



Support de cours HTML5

Jérôme AMBROISE

Next Formation
Semaine 20/02/2017 - 24/02/2017

Sommaire (1/2)

Introduction

- A. Présentation
- B. Historique
- C. Support des navigateurs

Conception de page HTML5

- A. Les balises
- B. Structure d'une page
- C. Le DOCTYPE
- D. Exemple de contenu <head>
- E. Les nouveaux éléments (balises, APIs)
- F. Les éléments obsolètes

Les bases de l'HTML

- A. Les balises de base
- B. Les commentaires
- C. Les attributs
- D. Exemples d'attributs
- E. Liste d'attributs
- F. Exemple d'un attribut : le style
- G. Les bonnes façons de faire

Les balises usuelles

- A. Les titres
- B. Les paragraphes
- C. Le formatage du texte
- D. Les liens hypertextes
- E. Les images
- F. Les tableaux
- G. Les listes à puces
- H. Les listes de description

Balises sémantiques HTML5

- A. Présentation
- B. Balise <section>
- C. Balise <article>
- D. Balise <header>
- E. Balise <footer>
- F. Balise <nav>
- G. Balise <aside>
- H. Balise <figure>
- I. Balise <details>, <summary>
- J. Balise <main>
- K. Balise <time>

Les formulaires

- A. Présentation
- B. Attributs de formulaires
- C. Text
- D. Password
- E. Textarea
- F. Radio
- G. Checkbox
- H. Sélect
- I. Soumission
- J. Réinitialisation
- K. Label
- L. Fieldset
- M. Datalist
- N. Keygen
- O. Output

Champs de textes particuliers HTML5

- A. Recherche
- B. E-mail
- C. URL
- D. Saisie numérique
- E. Rang
- F. Couleurs
- G. Téléphone
- H. Datetime locale
- I. Date
- J. Mois
- K. Semaine
- L. Heure

Sommaire (2/2)

Les attributs de formulaires

- C. Valeur par défaut
- D. Champ en lecture seule
- E. Champ désactivé
- F. Taille du champs
- G. Taille max de la valeur du champ
- H. Autocomplétion
- I. La “non-validation” d’un formulaire
- J. Champs externe à la balise <form>
- K. Champs de soumission multiple
- L. Changement d’encodage
- M. Changement de méthode
- N. La “non-validation” d’un champ
- O. Cible de la soumission
- P. Dimension d’un champ d’image
- Q. Liste déroulante
- R. Saisie multiple
- S. Expression régulière
- T. Attribute requis
- U. Le pas d’un chiffre

Sommaire (2/2)

Les graphiques HTML5

- A. Canvas :
 - a. Présentation
 - b. Utilisation
 - c. Ligne
 - d. Cercle
 - e. Texte
 - f. Contour de lettre
 - g. Dégradé linéaire
 - h. Dégradé circulaire
 - i. Image
- B. SVG :
 - a. Présentation
 - b. Cercle
 - c. Rectangle
 - d. Rectangle avec bordures arrondies
 - e. Etoile
 - f. Logo
- C. Comparaison entre SVG et Canvas

Les médias HTML5

- A. Présentation
- B. Support des navigateurs
- C. Formats des vidéos
- D. Formats audio
- E. Intégration d'une vidéo
- F. Lecture automatique
- G. Maîtriser les contrôles vidéos
- H. Intégration de l'audio
- I. Plug-ins
- J. L'élément <object>
- K. L'élément <embed>
- L. Youtube
 - a. Présentation
 - b. Intégration
 - c. Autoload
 - d. Boucle
 - e. Contrôles

Les APIs HTML5

- A. Géolocalisation
- B. Drag & Drop
- C. Stockage local
- D. Cache d'application
- E. WebWorker

1.

Introduction

Présentation

Historique

Support des navigateurs

Introduction

Présentation

- HTML, pour “Hypertext Markup Language”
- Langage de base pour la page web
- Prise en main simple
- Page HTML diffusable dans le monde entier

Introduction

Présentation

- Evolution dans le temps
- Complété par d'autres langages :
 - ◆ CSS
 - ◆ JavaScript
 - ◆ Images vectorielles (SVG)

Introduction

Présentation

- Basé sur XML, le HTML ajoute des termes pour définir des textes particuliers (titres, paragraphes, listes, ...)

Introduction

Historique

- 1996 : HTML 2
- De 1997 à 1999 : HTML 3, HTML 4
- 2011 - 2014 : HTML 5

Introduction

Support des navigateurs

- Support des navigateurs récents
- Interprétation des balises gérées par le navigateur
- Se renseigner sur le support de chaque balise et tester avec différents navigateurs (pc, tablette, téléphone)

2.

Conception de page web HTML5

Les balises

Structure d'une page

Le DOCTYPE

Exemple de contenu <head>

Les nouveaux éléments (balises, APIs)

Les éléments obsolètes

Introduction

Les balises

Syntaxe générale d'une balise :

```
<tagname>content goes here...</tagname>
```

- La fermeture des balises n'est pas obligatoire, mais très conseillée (exploitation JS, PHP, parser XML)

Notion de balises orphelines :

- Ne possèdent pas de contenu
- La fermeture de ces balises n'est pas obligatoire non plus, mais conseillée

Introduction

Les balises

Il existe 2 types d'affichage par défaut :

- Inline (exemple "``")
- Block (exemple "`<div></div>`")

Les balises de type "block" occupent une "ligne" à part entière opposé au type d'affichage "inline".

