

HTML5 & CSS 3



- HTML5 (tour d'horizon)
 - Contexte
 - Principes généraux
 - Balises structurantes et sémantiques
 - Web Forms 2.0
 - Médias
 - Applications
- CSS 3

3

Quoi de neuf ?

Permissivité

Allègement de l'entête

Nouvelles balises sémantiques

Nouvelles balises audio & vidéo

Amélioration des formulaires

Applications dynamiques

...

HTML5



- Contexte

HTML : HyperText Markup Language

Structuration de documents dédiés aux sites Web

XHTML : eXtensible HyperText Markup
Language

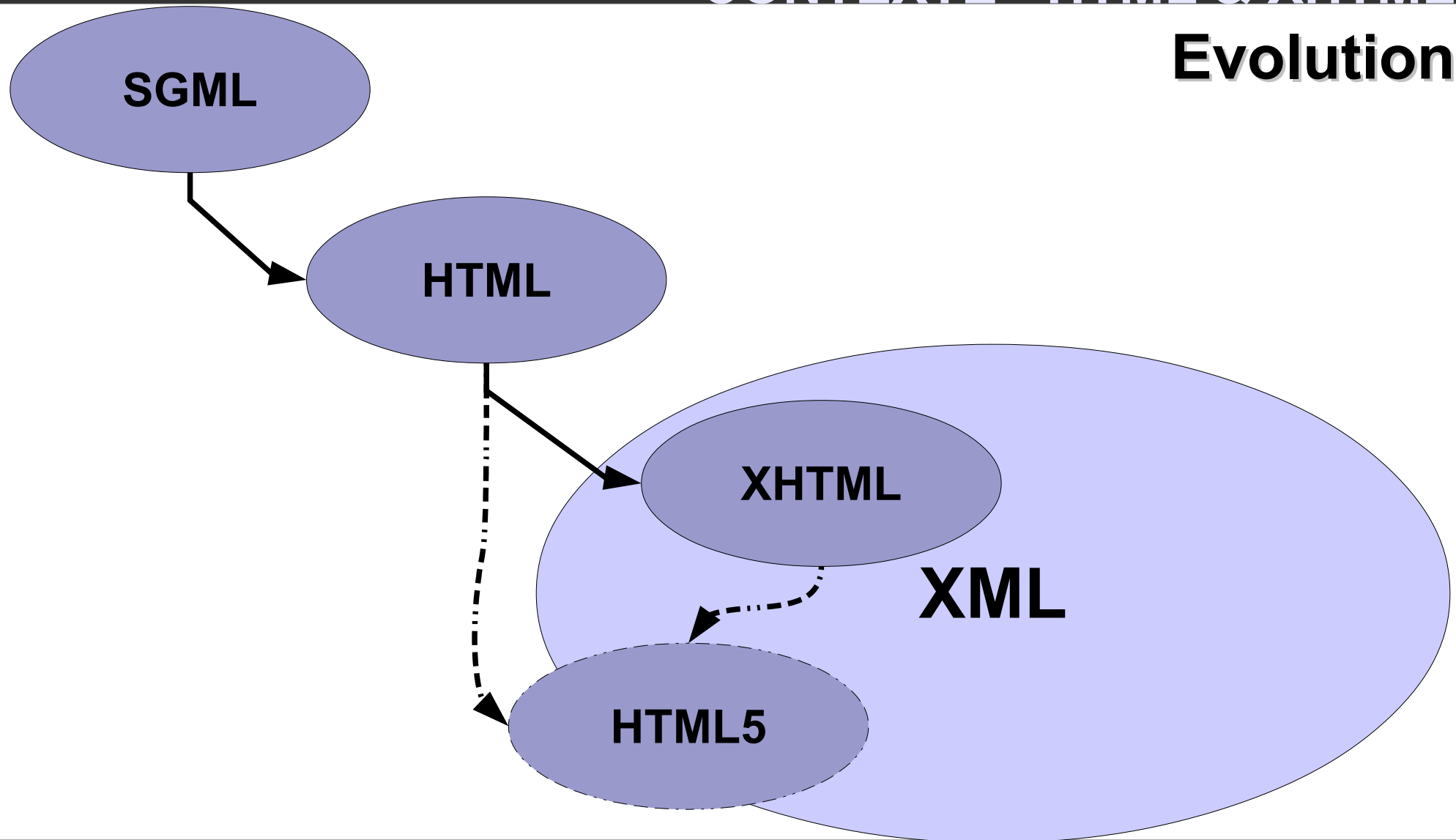
Equivalent à HTML

Mais reformulation de HTML en **XML**

Ecriture plus rigoureuse

CONTEXTE - HTML & XHTML

Evolution

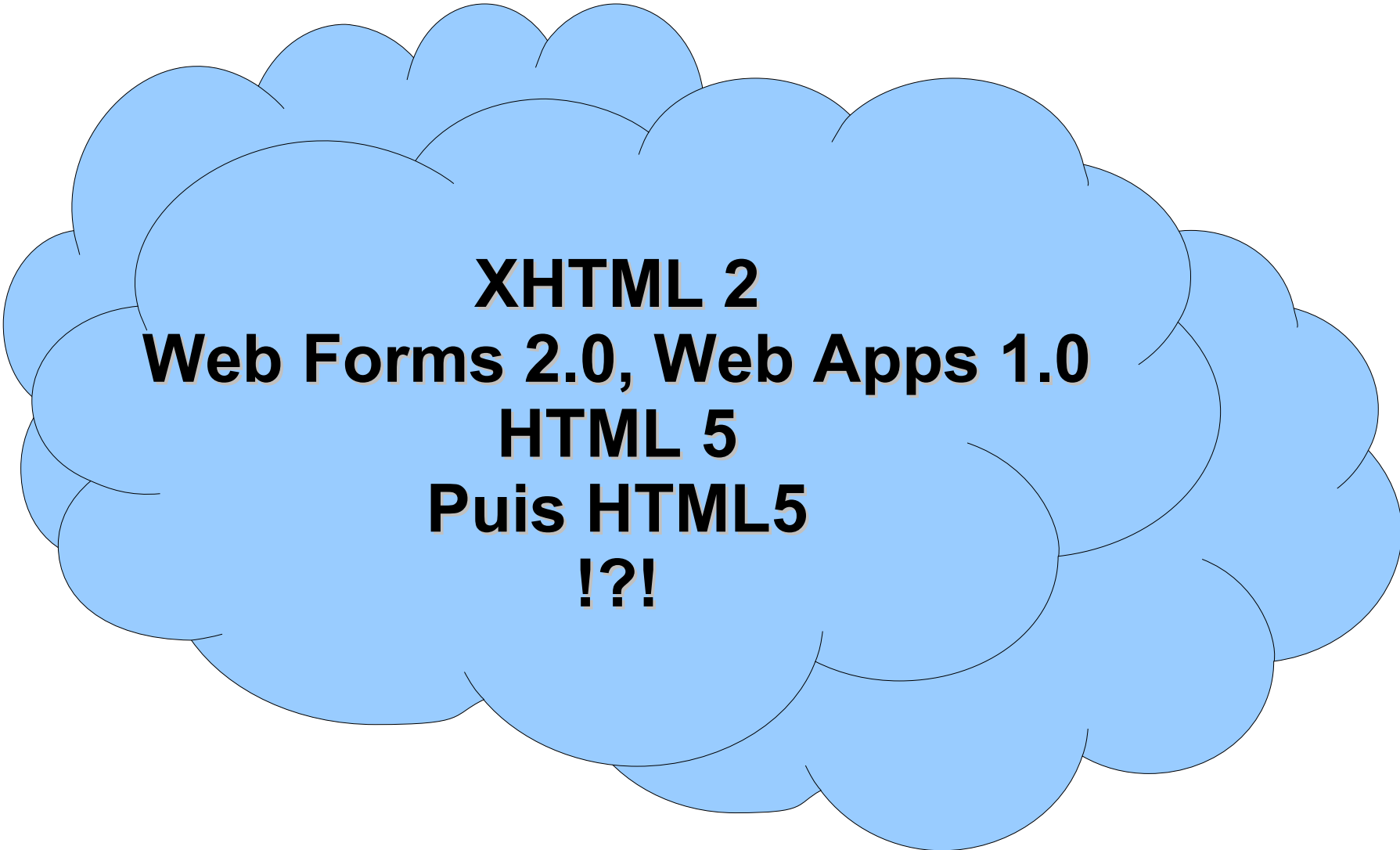


Fichiers au **format texte**

Extension **.html** ou htm

Sous l'autorité du **W3C**
World Wide Web Consortium
<http://www.w3c.org>

Langage non propriétaire



XHTML 2
Web Forms 2.0, Web Apps 1.0
HTML 5
Puis HTML5
!?!

HTML5 est en brouillon

La notion de proposition de recommandation
2 implémentations complètes

