

# Les technologies web

*HTTP://iihm.imag.fr/DEMEURE/Enseignements*

*Alexandre.Demeure@univ-grenoble-alpes.fr*

*2017*

# Le web

Le web est un ensemble de ressources hypertextes.

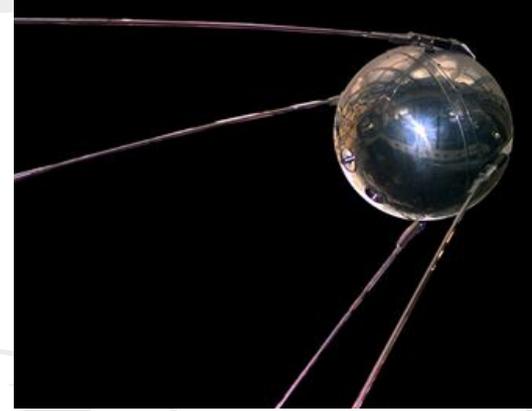
Il peut être vu comme un graphe dont les sommets sont les ressources et les arcs les hyperliens.

Le web se base sur les technologies standards d'internet (TCP, IP) mais n'est qu'une partie d'internet (ex : e-mail, messagerie instantanée, newsgroup, ...)

# Le web

## Petit historique : Émergence des réseaux

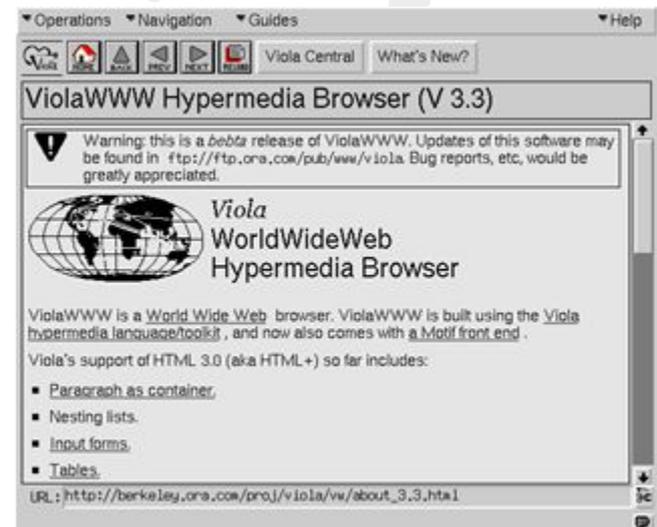
- 1957 : Lancement réussi de Spoutnik
- 1962 : USAF veut un réseau
  - qui résiste à une attaque nucléaire
  - qui permette de riposter
- 1969 : Arpanet
  - Echange de données entre universités
  - Les données suivent un chemin dynamique
- 1971 : le courrier électronique
- 1973-1974 : TCP/IP



# Le web

- 1989 : Naissance du web
  - Développé au CERN
  - Rendre internet plus accessible
  - HTTP : un protocole d'accès aux ressources
  - HTML : un langage de description de documents
  - <http://www.w3.org/History/1989/proposal.html>

...10010100101010101100110...



# Le web

- 1993 : Mosaic
  - Premier navigateur supportant les images
  - Introduction des formulaires
  - => On peut facilement envoyer de l'information à une ressource
- Fragmentation du langage, guerre des navigateurs
- 1997 : HTML 4
  - Nettoyage du langage, mise à niveau
  - Séparation du contenu et de sa présentation => CSS
- 2007-aujourd'hui : HTML 5
  - Document de travail toujours en cours d'évolution
  - Reflète l'évolution des usages
    - Publication de données
    - Usage de formulaire, e-commerce
    - Edition de contenu, partage, wiki, ...
    - Vidéo, youtube, mobilité, ...
    - Applications, google document, ...

# Le web

Dessine moi le web (Serveurs, site, ressource, application, ...)

