes ✓  
 DRM support No X  
 Media Source extensions No X  
 Subtitle support Yes ✓  
 Poster image support Yes ✓