Recommandation candidate en 2012 ?
signifie quasi prête

Compatibilité des navigateurs

« Pression » des auteurs

HTML5



- Principes généraux

- Supporter le **contenu existant...**
- ...et les **pratiques existantes**
- La spécification devrait indiquer aux navigateurs
 - Comment **gérer les documents bien formés...**
 - ...et les **documents mal formés**
(cf. gérer les erreurs)
- HTML5 → écrit pour les **utilisateurs, les auteurs et les implémenteurs**

Un doctype que l'on pourra retenir !

<!DOCTYPE html>

Mais qui ne contient plus de versionning ?!?

Cela part du principe que
HTML5 doit supporter le contenu existant (HTML,
XHTML, mais aussi le contenu futur...)

Utiliser le DOCTYPE HTML5
rend un document « standard »
quelle que soit la version d'HTML / XHTML utilisée

D'autres déclarations que l'on pourra retenir

<meta charset="UTF-8">

<script src="script.js" />

<link rel="stylesheet" href="style.css">

Moins de contraintes

Majuscules ou minuscules

Guillemets ou pas

Balises auto-fermantes ou pas

Autres permissivités que je n'ose pas dire

Au choix de l'auteur

A chacun son style

A confirmer

Déprécié devient obsolète

Pour maintenir la compatibilité avec l'existant

**On mentionne les éléments ex dépréciés
dans la spécification**

Pour indiquer au navigateur comment le gérer

Mais ils ne devraient plus être utilisés

Dans le lot : frame (& consorts), acronym (utiliser abbr),
font, big, center, font-size, bgcolor, cellspacing,
cellpadding...

Quelques changements notables ;-)

- **small** ne signale plus un texte en petite taille mais des petits caractères
- **b** ne signifie pas en gras, mais stylistiquement décalé sans avoir une importance supplémentaire (sinon, utiliser **strong**)
- **i** ne signifie pas en italique, mais dans une voix ou une humeur alternative sans emphase ni importance particulière (sinon, utiliser **em**)

HTML5 se veut universel et pas uniquement dédié à des supports **visuels**

Des éléments disparaissent

- frame
- acronym (remplacé par abbr)
- basefont
- big
- center
- ...