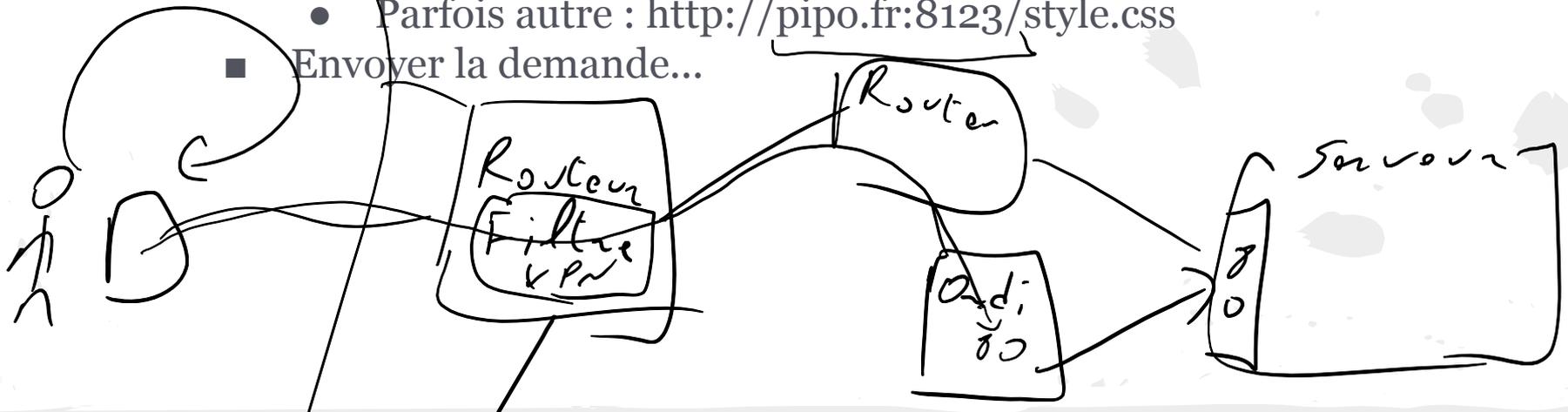
# Le web

Dessine moi le web (Serveurs, site, ressource, application, ...)

# Le web

## Comment accéder aux ressources ?

- Adresse : **http://iihm.imag.fr/demeure/index.html**
  - URL : Uniform Resource Locator
  - Ressource accessible via le protocole **http**
  - Serveur http : iihm.imag.fr
  - Adresse de la ressource sur le serveur : demeure/index.html
- Se connecter au serveur
  - Ouvrir une **socket** TCP
    - Port 80 par défaut
    - Parfois autre : http://pipo.fr:8123/style.css
  - Envoyer la demande...