Introduction

Structure d'une page

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title of the document</title>
</head>
<body>
  Content of the document.....
</body>
</html>
```

Introduction

Structure d'une page

Balise	Description
<code><!DOCTYPE html></code>	Déclaration de la page HTML5
<code><html></code>	Balise principale du code HTML
<code><head></code>	Partie non-affichée à l'utilisateur : <ul style="list-style-type: none">→ Metas→ Titre de la page→ Inclusions assets (css, js,img, fonts)
<code><body></code>	Partie affichée à l'utilisateur

Introduction

Différence DOCTYPE HTML4 - HTML5

HTML4 :

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

HTML5 :

```
<!DOCTYPE html>
```


Introduction

Exemple de contenu <head>

Les métas :

```
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<!-- Toujours utile ??? -->
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
<meta name="author" content="John Doe">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Titre et inclusions :

```
<title>HTML5 Syntax and Coding Style</title>
<base href="http://www.w3schools.com/images/" target="_blank">
<link rel="stylesheet" href="mystyle.css">
<script src="monscript.js" ></script>
```

Introduction

Les nouveaux éléments HTML5

Les balises

Les nouveaux éléments HTML5 les plus intéressants sont:

- Nouveaux éléments sémantiques comme `<header>`, `<footer>`, `<article>` et `<section>`
- Nouveaux attributs des éléments de formulaire comme le nombre, la date, l'heure, le calendrier et la plage
- Nouveaux éléments graphiques: `<svg>` et `<canvas>`
- Nouveaux éléments multimédias: `<audio>` et `<vidéo>`.

Introduction

Les nouveaux éléments HTML5

Les APIs

Nouvelles API HTML5 (Interfaces de programmation d'applications)

Les nouvelles API les plus intéressantes en HTML5 sont:

- Géolocalisation HTML
- HTML Drag and Drop
- Stockage Local HTML
- Cache d'application HTML
- HTML Web Worker
- HTML SSE

Introduction

Les éléments obsolètes

Removed Element	Use Instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	
	CSS
<frame> <frameset> <noframes>	
<strike>	CSS, <s>, or
<tt>	CSS

3.

Les bases de l'HTML

Les balises de base

Les commentaires

Les attributs

Exemples d'attributs

Liste d'attributs

Exemple d'un attribut : le style

Les bonnes façons de faire

Les bases

Les balises de base

Les titres

De <h1> à <h6> par ordre de priorités

```
<h1>This is heading 1</h1>
```

Les paragraphes

Les blocs de textes

```
<p>This is heading 1</p>
```

Les liens hypertextes

Faire le lien entre les différentes pages

```
<a href="http://www.w3schools.com">This is a link</a>
```

Les images

Afficher une image

```

```

Les bases

Les commentaires

```
<!-- This is a comment -->
```

```
<!--
```

```
    This is a comment
```

```
→
```

Commentaire conditionnel :

```
<!--[if IE 9]>
    .... some HTML here ....
<![endif]-->
```

Les bases

Les attributs

Tous les éléments HTML peuvent avoir des attributs

- Les attributs fournissent des informations supplémentaires sur un élément
- Les attributs sont toujours spécifiés dans la balise de début
- Les attributs viennent généralement dans des paires nom / valeur comme: name = "value"

```
<p id="presentation" class="article" title="Présentation">  
  <!-- Texte de présentation →  
  ...  
</p>
```


Les bases

Liste d'attributs

Attribut	Description
alt	Spécifie un texte alternatif pour une image, lorsque l'image ne peut pas être affichée
disabled	Spécifie qu'un champs de formulaire doit être désactivé
href	Spécifie l'URL (adresse Web) d'un lien
id	Spécifie un identifiant unique pour un élément
src	Spécifie l'URL (adresse Web) d'une image
style	Spécifie un style CSS en ligne pour un élément
title	Spécifie des informations supplémentaires sur un élément (affiché comme une info-bulle)

Les bases

Exemple d'un attribut : le style

Définir l'attribut de style d'un élément :

```
<h1 style= "color:blue;background-color:red;" > Accueil </h1>
```

Les bases

Les bonnes façons de faire

- Utiliser les minuscules (balises, attributs)
- Fermer les balises
- Éviter les espaces dans les attributs
- Utiliser les double quotes (valeurs d'attributs)

Remarque : dans le cas où la valeur d'un attribut contient des doubles quotes, il est possible d'utiliser les simples quotes.

- 80 caractères de large dans l'éditeur

4.

Les balises usuelles

Les titres

Les paragraphes

Le formatage du texte

Les liens hypertextes

Les images

Les tableaux

Les listes à puces

Les listes de description

Les balises usuelles

Les titres

`<h1>This is heading 1</h1>`

`<h2>This is heading 2</h2>`

`<h3>This is heading 3</h3>`

`<h4>This is heading 4</h4>`

`<h5>This is heading 5</h5>`

`<h6>This is heading 6</h6>`

- Style automatique selon la priorité du titre
- Prise en compte par les robots de référencement

Les balises usuelles

Les paragraphes

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

- Marge automatique entre les paragraphes
- Les espaces et les retours à la ligne ne sont pas pris en compte
- Pas de maîtrise sur le nombre de lignes affichées

Pour les retours à la lignes, utiliser `
` (éviter les abus...)

Pour les textes formatés, préférer la balise `<pre>`

Les balises usuelles

Le formatage du texte

Attribute	Description
Mise en gras	Sans importance sémantique : <code></code> Avec importance sémantique : <code></code>
Mise en italique	Sans importance sémantique : <code><i></i></code> Avec importance sémantique : <code></code>
Texte plus petit	<code><small></small></code>
Texte surligné	<code><mark></mark></code>
Texte supprimé	<code></code>
Texte souligné	<code><ins></ins></code>
Texte superposé	<code><sup></sup></code>
Texte indicé	<code><sub></sub></code>

Les balises usuelles

Le formatage du texte

Attribute	Description
Citations	Block : <code><blockquote></blockquote></code> (indentation) Attribut : <code>"cite"</code> (source) Inline : <code><q></q></code> (apostrophes)
Abréviation / Acronyme	<code><abbr></abbr></code> Attribut : <code>"title"</code> (nom long au survol)
Informations de contact	<code><address></address></code> (mise en italique)
Nom d'une oeuvre	<code><cite></cite></code> (mise en italique)
Substitution bi-directionnelle	<code><bdo></bdo></code> Attribut : <code>"dir"</code> (Valeur = <code>"rtl"</code>)

Les balises usuelles

Le formatage du texte

Attribute	Description
Code source	<code><code> </code></code>
Entrée clavier	<code><kbd> </kbd></code>
Exemple de sortie d'un programme	<code><samp> </samp></code>
Variable (inline)	<code><var> </var></code> (mise en italique)

Pour prendre en compte les espaces et retours à la ligne, entourer ses balises de la balise `<pre>`

Les balises usuelles

Les liens hypertextes

```
<a href="url">link text</a>
```

Permet de créer un lien interne ou externe au site web.

