

HTML5
CSS3
JAVASCRIPT

HTML5

INTRODUCTION

- Successeur d'HTML4 et XHTML 1
- Apporte de nouvelles balises et *fonctionnalités*
 - Éléments structurants (<header>, <footer>, <nav>, ...)
 - Nouveau formulaire (et validation)
 - Support en natif des formats *audio*, et *vidéo*
 - Canvas, SVG, Géolocalisation, drag'n drop, Stockage de données dans le navigateur, ...
- Première proposition formulée en 2008, par le *W3C* et le *WHATWG*

W3C ET WHATWG

- W3C
 - Organisme de formalisation du web
- WHATWG
 - Consortium fondé par Apple, Mozilla, et Opera (en 2004)
 - Rejoint le W3C en 2006

PHILOSOPHIE D'HTML 5

COMPATIBILITÉ

- Support les anciennes balises
- Devoir de proposer des alternatives aux absences de fonctionnalités
- Existence de fonctions pour tester la présence
- Possibilité d'utiliser les nouvelles fonctionnalités sans changer le reste

UTILISATION

- Production du DOM identique sur tous les navigateurs
- Séparation des rôles (HTML : fond, CSS : forme)
- Indépendance vis à vis des médias

SUPPORT D'HTML5

- Comparaison des navigateurs (<http://www.caniuse.com/>)
- Score sur HTML5 des navigateurs (<http://www.html5test.com/>)

APPLICATIONS WEB OU NATIVE ?

POUR

- Disponible sur toutes les plateformes
- Coût de développement faible
 - Langage simple
 - Débogage rapide
- Maintenance faible

CONTRE

- Moins bonne UI
- Performance moindre
- Limité aux navigateurs compatibles

DÉTECTION DE FONCTIONNALITÉS

- Pb de disponibilité de HTML5 => besoin de tester
- Différentes techniques de test
 - Manuel
 - Vérification d'une propriété particulière
 - Création d'un élément et vérification de l'existence d'une propriété
 - Création d'un élément et vérification de l'existence d'une méthode
 - Création d'un élément et vérification de la conservation d'une propriété
 - *Automatique*
 - Utilisation de bibliothèque spécifique ([Modernizr \(http://www.modernizr.com\)](http://www.modernizr.com))

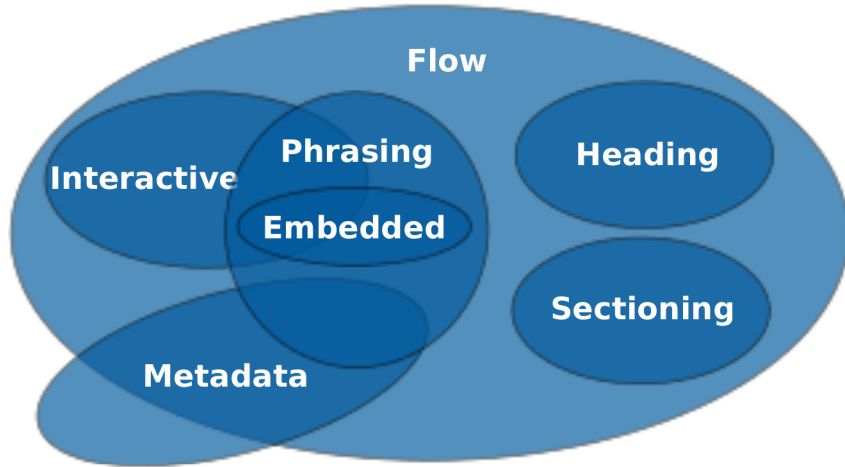
OUTILS DE DÉVELOPPEMENT

- HTML 5 :
 - HTML : éditeur classique
 - CSS : éditeur classique
 - Javascript : éditeur de code
 - Utilisation d'un IDE recommandé pour gagner du temps

STRUCTURE D'UNE PAGE WEB

- Page HTML5 composé :
 - un doctype
 - Avant : `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
 - Maintenant : `<!DOCTYPE html>`
 - un encodage
 - Avant : `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
 - Maintenant : `<meta charset="utf-8">`
 - d'un contenu