# Les technologies web

## HTTP

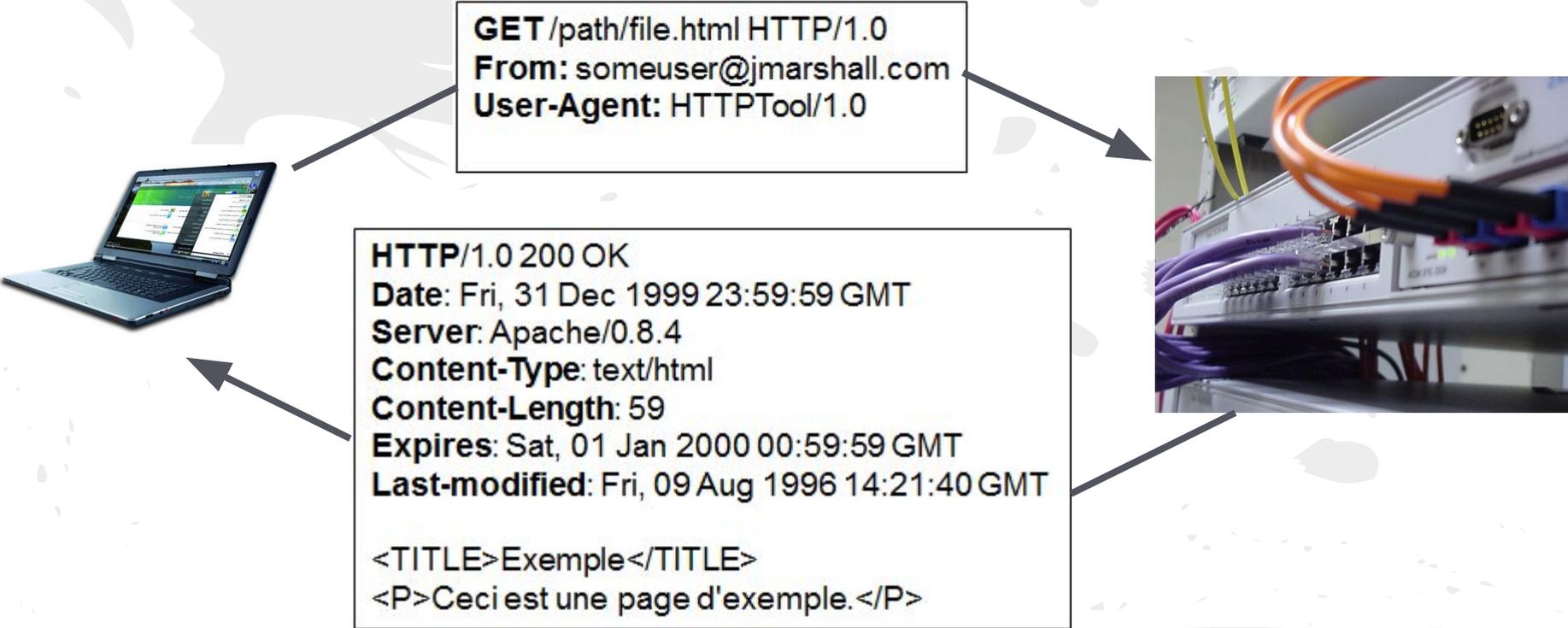
*Alexandre.Demeure@univ-grenoble-alpes.fr*

*2017*

# Le web : protocole HTTP

## Protocole HTTP 1.0

- HyperText Transfert Protocol
- Requêtes client : GET, HEAD, POST



The diagram illustrates the HTTP request cycle. On the left, a laptop represents the client. An arrow points from the laptop to a central box containing the request details. From this box, an arrow points to a server rack on the right, representing the server. A return arrow points from the server back to the laptop, indicating the response.

```
GET /path/file.html HTTP/1.0  
From: someuser@jmarshall.com  
User-Agent: HTTPTool/1.0
```

```
HTTP/1.0 200 OK  
Date: Fri, 31 Dec 1999 23:59:59 GMT  
Server: Apache/0.8.4  
Content-Type: text/html  
Content-Length: 59  
Expires: Sat, 01 Jan 2000 00:59:59 GMT  
Last-modified: Fri, 09 Aug 1996 14:21:40 GMT  
  
<TITLE>Exemple</TITLE>  
<P>Ceci est une page d'exemple.</P>
```



# Le web : protocole HTTP

## Protocole HTTP 1.0 : POST



```
POST /path/file.html HTTP/1.0
From: toto@pipo.org
User-Agent: Mozilla/4.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 40

nom=pauli&prenom=julien&data=some%20data
```



```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Server: Apache/0.8.4
Content-Type: text/html
Content-Length: 59
Expires: Sat, 01 Jan 2000 00:59:59 GMT
Last-modified: Fri, 09 Aug 1996 14:21:40 GMT

<TITLE>Exemple</TITLE>
<P>Ceci est une page d'exemple.</P>
```

# Le web : protocole HTTP

## Codes de réponses

- **1xx** : message d'information
- **2xx** : succès de la requête
- **3xx** : redirection vers une autre ressource
- **4xx** : erreur venant du client
- **5xx** : erreur venant du serveur

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Server: Apache/0.8.4
Content-Type: text/html
Content-Length: 59
Expires: Sat, 01 Jan 2000 00:59:59 GMT
Last-modified: Fri, 09 Aug 1996 14:21:40 GMT