- Lien interne : spécifier le chemin relatif à la page en cours

```
<a href="page/about.html">A propos</a>
```

- Lien externe : spécifier l'URL complet

```
<a href="http://www.google.fr">Google</a>
```

- Lien avec ancre : spécifier l'URL suffixé de l'ID de l'élément à visionner

```
<a href="page/top-articles.html#article4">Article 4</a>
```

Les balises usuelles

Les liens hypertextes

Attribut principal = “href” (hyperlink reference)

Attribute secondaire = “target”

Valeur	Description
_self (par défaut)	Ouvre le lien dans le document actuel
_blank	Ouvre le lien dans une nouvelle page ou un nouvel onglet
_top	Ouvre le lien dans le corps principal

Les balises usuelles

Les images

```

```

Permet d'afficher une image (JPG, PNG, GIF)

Attribut	Description
src	Chemin relatif de l'image
alt	Texte alternatif
width/height	Définir la largeur et la hauteur
style	Définir la largeur et la hauteur (priorité max)

Les balises usuelles

Les images

Aller plus loin avec les images :

→ Image en tant que lien

Balise map :

→ Rendre une zone de l'image cliquable

```
<map name="planetmap">  
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">  
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">  
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">  
</map>
```

Les balises usuelles

Les tableaux

Balise principale : <table>

Balise	Description
<caption>	Légende du tableau
<tr>	Définir une ligne
<th>	Définir une cellule de titre
<td>	Définir une cellule de contenu

Les balises usuelles

Les tableaux

Les attributs de cellules de tableaux

Balise	Description
colspan	Attribuer à une cellule plusieurs colonnes
rowspan	Attribuer à une cellule plusieurs lignes

Les balises usuelles

Les tableaux

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```


Les balises usuelles

Les listes à puces

Liste à puce non ordonnée : ``

Liste à puce ordonnée : ``

Les éléments de puces : ``

➔ Changer le style d'une puce non-ordonnée :

Élément de style => "list-style-type"

Valeurs possibles : disc, circle, square, none

➔ Changer le style d'une puce ordonnée :

Attribut => "type"

Valeurs possibles : 1, A, a, I, i

Les balises usuelles

Les listes à puces

Liste non-ordonnée :

```
<ul style="list-style-type:square">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

Liste ordonnée :

```
<ol type="I">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

Les balises usuelles

Les listes de descriptions

Les listes de description permettent d'ajouter une description à chaque puce.

Balise principale : <dl>

Balise	Description
<dt>	Élément de la liste
<dd>	Description d'un élément de la liste

Les balises usuelles

Les listes de descriptions

Liste de description :

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

5.

Balises sémantiques HTML5

Présentation

Balise <section>

Balise <article>

Balise <header>

Balise <footer>

Balise <nav>

Balise <aside>

Balise <figure>

Balise <details>, <summary>

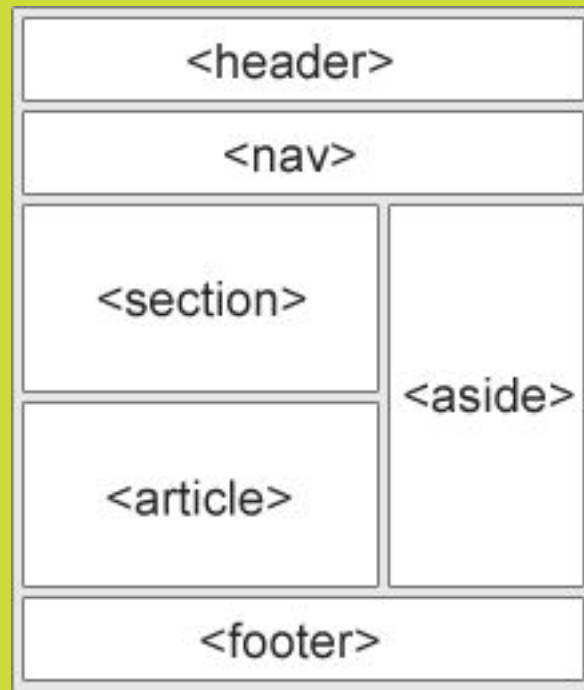
Balise <main>

Balise <time>

Les balises sémantiques HTML5

Présentation

Balise	Description
<header>	En-tête
<nav>	Barre de navigation
<section>	Section d'un document
<article>	Structure indépendante logique
<aside>	Contenu secondaire
<footer>	Pied de page



Les balises sémantiques HTML5

Balise <section>

L'élément <section> définit une section dans un document.

Selon la documentation HTML5 du W3C: "Une section est un groupement thématique de contenu, généralement avec un en-tête."

Une page d'accueil pourrait normalement être divisée en sections pour l'introduction, le contenu et les informations de contact.

Les balises sémantiques HTML5

Balise <article>

L'élément <article> spécifie un contenu indépendant et autonome.

Un article devrait avoir un sens propre, et il devrait être possible de le lire indépendamment du reste du site Web.

Les balises sémantiques HTML5

Balise `<header>`

L'élément `<header>` spécifie un en-tête pour un document ou une section du document.

Balise `<footer>`

L'élément `<footer>` spécifie un pied de page pour un document ou une section du document.