BALISES



RETOUR SUR LES BALISES

- Usage du minuscule pour les balises (comme XML)
- Possibilité d'utiliser le couple balise ouvrante et fermante ou auto-fermante
 - `<meta></meta>`
 - `<meta />`
 - Préférence pour la dernière quand cela est possible
- Présence d'attributs pour les balises (utilisation comme XML)
- Présence de booléen pour certaine balise
 - `<input disabled />` :
 - `<input disabled="" />` :
 - `<input disabled="smth" />` :
- Possibilité de mettre des attributs personnalisés
 - doivent commencer par `data-` (utile pour les CSS)

FORME CLASSIQUE D'UN DOCUMENT HTML5

```
<DOCTYPE html>
<html>
  <head>
    <!-- Ici on place les méta données -->
  </head>
  <body>
    <!-- Ici on place le contenu -->
  </body>
</html>
```

ATTRIBUTS GLOBAUX DES BALISES (1/2)

- `accesskey` : permet de spécifier une lettre pour créer un raccourcis clavier
- `class` : permet de spécifier une classe pour être manipuler par la suite
- `contenteditable` : permet de modifier le contenu d'une balise (comme ici)
- `contextmenu` : pour spécifier un menu contextuel (pas encore supporté)
- `dir` : permet de spécifier la direction d'écriture (`rtl` ou `ltr`)
- `draggable` : élément draggable
- `dropzone` : zone de dépôt d'un élément draggable

ATTRIBUTS GLOBAUX BALISES (2/2)

- `hidden` (attribut booléen) : pour "cacher" un élément (toujours visible dans le code source)
- `id` : permet de spécifier un id pour être manipuler par la suite
- `lang` : spécifie la langue de l'objet (utile avec le spellchecker (dans l'idée) ou mise en forme)
- `spellcheck` : pour permettre de vérifier l'orthographe (`lang` n'est pas souvent pris en compte...)
- `style` : définition d'un style particulier
- `tabindex` : pour faciliter la navigation en utilisant la touche Tab
- `title` : information additionnelle (souvent utilisé pour les *tooltips*)

CONSEILS

- Utilisation de balises pour leur sens
- Utilisation du moins de balises possibles
- Être concis et précis sur l'utilisation
- Ne pas faire de présomption sur l'utilisateur (pourrait être un robot)
- Bien comprendre le sens (et l'utilisation des balises)
 - [Explication des balises \(W3\)](http://www.w3.org/TR/html-markup/elements.html) (<http://www.w3.org/TR/html-markup/elements.html>)
 - [W3 School](http://www.w3schools.com/tags/tag_script.asp) (http://www.w3schools.com/tags/tag_script.asp)

LES MÉTA DONNÉES

- **title** : titre de la page
- **base** : Url de référence du site (permet l'utilisation d'URL relative)
(attribut href)
- **meta** :
 - name="author" : auteur de la page
 - name="description" : description de la page
 - name="application name" : nom de l'application
 - name="keywords" : mots clef de la page (séparé par des ,)
 - name="robots" : indication pour les robots (pas toujours supporté)
 - charset : indique l'encodage de la page
 - http-equiv : changer l'entête HTTP (*refresh*, *content-type*, ...)
- **link** : pour les CSS +
 - rel="shortcut icon" href="icon.ico" : spécifie une icone pour la page
 - rel="prefetch" href="page2.html" : permet de précharger une page

CHARGEMENT DE SCRIPT

```
<script src="simple2.js"></script>
```

- Différents modes de chargement
 - Mode par défaut : dès la rencontre de la balise
 - Différé (après le chargement de la page) : `defer` (attribut booléen)
 - Chargement asynchrone : `async` (attribut booléen)

```
<script async defer src="simple2.js"></script>
```

LES BALISES DE TEXTE (1/2)

- a – lien
- em – emphase
- strong – gras
- small – petit
- s – barré
- cite – citation (titre de livre)
- q – citation (phrase)
- dfn – définition
- abbr – abréviation
- time – date and/or time
- code – partie de code
- var – variable ou paramètre
- samp – exemple de sortie

LES BALISES DE TEXTE (2/2)

- `kbd` – entrée clavier
- `sub` – indice
- `sup` – exposant
- `i` – italique (par convention : anglicisme par exemple)
- `b` – gras (par convention)
- `u` – souligné
- `mark` – texte important
- *`ruby`* – *ruby annotation*
- *`rt`* – *ruby texte*
- *`rp`* – *ruby parentheses*
- *`bdi`* – *BiDi isolate*
- *`bdo`* – *BiDi override*
- `span` – span