→ Des éléments qui étaient souvent déjà abandonnés en XHTML

HTML5



- Balises structurantes et sémantiques
 - header
 - footer
 - nav
 - aside
 - section
 - article
 - figure
- Pour les navigateurs ne supportant pas ces éléments

Nouveaux éléments

- **header** : en-tête ou ouverture de page / section
- **footer** : pied de page / section
- **nav** : informations de navigation (liens principaux)
- **aside** : contenu indirectement apparenté (qui n'est pas indispensable au contenu principal (un « a coté », souvent contextuel))

Ces éléments peuvent concerner le document, mais aussi des articles ou des sections (voir ci-après)

Nouveaux éléments

- **section** : morceau générique de contenu apparenté (de même thème)
- **article** : une section de contenu apparenté, plus spécialisé / autonome (par rapport à une section)
 - Un billet de blog, une réponse dans un fil de discussion, mais aussi un widget météo...
- **figure** : illustration, photos, diagrammes... Associé à figcaption (légende de la figure)

Une section peut contenir des articles et inversement

BALISES STRUCTURANTES ET SEMANTIQUES - HTML5

Pour les navigateurs ne supportant pas ces éléments (inconnus) → IE < 9 notamment

```
<!--[if lt IE 9]>
```

```
    <script src="html5-ietrick.js"></script>
```

```
<![endif]-->
```

En javascript :

```
document.createElement("header");
```

```
document.createElement("footer");
```

etc.

→ Ces éléments sont alors connus dans le DOM

Pour les navigateurs gérant mal ces éléments

Souvent, les navigateurs n'affichent pas ces éléments
comme des blocs,
mais comme des éléments en ligne

→ utiliser les CSS et la propriété `display` pour les utiliser
comme on le souhaite

Par exemple `display : block`

HTML5



- Nouveaux éléments
 - Généraux
 - Audio
 - Video
 - Canvas
 - Éléments de formulaires

Nouveaux éléments

- **time** : date et/ou heure
 - Accompagnée (ou non) de datetime au format ISO AAAA-MM-JJThh:mm:ss (complètement ou non)
 - S'il ny a pas d'attribut datetime, le contenu doit être au format ISO

<time datetime="18:00">18 h</time>

<time datetime="2010-11-15">15 novembre</time>

<time>2010-11-15</time>

Nouveaux éléments

- **meter** : une mesure avec d'éventuels min / max, low, high, optimum et valeur

<meter>23 étudiants</meter>

<meter max="32">23 étudiants</meter>

<meter low="-273" high="100" min="-7" max="22" optimum="12" value="4">Fait pas chaud pour la saison !</meter>

Nouveaux éléments

- **progress** : une valeur qui évolue

```
<progress>50%</progress>
```

```
<progress min="0" max="100"  
  value="50"></progress>
```

Un attribut intéressant : **scoped**

- Cet attribut peut s'appliquer à un élément style de la manière suivante

```
<article>  
  <style scoped>  
    p { color: red; }  
  </style>  
  <p>Paragraphe vert</p>  
</article>
```

Ne s'applique qu'au contenu qui le contient...



Mais actuellement, aucun navigateur ne l'a implémenté

AUDIO

<audio src="fichier.mp3"></audio>

Attributs :

- **autoplay** permet de lancer la lecture immédiatement :-(
- **loop** permet de le jouer en boucle :-(:-(:-(
- **controls** permet de faire apparaître des boutons pour mettre en pause, lire ou régler le volume
- **preload** (none, auto et *metadata*) permet de gérer le préchargement
- Une **API Javascript** permet également d'interagir avec l'élément avec les méthodes play, pause, volume...

AUDIO

- Le format mp3 est généralement supporté (par les navigateurs supportant la balise audio), mais ce n'est pas un format ouvert
- Les fichiers .ogg (généralement décodés avec le codec Vorbis non breveté) ne sont pas supportés par tous les navigateurs supportant la balise audio

- Une solution :

```
<audio>
```

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

```
  <source src="fichier.mp3" type="audio/mpeg">
```

```
</audio>
```

VIDEO

```
<video src="fichier.mp4" width="1600"  
height="900"></video>
```

- Un attribut **poster** permet d'afficher une vignette avant le lancement de la vidéo (poster="image.jpg")
- Les attributs **autoplay**, **loop** et **preload** sont disponibles, ainsi que **controls**
- Une **API Javascript** est disponible, avec la possibilité de faire play, pause, stop...