<TITLE>Exemple</TITLE>
<P>Ceci est une page d'exemple.</P>
```

# Le web : protocole HTTP

## HTTP 1.1

- A une adresse peuvent se trouver plusieurs serveurs

```
POST /path/file.html HTTP/1.1  
Host: www.totoland.com  
From: toto@pipo.org  
User-Agent: Mozilla/4.0
```

- Nouvelles entêtes:
  - **Accept-Charset** : encodage des données (UTF-8, ...)
  - **Accept-Language**, ex : fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
- Persistance de la connexion
- Données compressées, gestion du cache, ...
- Mode de transfert de données par fragments

# Le web : protocole HTTP

Et bien d'autres choses encore

- HTTPS
  - Sécurisé avec SSL/TLS => eCommerce
  - Port par défaut : 443
- Autre commandes:
  - **PUT** : Ajouter une ressource
  - **DELETE** : Supprimer une ressource
  - **PATCH** : Modifier une ressource
  - **OPTIONS** : obtenir les options de communications
  - **CONNECT** : Utiliser un tunnel de communication
- ...

# Les technologies web

# HTML

*Alexandre.Demeure@univ-grenoble-alpes.fr*

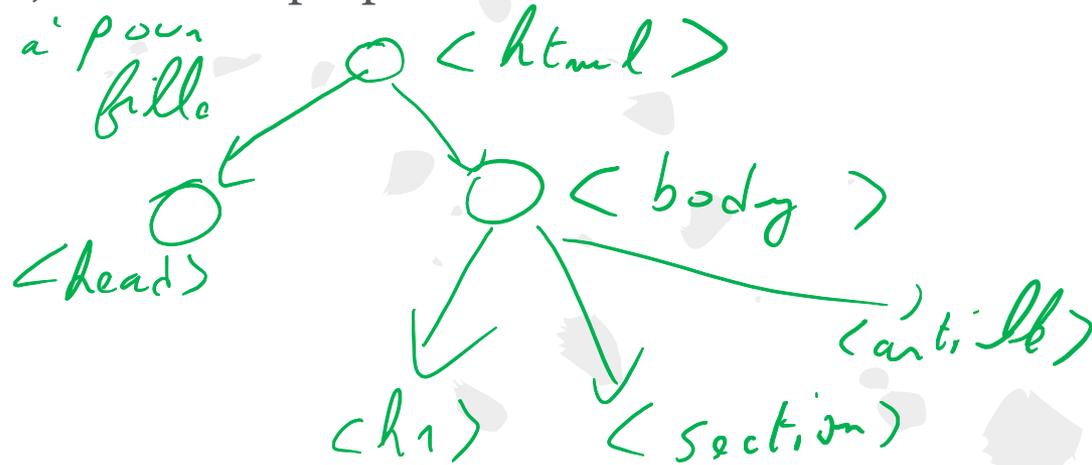
*2017*

# Le langage HTML

## Les balises et leur sémantique

- **html** : ouvre le document
- **head** : description du document
- **body** : corp du document, le contenu proprement dit
- Balises structurantes
- Balises sémantiques