Les balises sémantiques HTML5

Balise `<nav>`

L'élément `<nav>` définit un ensemble de liens de navigation.

L'élément `<nav>` est uniquement destiné au bloc principal de liens de navigation.

Les balises sémantiques HTML5

Balise <aside>

L'élément <aside> définit un contenu en dehors du contenu dans lequel il est placé (comme une barre latérale).

Le contenu de côté doit être lié au contenu environnant.

Les balises sémantiques HTML5

Balise <figure>

Le but d'une légende de chiffre est d'ajouter une explication visuelle à une image.

En HTML5, une image et une légende peuvent être regroupées dans une balise <figure>

```
<figure>  
    
  <figcaption>Fig1. - The Pulpit Rock, Norway.</figcaption>  
</figure>
```

Les balises sémantiques HTML5

Balise <details>, <summary>

La balise <details> spécifie des détails supplémentaires que l'utilisateur peut afficher ou masquer sur demande.

Le contenu d'un élément <details> ne doit pas être visible sauf si l'attribut open est défini.

Remarque :

Incompatible avec Internet Explorer

Les balises sémantiques HTML5

Balise <details>, <summary>

```
<details>
  <summary>Copyright 1999-2014.</summary>
  <p> - by Refsnes Data. All Rights Reserved.</p>
  <p>All content and graphics on this web site are the property of
the company Refsnes Data.</p>
</details>
```

Les balises sémantiques HTML5

Balise <main>

La balise <main> spécifie le contenu principal d'un document. Le contenu de l'élément <main> doit être unique au document.

Il ne doit contenir aucun contenu qui est répété dans les documents tels que les barres latérales, les liens de navigation, les informations sur les droits d'auteur, les logos de site et les formulaires de recherche.

Les balises sémantiques HTML5

Balise <time>

La balise <time> définit une date / heure lisible par l'homme.

Cet élément peut également être utilisé pour encoder des dates et des heures d'une manière lisible par machine afin que les agents utilisateurs puissent proposer d'ajouter des rappels d'anniversaire ou des événements planifiés au calendrier de l'utilisateur et les moteurs de recherche peuvent produire des résultats de recherche plus intelligents.

Les balises sémantiques HTML5

Balise <time>

```
<p>We open at <time>10:00</time> every morning.</p>
```

```
<p>I have a date on <time datetime="2008-02-14 20:00">Valentines  
day</time>.</p>
```

6.

Les formulaires

Présentation

Attributs de formulaires

Type de champs

- Texte
 - ◆ Text
 - ◆ Password
 - ◆ Textarea
- Sélection
 - ◆ Radio
 - ◆ Checkbox
 - ◆ Select
- Soumission
- Réinitialisation
- Présentation
 - ◆ Label
 - ◆ Fieldset
- HTML5
 - ◆ Datalist
 - ◆ Keygen
 - ◆ Output

Les formulaires

Présentation

Balise principale d'un formulaire : `<form></form>`

Attributs :

- “action” : page de traitement (page “.php” par exemple)
- “method” : mode de transmission de l'information (GET, POST)

Un formulaire est un ensemble de champs :

Balise de champs : `<input>` (balise orpheline)

Chaque champs de formulaire doit avoir un attribute “name” pour pouvoir être traité ensuite permettant de l'identifier de façon unique.

Les formulaires

Attributs de formulaires

Balise	Description
accept-charset	Jeu de caractères (par défaut celui de la page)
action	Page de traitement du formulaire
autocomplete	Autocomplétion du navigateur (par défaut "on")
enctype	Spécifie le codage des données soumises (par défaut: "url-encoded")
method	Spécifie la méthode HTTP utilisée lors du dépôt du formulaire (par défaut: GET)
name	Spécifie un nom utilisé pour identifier le formulaire (DOM: document.forms.name)
novalidate	Spécifie que le navigateur ne doit pas valider le formulaire.
target	Spécifie la cible de l'adresse dans l'attribut action (par défaut: _self).

Les formulaires

Type de champs - Champs de texte “text”

Champ de texte simple : `<input type="text">`

Définit un champ de saisie de texte d'une ligne

```
<form>
```

```
  First name:<br>
```

```
  <input type="text" name="firstname">
```

```
</form>
```

Attribute :

→ “value” : valeur par défaut

Les formulaires

Type de champs - Champs de texte “password”

Champ de mot de passe: `<input type="password" />`

Définit un champ de saisie d'un mot de passe

```
<form>
  Mot de passe : <br/>
  <input type="password" name="motdepasse" />
</form>
```

Les formulaires

Type de champs - Champs de texte “textarea”

Zone de texte : `<textarea></textarea>`

L'élément `<textarea>` définit un champ d'entrée multi-lignes (une zone de texte):

```
<textarea name="message" rows="10" cols="30">
```

```
The cat was playing in the garden.
```

```
</textarea>
```

Attributs :

- “rows” : spécifie le nombre visible de lignes
- “cols” : spécifie la largeur visible

Les formulaires

Type de champs - Sélection “radio”

Sélection unique : `<input type="radio" />`

Les boutons radio permettent à un utilisateur de sélectionner UN élément parmi un nombre limité de choix.

```
<form>
  <input type="radio" name="yes" value="oui" checked /> Oui<br/>
  <input type="radio" name="no" value="non" /> Non<br/>
</form>
```

Attributs :

- “value” : valeur associée à la sélection
- “checked” : sélection par défaut (pas de valeur)

Les formulaires

Type de champs - Sélection “checkbox”

Sélection unique : `<input type="checkbox">`

Les cases à cocher permettent à un utilisateur de sélectionner des ZÉRO ou PLUS options d'un nombre limité de choix.

```
<form>  
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>  
  <input type="checkbox" name="vehicle2" value="Car"> I have a car  
</form>
```

Attributs :

→ “value” : valeur associée à la sélection

Les formulaires

Type de champs - Sélection “select”