VIDEO

- Pour la vidéo également, les formats de fichiers peuvent être brevetés (mp4) ou non (Theora Vidéo) :

```
<video>
```

```
  <source src="fichier.ogv" type="video/ogg">
```

```
  <source src="fichier.mp3" type="video/mp4">
```

```
</video>
```

CANVAS

```
<canvas id="monCanvas" width="200"  
height="400"></canvas>
```

- Un « contexte » (bidimensionnel)

```
var can = document.getElementById("monCanvas");  
var ctx = can.getContext("2d");
```
- une **API Javascript** qui permet de dessiner :
 - Des traits, des formes, de gérer le remplissage, des ombres, des courbes, des arcs de cercles, du texte (attention, pas du contenu textuel, mais du dessin)...

CANVAS - REMARQUES

- L'élément **canvas** « sort » du DOM (il n'y a pas de noeuds, textes et attributs)
 - Un problème quand à l'accessibilité
 - Un incohérence avec les principes d'HTML5
- Un groupe de travail réfléchi à la question et à des solutions
- Remarque : quid de SVG ?

Web Forms 2.0

- Nouveaux **types de champs input**
 - **search** (différenciation visuelle d'un champ de saisie standard)
 - **email** (avec format imposé + comportement différent parfois : clavier de saisie sur Safari Mobile par exemple)
 - **url** (avec format imposé + comportement)
 - **tel** (? + comportement)
 - **color** : choix d'une couleur hexadécimale (aucun navigateur)

- Nouveaux **types de champs input**
 - **range** : un slider avec des attributs min et max
 - **number** : un champ de saisie + un menu déroulant (avec valeur numérique imposée ?)
 - **date**
 - **datetime** (incluant le fuseau horaire)
 - **datetime-local** (sans fuseau horaire)
 - **time**
 - **month**
 - **week** (semaine dans l'année)

- **Nouveaux champ datalist**
 - = un hybride entre un champ de type text (saisie libre) et un champ de liste de type select

- Nouveaux attributs
 - **required** : contenu obligatoire
 - **autocomplete** (on / off) : permettre (ou non) au navigateur d'utiliser l'autocomplétion

HTML5



- Nouveaux attributs
 - draggable
 - hidden
 - contenteditable
 - Attributs personnalisés

NOUVEAUX ATTRIBUTS - HTML5

- draggable
 - Nouvel attribut draggable
 - Nécessite Javascript pour gérer le comportement
- hidden
 - Si un élément est hidden → les agents utilisateurs n'en tiennent pas compte → ne l'affichent pas.
- contenteditable
 - Rend une zone éditable
- Attributs personnalisés
 - **data-**monattribut (que l'on pourra manipuler en CSS → voir exemple)

HTML5



- Nouvelles applications
 - Géolocalisation
 - File API et drag & drop
 - IndexedDB
 - Web storage
 - Applications hors ligne
 - Web messaging
 - Push
 - Web workers
 - Web socket

- Géolocalisation
 - API Geolocation
 - Fonctionne avec Javascript et l'objet navigator.geolocation
 - Récupère les coordonnées du poste de l'utilisateur en longitude, latitude et altitude
 - getCurrentPosition
 - watchPosition (suivi en continu)
 - Une série de propriétés
 - Position
 - Vitesse
 - Précision

- File API
 - Sélectionner des fichiers, accéder à leurs données...
 - Améliorer l'upload
 - Recueil d'informations au lancement de l'upload par exemple
 - Accéder au contenu
 - Déplacer, copier, supprimer, écrire...
 - Dans une « sandbox », partie limitée du disque dur (! sécurité)