LES BALISES DE GROUPE

- p – paragraphe
- hr – changement de thème (paragraphe par exemple)
- br – retour de ligne
- pre – texte préformaté
- blockquote – bloc de citation
- ol – liste ordonnée
- ul – liste énumérée
- li – list item
- dl – liste de description
- dt – *definition term or name*
- dd – *definition description or value*
- figure – figure
- figcaption – titre ou légende d'une figure
- div – div

BALISES DE SECTION

- `body` – corps du document
- `section` – section
- `nav` – groupe de liens
- `article` – article
- `aside` – contenu indirect
- `h1` – entête
- `h2` – entête
- `h3` – entête
- `h4` – entête
- `h5` – entête
- `h6` – entête
- `hgroup` – regroupement d'entête
- `header` – entête de section
- `footer` – pied de page de section
- `address` – contact

BALISES DE TABLEAUX

- `table` – tableaux
- `caption` – titre du tableau
- `colgroup` – groupe de colonnes
- `col` – colonne
- `tbody` – contenu du tableau
- `thead` – entête du tableau
- `tfoot` – pied de page du tableau
- `tr` – ligne
- `td` – cellule
- `th` – entête de colonne

BALISES DE FORMULAIRE

- `form` – définition d'un formulaire
- `fieldset` – groupe d'objet d'un formulaire
- `legend` – titre ou légende
- `label` – titre d'un objet
- `input` – entrée
- `button` – bouton
- `select` – combobox
- `datalist` – liste de valeurs prédéfinies
- `optgroup` – groupe d'option (pour le `select`)
- `option` – option
- `textarea` – zone de texte
- `keygen` – génération d'une paire de clef privée/publique
- `output` – résultat d'un calcul (valeurs saisie dans le formulaire)
- `progress` – barre de progression
- `meter` – jauge scalaire (force d'un mdp)

EXEMPLE DE PAGE

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Exemple de titre</title>
</head>
<body>
  <h1>Ceci est un titre de niveau UN</h1>
  <p>Ici on trouve du texte dans un paragraphe</p>
  <p>Voici un <em>second</em> paragraphe</p>
</body>
</html>
```

CE QUI DONNE

Ceci est un titre de niveau UN ← h1

Ici on trouve du texte dans un paragraphe ← p

Voici un *second* paragraphe ← p

↑
em

EXEMPLE DE FORMULAIRE

```
<form method="post">  
  <label>Votre nom : </label><input name="name"/>  
  <button>Ok</button>  
</form>
```

Ce qui donne :

Votre nom :

Ok

BALISE D'OBJET EXTERNE

- `img` – image
- `iframe` – navigateur embarqué
- `embed` – affichage d'applications externes (flash,)
- `object` – objet externe
- `param` – paramètre pour les plugins
- `video` – video
- `audio` – audio
- `source` – media source
- `track` – piste supplémentaire
- `canvas` – canvas
- `map` – image-map definition
- `area` – image-map hyperlink

EXERCICE 1

- Créer une page HTML5 *VALIDE* qui ressemble à :

Login :

Password :

Sexe : Masculin Féminin

Avez vous ?

Portable Tête Chien

Valider

Annuler

EXERCICE 2

- Sans utiliser de CSS, modifier la page précédente afin d'avoir :

Authentification

Login :

Password :

Renseignements supplémentaires

Sexe : Masculin Féminin

Avez vous ?

un portable Une tête Un chien

Valider

Annuler

EXERCICE 3

- Reproduisez la page ci-dessous

Page de démonstration des balises XHTML

Listes

Il est possible d'utiliser des listes

1. Il faut commencer par choisir le bon type
2. Puis créer autant d'éléments que nécessaire
3. Bien évidemment

Mais on peut aussi utiliser un autre type de liste

- Il existe donc deux types de liste
 - les listes énumérées
 - les listes ordonnées
 - mais aussi les liste de définition
- Toute la problématique est de choisir la bonne sémantiquement parlant

Notez que pour l'instant nous ne nous attachons qu'à la *sémantique* des balises.

Nous ne pouvons inclure dans une page HTML directement le symbole <. Pour pouvoir l'utiliser, il faut écrire <. C'est d'ailleurs aussi le cas pour le symbole & ou il faut écrire &.

Ce cadre n'est pas à reproduire, mais il vous donne des explications

Titre de niveau 1
Titre de niveau 2
Définition d'un terme
Vous n'avez pas à utiliser de CSS particulière.