Liste déroulante : <select></select>

Définit une liste déroulante avec différentes options

→ Chaque option est précisée grâce à la balise
“<option></option>”

Attribut de chaque option : “value” (valeur traitée)

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
</select>
```

Attribuer une valeur sélectionnée par défaut :

Attribut : “selected” (sans valeur)

Les formulaires

Type de champs - Soumission “submit”

Sélection unique : `<input type="submit">`

Permet de soumettre le formulaire

```
<form>  
  <input type="submit" value="Envoyer">  
</form>
```

Attributs :

→ “value” : libellé du champs de validation

Les formulaires

Type de champs - Réinitialisation “reset”

Sélection unique : `<input type="reset">`

Réinitialise toutes les valeurs de formulaire à leurs valeurs par défaut:

```
<form>  
  <input type="reset">  
</form>
```

Les formulaires

Libellé de champs

“label”

Libellé : `<label></label>`

L'élément `<label>` n'affiche pas quelque chose de spécial pour l'utilisateur. Cependant, il fournit une amélioration de la facilité d'utilisation pour les utilisateurs de souris, car si l'utilisateur clique sur le texte dans l'élément `<label>`, il bascule sur le champs associé.

→ L'association est faite grâce à l'attribut “for” du label correspondant à l'attribut “id” du champs concerné.

```
<form>  
  <label for="surnom">Surnom</label>  
  <input type="text" name="surnom" id="surnom">  
</form>
```

Les formulaires

Regroupement de champs “fieldset”

Regroupement : `<fieldset></fieldset>`

Regroupe les données associées dans un formulaire.

→ Balise “legend” : associe une légende au regroupement

```
<form action="action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Les formulaires

Champs HTML5 - Sélection

“datalist”

Liste de sélection: `<datalist></datalist>`

Spécifie une liste d'options prédéfinies .

→ Association de la liste de données avec un champs :

◆ Attribut “list” du champs : `<input list="NOM_UNIQUE">`

◆ Attribute “id” de la “datalist” : `<datalist id="NOM_UNIQUE">`

`<form>`

`<input list="browsers">`

`<datalist id="browsers">`

`<option value="Internet Explorer">`

`<option value="Firefox">`

`<option value="Chrome">`

`</datalist>`

`</form>`

Les formulaires

Champs HTML5 - Générateur de clefs “keygen”

Liste de sélection: <keygen>

L'élément <keygen> spécifie un champ générateur de paires de clés dans un formulaire.

```
<form action="action_page.php">
  Username: <input type="text" name="user">
  Encryption: <keygen name="security">
  <input type="submit">
</form>
```


Les formulaires

Champs HTML5 - Résultat d'un calcul "output"

Liste de sélection: <output>

Représente le résultat d'un calcul (comme celui réalisé par un script)

Attribut particulier :

→ "for" : reprend les valeurs des id de chaque champ <input>

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
```

0

```
<input type="range" id="a" name="a" value="50">
```

100 +

```
<input type="number" id="b" name="b" value="50">
```

=

```
<output name="x" for="a b"></output>
```

```
<input type="submit">
```

```
</form>
```

7.

Champs de texte particuliers HTML5

Recherche

E-mail

URL

Saisie numérique

Rang

Couleurs

Téléphone

Datetime locale

Date

Mois

Semaine

Heure

Champs de texte particuliers

HTML5

Recherche

Est utilisé pour les champs de recherche (un champ de recherche se comporte comme un champ de texte régulier).

```
<form>  
  Search Google:  
  <input type="search" name="googlesearch">  
</form>
```

N'est pas pris en charge dans IE9 et versions antérieures.

Champs de texte particuliers HTML5

E-mail

Utilisé pour les champs de saisie qui doivent contenir une adresse e-mail.

```
<form>  
  E-mail:  
  <input type="email" name="email">  
</form>
```

Non pris en charge dans IE9 et versions antérieures.

Champs de texte particuliers

HTML5

URL

Est utilisé pour les champs d'entrée qui doivent contenir une adresse URL.

```
<form>  
  Add your homepage:  
  <input type="url" name="homepage">  
</form>
```

N'est pas pris en charge dans IE9 et versions antérieures.

Champs de texte particuliers

HTML5

Saisie numérique

Définit un champ de saisie numérique.

```
<form>  
  Quantity (between 1 and 5):  
  <input type="number" name="quantity" min="1" max="5">  
</form>
```

N'est pas pris en charge dans IE9 et versions antérieures.

Champs de texte particuliers

HTML5

Rang

Définit un contrôle pour entrer un nombre dont la valeur exacte n'est pas importante (comme un contrôle de curseur). La plage par défaut est de 0 à 100.

```
<form>  
  <input type="range" name="points" min="0" max="10">  
</form>
```

N'est pas pris en charge dans IE9 et versions antérieures.

Champs de texte particuliers

HTML5

Couleurs

Utilisé pour les champs d'entrée qui doivent contenir une couleur.

```
<form>  
  Select your favorite color:  
  <input type="color" name="favcolor">  
</form>
```


Champs de texte particuliers

HTML5

Téléphone

Est utilisé pour les champs de saisie qui doivent contenir un numéro de téléphone.

```
<form>  
  Telephone:  
  <input type="tel" name="usrtel">  
</form>
```

Actuellement pris en charge uniquement dans Safari 8.

Champs de texte particuliers HTML5

Datetime locale

Spécifie un champ de saisie de date et d'heure, sans fuseau horaire.

```
<form>
  Birthday (date and time):
  <input type="datetime-local" name="bdaytime">
</form>
```

Non pris en charge dans Firefox, ou Internet Explorer 12 et versions antérieures.

Champs de texte particuliers

HTML5

Date

Utilisé pour les champs de saisie qui doivent contenir une date.