- Drag & Drop
 - Avec l'attribut draggable
 - Javascript peut gérer les événements liés au clic, au trajet et au « drop » (ondrop)...
- NB : on peut glisser des objets depuis son PC dans le document (→ upload)
 - Drag & Drop + API Data Transfert

- Web Storage
 - API sessionStorage
 - Stockage de données de session
 - Durée de vie courte
 - API localStorage
 - Stockage de données sur la machine cliente
 - Restent stockées même après la fermeture du navigateur
 - = Alternatives aux sessions côté serveur et aux cookies (et sans doute beaucoup plus)
 - = Scripts exécutés sur le client uniquement (sans requêtes au serveur)

NOUVELLES APPLICATIONS - HTML5

- Indexed Database (et Web SQL Database)
 - IndexedDB +/- des catalogues de paires « clé / valeur » indexées
 - Pensés pour JavaScript
 - Pouvoir stocker des données en quantité
 - Pouvoir les trier, les filtrer, les rechercher
 - Web SQL Database
 - Basé sur SQLite
 - (déclaré obsolète en 2010)

- API Offline Web
 - Mise en cache des contenus et des actions hors ligne
 - Synchronisation en ligne dès qu'une connexion est disponible
 - = stocker tous les fichiers nécessaires au fonctionnement d'une application Web même lorsque l'on est déconnecté (stockés dans un « cache »)
 - = stocker les données lorsque l'utilisateur travaille en mode déconnecté
 - = synchroniser les données à la connexion

- Push
 - Événements envoyés par le serveur
 - A l'initiative du serveur
 - SSE : Server-Send Events
 - C'est le client qui initie la connexion et va ensuite rester à l'écoute
 - Le serveur peut alors envoyer des informations au format texte

- Web Messaging
 - Communications entre documents
 - Possibles entre documents de domaines différents
= Cross-document messaging

- Web Sockets
 - Communications par sockets avec un serveur
 - Connexion bidirectionnelle permanente
 - Informations échangées en temps réel
 - A la différence du push
 - Applications :
 - Ce qui nécessite du « temps réel »
 - Chat, jeux, travail collaboratif...

- Web Workers
 - Javascript en multithread
 - Javascript en tâche de fond
 - Non bloquant pour le navigateur
 - Chaque Worker s'exécute dans un thread séparé
 - Pourront être mis à profit pour les traitements lourds

3

CSS 3

- Nouvelles propriétés
 - Contenu
 - Décorations
- Sélecteurs
 - Sélecteur adjacent
 - Sélecteur d'attribut
 - Pseudo-classes & pseudo éléments
- Média Queries
- Transformations & transitions

NOUVELLES PROPRIETES – CSS 3

- word-wrap
 - normal / break-word
- overflow (CSS 2)
 - hidden / scroll / auto
- text-overflow
 - ellipsis (...)
- overflow-x et overflow-y
 - variantes de overflow (CSS 2) sur l'axe des x ou l'axe des y
- resize
 - none, horizontal, vertical, both (cf textarea)

NOUVELLES PROPRIETES – CSS 3

- columns-count
 - nombre de colonnes
- column-width
 - largeur des colonnes
- column-min-width
 - largeur minimale des colonnes
- column-width-policy
 - flexible / strict / inherit
- column-gap
 - distance entre chaque colonne
- column-rule
 - couleur, style et largeur de la séparation

- @font-face
 - polices « exotiques »
- border-radius
 - « arrondir les angles »
- opacity
 - opacité / degré de transparence (entre 0 et 1)
- box-shadow
 - ombre portée
- text-shadow
 - ombre portée sur un texte

- background-size
 - dimensions des images d'arrière-plan (px ou %)
- Arrière-plans multiples
 - background-image: url("1.jpg"), url("2.jpg") ;
 - background-position: left top, right bottom;
 - background-repeat: no repeat;
- + d'autres propriétés sans doute à l'avenir...