```
<html >  
<head >  
  </head >  
<body >  
  <h1 >  
  <section > - </section >  
  <article > - </article >  
</body >  
</html >
```



# Le langage HTML

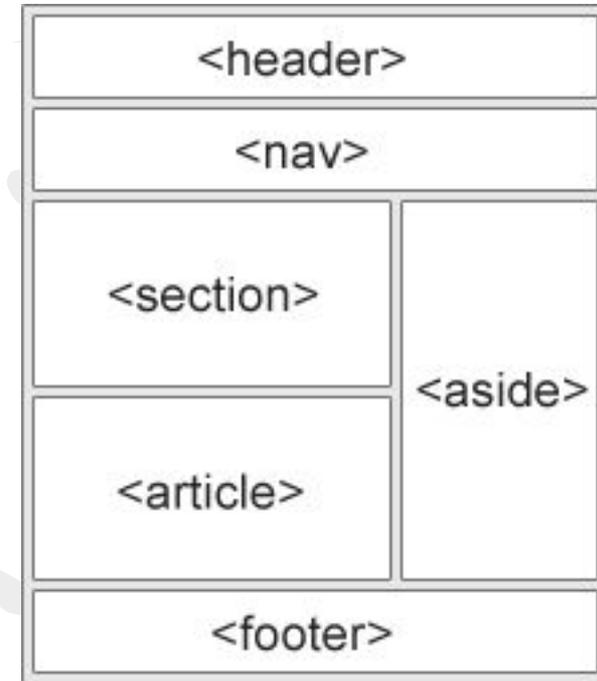
## Les balises et leur sémantique

- Balises structurantes, “non sémantiques”
  - div : définit un bloc
  - span : définit un bloc en ligne

# Le langage HTML

## Les balises et leur sémantique

- Balises sémantiques
  - h1-6 : titres : un paragraphe
  - strong, em : marque l'importance
  - blockquote, q, cite : citations
  - code : extrait de code informatique
  - sup, sub : exposant et indice
  - ul + li : liste non ordonnée
  - ol + li : liste ordonnée
  - table + tr + th + td + caption : tableau
  - details + summary : informations
  - a : un lien vers une autre ressource
  - ...
  - canvas : zone de dessin
  - video + source : vidéo
  - audio : un son, une musique



# Le langage HTML

## Les formulaires

- Permet à l'utilisateur de transmettre de l'information
- Utilisation de la balise form (voir [MDN](#))
- Transfert des données par HTTP

# Le langage HTML

## Les formulaires

- Permet à l'utilisateur de transmettre de l'information
- Utilisation de la balise form (voir [MDN](#))
- Transfert des données par HTTP

# Le langage HTML

## Les formulaires

- Permet à l'utilisateur de transmettre de l'information
- À l'intérieur d'une balise form on a
  - Des attributs
    - action : **URL** ou **mailto**
    - method : **get** ou **post**
    - **enctype**: application/x-www-form-urlencoded, text/plain, multipart/form-data
    - onsubmit, onreset

# Le langage HTML

## Les formulaires

- Permet à l'utilisateur de transmettre de l'information
- À l'intérieur d'une balise form on a
  - Un sous-arbre HTML avec en particulier les balises
    - **input** : une entrée
      - Vérification possible avec expression régulière
      - Attribut pattern  
`pattern="(http|https|ftp)\:\/\/\[a-zA-Z0-9\-\.\ \/]*"`
    - **textarea** : zone de texte multi-lignes
    - **select + optgroup + option**: liste d'options
    - **output** : résultat d'un calcul sur les autres données
    - **meter** : <http://jsbin.com/damuparemu/edit?html,output>
    - **progress** : <http://jsbin.com/xigano/edit?html,output>
    - **datalist** : <http://jsbin.com/teguze/1/edit?html,output>
    - attribut **required**  
=> Pseudo classes CSS (:optional, :required, :valid, :invalid)

# Les technologies web

# CSS

*Alexandre.Demeure@univ-grenoble-alpes.fr*

*2017*

# CSS

## Cascading Style Sheets

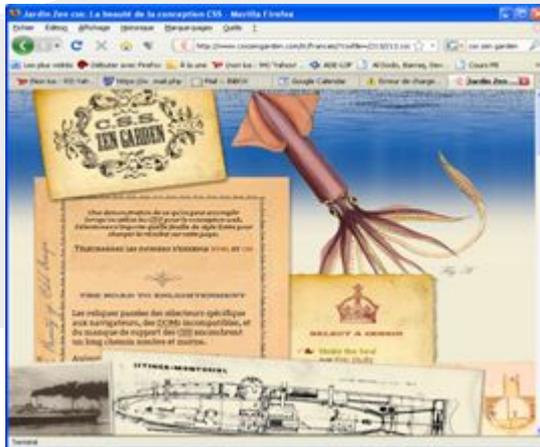
- Mise en page
- Couleurs
- Fontes
- Marges, bordures
- ...

## Bien séparer les préoccupations

- HTML => Structure et sémantique
- CSS => Comment afficher le HTML

# CSS

## Exemple : CSS Zen Garden



# CSS

## Lier HTML et CSS

- Le code HTML référence les CSS
  - A l'intérieur de la balise <head>  
`<link rel="stylesheet" type="text/css" media="screen" href="site.css">`
- CSS définit des règles
  - Qui désignent des éléments HTML : via des **sélecteurs**
  - Qui en modifient les propriétés : via des **attributs**