```
<form>
  Enter a date before 1980-01-01:
  <input type="date" name="bdaybefore" max="1979-12-31"><br>
  Enter a date after 2000-01-01:
  <input type="date" name="bdayafter" min="2000-01-02"><br>
</form>
```

Non pris en charge dans Firefox, ou Internet Explorer 11 et versions antérieures.

Champs de texte particuliers HTML5

Mois

Permet à l'utilisateur de sélectionner un mois et une année.

```
<form>
  Birthday (month and year):
  <input type="month" name="bdaymonth">
</form>
```

N'est pas pris en charge dans Firefox ou Internet Explorer 11 et les versions antérieures.

Champs de texte particuliers

HTML5

Semaine

Permet à l'utilisateur de sélectionner une semaine et une année.

```
<form>  
  Select a week:  
  <input type="week" name="week_year">  
</form>
```

N'est pas pris en charge dans Firefox ou Internet Explorer 11 et les versions antérieures.

Champs de texte particuliers

HTML5

Heure

Permet à l'utilisateur de sélectionner une heure (pas de fuseau horaire).

```
<form>  
  Select a time:  
  <input type="time" name="usr_time">  
</form>
```

N'est pas pris en charge dans Firefox ou Internet Explorer 12 et versions antérieures.

8.

Les attributs de formulaires

- Valeur par défaut
- Champ en lecture seule
- Champ désactivé
- Taille du champs
- Taille max de la valeur du champ
- Autocomplétion
- La “non-validation” d’un formulaire
- Champs externe à la balise <form>
- Champs de soumission multiple
- Changement d’encodage
- Changement de méthode
- La “non-validation” d’un champ
- Cible de la soumission
- Dimension d’un champ d’image
- Liste déroulante
- Saisie multiple
- Expression régulière
- Attribute requis
- Le pas d’un chiffre

Les attributs de formulaires

Valeur par défaut

L'attribut value spécifie la valeur initiale d'un champ de saisie.

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John">
</form>
```


Les attributs de formulaires

Champ en lecture seule

L'attribut `readonly` spécifie que le champ d'entrée est en lecture seule (ne peut pas être modifié).

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" readonly>
</form>
```

Les attributs de formulaires

Champ désactivé

Un champ de saisie désactivé est inutilisable et impossible à cliquer, et sa valeur ne sera pas envoyée lors du dépôt du formulaire.

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" disabled>
</form>
```

Les attributs de formulaires

Taille du champs

L'attribut size spécifie la taille (en caractères) du champ de saisie.

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" size="40">
</form>
```

Les attributs de formulaires

Taille max de la valeur du champs

L'attribut maxlength spécifie la longueur maximale autorisée pour le champ de saisie.

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" maxlength="10">
</form>
```

Les attributs de formulaires

HTML5 introduit de nombreux autres attributs pour les champs de formulaire.

Les attributs de formulaires HTML5

L'auto-complétion

L'auto-complétion fonctionne sur la balise `<form>` et sur la balise `<input>`.

Champs compatibles : text, search, url, tel, email, password, datepickers, range, and color.

Lorsque l'auto-complétion est activée, le navigateur complète automatiquement les valeurs d'entrée en fonction des valeurs que l'utilisateur a saisies auparavant.

Les attributs de formulaires HTML5

L'auto-complétion

Exemple :

```
<form action="/action_page.php" autocomplete="on">  
  First name:<input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  E-mail: <input type="email" name="email" autocomplete="off"><br>  
  <input type="submit">  
</form>
```

Les attributs de formulaires HTML5

La “non-validation” d’un formulaire

L'attribut novalidate est un attribut <form>.

Lorsqu'il est présent, novalidate spécifie que les données du formulaire ne doivent pas être validées lorsqu'elles sont soumises.

```
<form action="/action_page.php" novalidate>  
  E-mail: <input type="email" name="user_email">  
  <input type="submit">  
</form>
```


Les attributs de formulaires HTML5

La “non-validation” d’un formulaire

L'attribut autofocus spécifie que le champ de saisie doit automatiquement obtenir la mise au point lorsque la page est chargée.

First name:<input type="text" name="fname" autofocus>

Les attributs de formulaires HTML5

Champs externes à la balise `<form>`

L'attribut de formulaire spécifie un ou plusieurs formulaires auxquels appartient un élément `<input>`.

Conseil: Pour faire référence à plus d'un formulaire, utilisez une liste d'ID de formulaire séparée par des espaces.

Les attributs de formulaires HTML5

Champs externes à la balise <form>

Exemple :

```
<form action="/action_page.php" id="form1">  
  First name: <input type="text" name="fname"><br>  
  <input type="submit" value="Submit">  
</form>
```

Last name: <input type="text" name="lname" form="form1">

Les attributs de formulaires HTML5

Champs de soumission multiple

L'attribut formaction spécifie l'URL d'un fichier qui traitera le contrôle d'entrée lorsque le formulaire est soumis.

- L'attribut formaction remplace l'attribut action de l'élément <form>.
- L'attribut formaction est utilisé avec type = "submit" et type = "image".

Les attributs de formulaires HTML5

Champs de soumission multiple

Exemple :

```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formaction="/action_page2.php"  
    value="Submit as admin">  
</form>
```

Les attributs de formulaires HTML5

Changement d'encodage

L'attribut `formenctype` spécifie comment les données de formulaire doivent être encodées lorsqu'elles sont soumises (uniquement pour les formulaires avec `method = "post"`).

- L'attribut `formenctype` remplace l'attribut `enctype` de l'élément `<form>`.
- L'attribut `formenctype` est utilisé avec `type = "submit"` et `type = "image"`.

Les attributs de formulaires HTML5

Changement d'encodage

Exemple :

```
<form action="/action_page_binary.asp" method="post">  
  First name: <input type="text" name="fname"><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formenctype="multipart/form-data"  
    value="Submit as Multipart/form-data">  
</form>
```

Les attributs de formulaires HTML5

Changement de méthode

L'attribut formmethod définit la méthode HTTP pour envoyer des données de formulaire à l'URL d'action.