- Sélecteur adjacent (CSS 2.1)
 - $E+F \rightarrow$ le frère adjacent direct (successeur dans le code)
- Sélecteur adjacent général
 - $E\sim F \rightarrow$ frère adjacent (pas forcément direct)

- Sélecteur d'attribut (CSS 2)
 - [attribut]
ou
 - [attribut="valeur"]
 - sélection d'éléments par attributs
- CSS 3
 - [attribut^="valeur"]
 - valeur de l'attribut commence par...
 - [attribut\$="valeur"]
 - valeur de l'attribut finit par...
 - [attribut*="valeur"]
 - valeur de l'attribut contient...

- **:lang** (éléments dans une langue donnée)
- **:empty** (éléments vides de tout contenu → `<div></div>`)
- **:root** (racine html → utilité ? Poids supérieur au tag html)
- **:target** (éléments ciblés par des ancres ``)
- **:last-child** (dernier enfant d'un élément dans la fratrie)
 - NB : first-child existe en CSS 2.1

- **:nth-child(x)** (où x est le rang de l'élément dans la fratrie)
 - ou une formule, even (pair) ou odd (impair)
 - exemple
 - div p:nth-child(2)
 - 2e enfant de div, si c'est un p
- **:nth-of-type(x)**
 - Comparable à **:nth-child**
 - Mais, 2e **p** enfant de div (exemple précédent), même si ce n'est pas le 2e enfant
- **:only-child**
 - élément qui n'a pas de frère

- **:only-of-type**
 - élément de type unique dans une fratrie d'éléments
- **:first-of-type**
 - premier élément de son type dans la fratrie
- **:last-of-type**
 - dernier élément de son type dans la fratrie

- **:enabled**
 - élément de formulaire actif
- **:disabled**
 - élément de formulaire inactif
- **:checked**
 - élément de formulaire coché
- **:required**
 - élément de formulaire nécessitant une valeur
- **:optional**
 - élément de formulaire ne nécessitant pas obligatoirement une valeur

- **:valid**
 - élément de formulaire dont la saisie est valide
- **:invalid**
 - élément de formulaire dont la saisie n'est pas valide

Limiter la portée des directives CSS selon les caractéristiques de rendu

- Dans l'inclusion d'une feuille de style
 - `<link rel="stylesheet" href="..." media="screen and (width: 900px)" />`
- Dans une règle `@media`
 - `@media screen and (width: 900px) { ...`
- `and (width: 900px)`
- `and (not width: 900px)`
- `and (min-width: 900px)`
- La , fait office de « ou »

- transform
 - transform: scale(x) où x vaut entre 0 et n
 - une valeur < 1 réduit l'élément
 - une valeur > 1 l'agrandit
 - transform: rotate (xdeg)
 - rotation de x degrés dans le sens des aiguilles d'une montre
 - transform: skew, skewX et skewY
 - étirement sur l'axe des X ou des Y
 - transform: translate
 - translation sur l'axe X et Y
 - transform: matrix
 - transformation à partir d'une matrice (mathématique)

TRANSITIONS – CSS 3

- Principe = transition d'une valeur de propriété vers une autre
 - avec les pseudo-classes CSS
 - en définissant la/les propriété(s) à animer
 - et la durée de l'animation
 - `img {
width: 100px;
transition-property: width;
transition-duration: 2s;
}`
 - `img:hover {
width: 400px;
}`

- transition-property
 - all : toutes les propriétés (possibles) sont concernées
 - une propriété mentionnée
 - none : aucune propriété
- transition-duration
 - en s
 - en ms
- transition-timing-function
 - ease (rapide au début, puis plus lent)
 - linear (vitesse constante)
 - ease-in (lent au début, puis accélère)
 - ease-out (rapide puis décélère)
 - ease-in-out (départ lent, fin lente)

HTML5 & CSS 3



- Période de transition
 - Ne pas attendre les spécifications officielles ?
 - Etre en veille constante
 - Appliquer ce qui peut l'être

3