```
body {  
    margin: 0px;  
    padding: 0px;  
}  
  
article > p {  
    background: red;  
}
```

# CSS

## Les sélecteurs

- #bob : la balise dont l'id est bob
- div : les balises div
- div a : les balises a descendantes d'un div
- div > a : les balises a filles d'un div
- div + a : les balises plus proches sœurs d'un div
- div.toto : les balises div dont une des classes est toto
- div.titi.toto : idem mais en plus une autre est titi
- a[href \$= "pdf"] : les liens (a) qui pointe vers une adresse qui se termine par "pdf"
- ...

## Priorité selon la précision

<http://specificity.keegan.st/>

# CSS

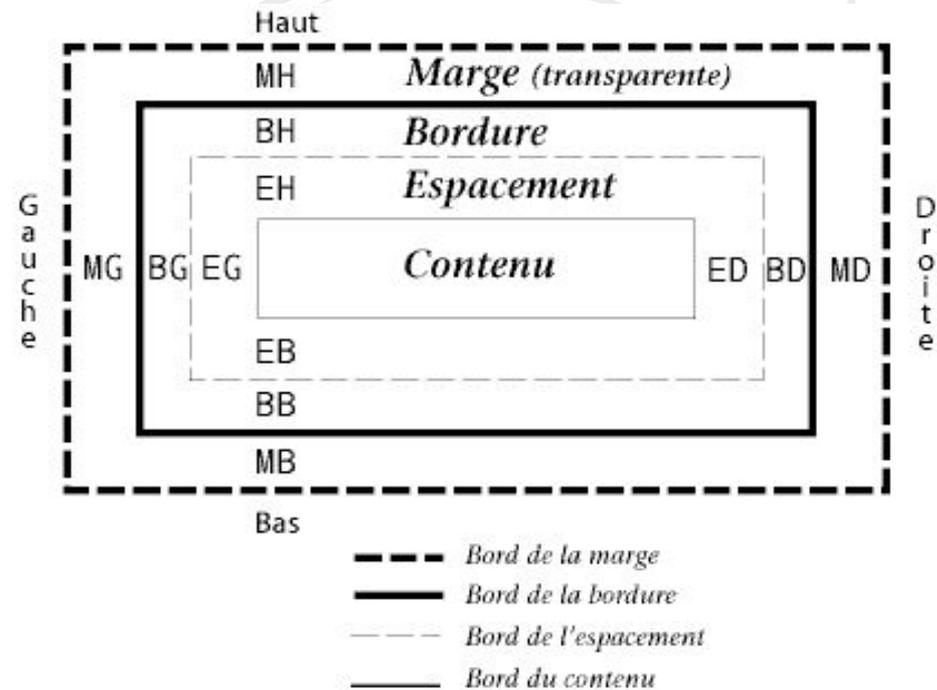
## Les pseudos éléments (::) et pseudo classes (:)

- p::first-line : la première ligne
- p::first-letter : la première lettre
- p::selection : le texte sélectionné
- div::before, div::after : pour ajouter du texte avant ou après
- a:link : les liens pas encore visités
- a:visited : les liens déjà visités
- div:hover : l'élément sur lequel est placé le curseur souris
- input:focus : l'élément qui a le focus
- a:hover:visited : ....
- a:last-of-type : le dernier lien
- tr:nth-child(2n+1) OU tr:nth-child(odd) : les lignes impaires
- tr:nth-child(2n) OU tr:nth-child(even) : les lignes paires
- ...\*:not(FOO) : les élément qui ne sont pas FOO

# CSS

## Les attributs CSS

- Le modèle de la boîte
  - width, height
  - min-width, max-width
  - min-height, max-height
  - margin, margin-left, ...
  - padding, padding\_top, ...
  - border, border-right...
  - border-width, border-top\_width, ...
  - border-color, border-bottom-color, ...
  - border-style, border-top-style, ...
  - Exemples :
    - border : dashed black 3px;
    - padding-left : 5%;
    - margin : 10px;



# CSS

## Les attributs CSS

- Le modèle de la boîte : le fond
  - voir <http://lea.verou.me/css3patterns/>
  - `background-color : #6495ed;`
  - `background-image : url('paper.gif');`
  - `background-repeat : repeat-x;`
  - `background-attachment : fixed;`
  - `background-position : right top;`

# CSS

## Les attributs CSS

- Le texte
  - Color
  - Direction
  - Letter-spacing
  - Line-height
  - Text-align
  - Text-transform (majuscules, minuscules, ...)
  - Vertical-align
  - White-space
  - Word-spacing
  - ...
  - Font-family (Arial, Comic, ...)
  - Font-size
  - Font-style (Normal, italic, oblique)
  - Font-weight (normal, bold)

# CSS

## Les attributs CSS

- La mise en page (layout)