- L'attribut formmethod remplace l'attribut method de l'élément <form>.
- L'attribut formmethod peut être utilisé avec type = "submit" et type = "image".

Les attributs de formulaires HTML5

Changement de méthode

Exemple :

```
<form action="/action_page.php" method="get">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formmethod="post" formaction="action_page_post.asp"
  value="Submit using POST">
</form>
```

Les attributs de formulaires HTML5

La “non-validation” d’un champs (surcharge <form>)

L'attribut formnovalidate remplace l'attribut novalidate de l'élément <form>.

L'attribut formnovalidate peut être utilisé avec type = "submit".

```
<form action="/action_page.php">  
  E-mail: <input type="email" name="userid"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formnovalidate value="Submit without validation">  
</form>
```

Les attributs de formulaires HTML5

Cible de la soumission

L'attribut `formtarget` spécifie un nom ou un mot-clé qui indique où afficher la réponse qui est reçue après avoir envoyé le formulaire.

- L'attribut `formtarget` remplace l'attribut cible de l'élément `<form>`.
- L'attribut `formtarget` peut être utilisé avec `type = "submit"` et `type = "image"`.

Les attributs de formulaires HTML5

Cible de la soumission

Exemple :

```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit as normal">  
  <input type="submit" formtarget="_blank"  
    value="Submit to a new window">  
</form>
```

Les attributs de formulaires HTML5

Dimensions d'un champ d'image

Les attributs hauteur et largeur spécifient la hauteur et la largeur d'un élément `<input type = "image">`.

```
<input type="image" src="img_submit.gif" alt="Submit" width="48"  
height="48">
```

Les attributs de formulaires HTML5

Liste déroulante

L'attribut list fait référence à un élément <datalist> qui contient des options prédéfinies pour un élément <input>.

```
<input list="browsers">
```

```
<datalist id="browsers">  
  <option value="Internet Explorer">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>
```

Les attributs de formulaires HTML5

Liste déroulante

Les attributs min et max spécifient les valeurs minimum et maximum d'un élément `<input>`.

Les attributs min et max fonctionnent avec les types d'entrée suivants: nombre, plage, date, date-heure locale, mois, heure et semaine.

Les attributs de formulaires HTML5

Liste déroulante

Exemple :

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">
```

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min="1" max="5">
```


Les attributs de formulaires HTML5

Saisie multiple

L'attribut multiple spécifie que l'utilisateur peut entrer plus d'une valeur dans l'élément `<input>`.

L'attribut multiple fonctionne avec les types d'entrée suivants: courrier électronique et fichier.

Les attributs de formulaires HTML5

Saisie multiple

Exemple :

Select images: `<input type="file" name="img" multiple>`

Les attributs de formulaires HTML5

Expression régulière

L'attribut pattern spécifie une expression régulière que la valeur de l'élément `<input>` est cochée.

→ L'attribut pattern fonctionne avec les types d'entrée suivants: texte, recherche, url, tel, email et mot de passe.

Astuce: utilisez l'attribut de titre global pour décrire le modèle pour aider l'utilisateur.

Les attributs de formulaires HTML5

Expression régulière

Exemple :

Country code: `<input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">`

Les attributs de formulaires HTML5

Expression régulière

L'attribut d'espace réservé spécifie un indice qui décrit la valeur attendue d'un champ de saisie (une valeur d'échantillon ou une courte description du format).

- L'astuce s'affiche dans le champ de saisie avant que l'utilisateur n'inscrive une valeur.
- L'attribut d'espace réservé fonctionne avec les types d'entrée suivants: texte, recherche, url, tel, email et mot de passe.

Les attributs de formulaires HTML5

Expression régulière

Exemple :

```
<input type="text" name="fname" placeholder="First name">
```

Les attributs de formulaires HTML5

Attribut requis

L'attribut requis spécifie qu'un champ de saisie doit être rempli avant de soumettre le formulaire.

L'attribut requis fonctionne avec les types d'entrée suivants: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

Username: `<input type="text" name="usrname" required>`

Les attributs de formulaires HTML5

Le pas d'un chiffre

L'attribut step spécifie les intervalles de numéros légaux pour un élément `<input>`.

Exemple: si `step = "3"`, les numéros légaux peuvent être -3, 0, 3, 6, etc.

- Astuce: L'attribut step peut être utilisé avec les attributs max et min pour créer une plage de valeurs légales.
- L'attribut step fonctionne avec les types d'entrée suivants : number, range, date, datetime-local, month, time and week.

Les attributs de formulaires HTML5

Le pas d'un chiffre

Exemple :

```
<input type="number" name="points" step="3">
```

9.

Les graphiques HTML5

Canvas :

- Présentation
- Utilisation
- Ligne
- Cercle
- Texte
- Contour de lettre
- Dégradé linéaire
- Dégradé circulaire
- Image

SVG :

- Présentation
- Cercle
- Rectangle
- Rectangle avec bordures arrondies
- Etoile
- Logo

Comparaison entre SVG et Canvas

Les graphiques HTML5

Canvas - Présentation

Le code HTML `<canvas>` est utilisé pour dessiner des graphiques 2D, à la volée, via JavaScript.

Cet élément est seulement un conteneur pour les graphiques. Il faut utiliser JavaScript pour dessiner les graphiques.