  - display: none, inline, block, flex
  - position: static, absolute, fixed, relative
  - float: none, left, right
  - flex: parties taille\_min ...
  - position : absolute, fixed, relative, static
  - bottom, top, left, right
  - z-index
  - overflow: hidden, scroll, visible

# CSS

## Les attributs CSS

- Les listes
- Les tableaux
- ... Mozilla Web Developer Network, W3C !!!
- Quelques attributs récents
  - transform : Transformation géométrique 2D/3D
  - transition : Interpolation sur des propriétés
  - animation, keyframes : Animations

### Exemples en lignes

- [La Cascade](#)
- [Animate.css](#)
- [Galerie](#)

# Les technologies web

## AJAX et API DOM

*Alexandre.Demeure@univ-grenoble-alpes.fr*

*2017*

# AJAX

Qu'est ce que c'est ?

# AJAX

Asynchronous **J**Avascript and **X**ml

# AJAX

Asynchronous **J**Avascript and **X**ml

# API DOM

Quelques attributs et méthodes des noeuds de l'arbre:

- **textContent** : le contenu sous forme de texte (E/S)
- **innerHTML** : le contenu interprété en HTML (E/S)
- **nodeName** : le nom de la balise (S)
- **parentElement** : le noeud parent du noeud
- **children** : les noeuds enfants du noeud
- **classList** : objet donnant accès à la liste des classes associée au noeud
- ...
- **appendChild( node )** : Ajoute node à la liste des noeuds enfants
- **removeChild( node )** : Supprime node de la liste des noeuds enfants
- **getAttribute(name)** : valeur de l'attribut name du noeud
- **setAttribute(name, value)** : fixe la valeur de l'attribut
- **querySelector( sel )** : premier noeud correspondant au sélecteur CSS sel évalué à partir du noeud
- **querySelectorAll( sel )** : liste des noeuds correspondants au sélecteur CSS sel évalué à partir du noeud
- ...

# API DOM

Quelques attributs et méthodes liés à l'interaction:

- **onclick** : la fonction qui sera appelée en cas de click
- **onmousedown, onmousemove, onmouseup**
- ...
- **addEventListener( eventName, callback)** : appelle callback si l'évènement nommé eventName est déclenché sur le noeud
- **removeEventListener( eventName, callback)** : n'appelle plus callback si l'évènement nommé eventName est déclenché sur le noeud

# Mise en pratique

## Création d'un mini JQuery

- `$( SELECTEUR_CSS )` ou `$( Element )`
- `$(...).each( FONCTION )`
- `$(...).html( CODE_HTML )`
- `$(...).text( TEXTE )`
- `$(...).click( FONCTION )`
- Chaînable : `$(...).html(...).click(...)`