Les graphiques HTML5

Canvas - Utilisation

Une toile est une zone rectangulaire sur une page HTML.
Par défaut, une toile n'a pas de frontière et aucun contenu.

```
<canvas id="myCanvas" width="200" height="100" style="border:1px  
solid #000000;">  
</canvas>
```



Les graphiques HTML5

Canvas - Ligne

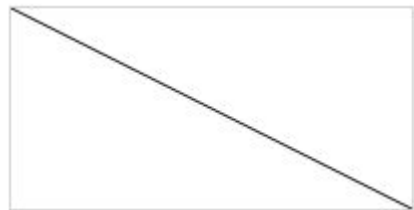
HTML :

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
```

Your browser does not support the HTML5 canvas tag.</canvas>

Javascript :

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.moveTo(0,0);  
ctx.lineTo(200,100);  
ctx.stroke();
```

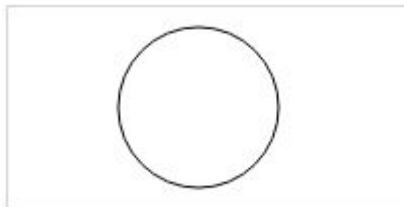


Les graphiques HTML5

Canvas - Cercle

Javascript :

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.beginPath();  
ctx.arc(95,50,40,0,2*Math.PI);  
ctx.stroke();
```



Les graphiques HTML5

Canvas - Texte

Javascript :

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.font = "30px Arial";  
ctx.fillText("Hello World",10,50);
```



Hello World

Les graphiques HTML5

Canvas - Contour de lettre

Javascript :

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.font = "30px Arial";  
ctx.strokeText("Hello World", 10, 50);
```



Hello World

Les graphiques HTML5

Canvas - Dégradé linéaire

Javascript :

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
  
// Create gradient  
var grd = ctx.createLinearGradient(0,0,200,0);  
grd.addColorStop(0,"red");  
grd.addColorStop(1,"white");  
  
// Fill with gradient  
ctx.fillStyle = grd;  
ctx.fillRect(10,10,150,80);
```



Les graphiques HTML5

Canvas - Dégradé circulaire

Javascript :

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
  
// Create gradient  
var grd = ctx.createRadialGradient(75,50,5,90,60,100);  
grd.addColorStop(0,"red");  
grd.addColorStop(1,"white");  
  
// Fill with gradient  
ctx.fillStyle = grd;  
ctx.fillRect(10,10,150,80);
```



Les graphiques HTML5

Canvas - Image

HTML :

```
<p>Image to use:</p>
```

```

```

```
<p>Canvas to fill:</p>
```

```
<canvas id="myCanvas" width="250" height="300"  
style="border:1px solid #d3d3d3;">
```

```
Your browser does not support the HTML5 canvas tag.</canvas>
```

```
<p><button onclick="myCanvas()">Try it</button></p>
```

Les graphiques HTML5

Canvas - Image

JavaScript :

```
function myCanvas() {  
    var c = document.getElementById("myCanvas");  
    var ctx = c.getContext("2d");  
    var img = document.getElementById("scream");  
    ctx.drawImage(img,10,10);  
}
```

Les graphiques HTML5

SVG - Présentation

SVG signifie Scalable Vector Graphics. Il est utilisé pour définir les graphiques pour le Web et est recommandé par W3C.

- L'élément HTML `<svg>` est un conteneur pour les graphiques SVG.
- SVG est basé sur XML, ce qui signifie que chaque élément est disponible dans le DOM de SVG.
- En SVG, chaque forme dessinée est connu comme un objet. Si les attributs d'un objet SVG sont modifiés, le navigateur peut automatiquement recréer la forme.

Les graphiques HTML5

SVG - Exemple : Cercle

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"  
  fill="yellow" />  
</svg>
```



Les graphiques HTML5

SVG - Exemple : Rectangle

```
<svg width="400" height="100">  
  <rect width="400" height="100"  
style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />  
</svg>
```



Les graphiques HTML5

SVG - Rectangle avec bordures arrondies

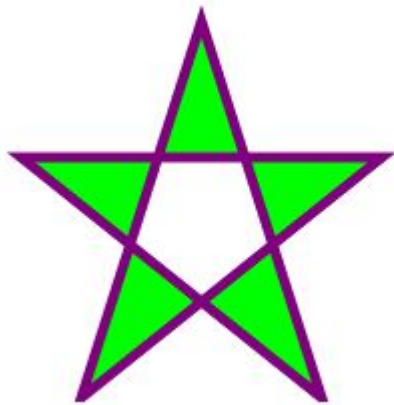
```
<svg width="400" height="180">  
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"  
    style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />  
</svg>
```



Les graphiques HTML5

SVG - Exemple : Étoile

```
<svg width="300" height="200">  
  <polygon points="100,10 40,198 190,78 10,78 160,198"  
  style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;"  
  />  
</svg>
```



Les graphiques HTML5

SVG - Exemple : Logo



```
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1"
x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%"
style="stop-color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%"
style="stop-color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)"
/>
  <text fill="#ffffff" font-size="45" font-family="Verdana"
x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>
```

Les graphiques HTML5

Comparaison entre SVG et Canvas

SVG	Canvas
<ul style="list-style-type: none">→ Indépendant de la résolution→ Prise en charge des gestionnaires d'événements→ Convient mieux aux applications avec de grandes zones de rendu (Google Maps)→ Rendu lent si complexe (tout ce qui utilise beaucoup le DOM)→ Ne convient pas aux applications de jeu	<ul style="list-style-type: none">→ Dépend de la résolution→ Pas de prise en charge des gestionnaires d'événements→ Mauvaise capacité de rendu du texte→ Possibilité d'enregistrer l'image en .png ou .jpg→ Convient aux jeux à forte intensité graphique

10.

Les médias HTML5

Présentation

Support des navigateurs

Formats des vidéos

Formats audio

Intégration d'une vidéo

Lecture automatique

Maîtriser les contrôles vidéos

Intégration de l'audio

Plugins

L'élément <object>

L'élément <embed>

Youtube

- Présentation
- Intégration
- Autoload
- Boucle
- Contrôles

Les médias HTML5

Présentation

Le multimédia est disponible dans de nombreux formats différents. Il peut être presque tout ce que vous pouvez entendre ou voir.

Exemples: Images, musique, sons, vidéos, disques, films, animations, etc.

Les médias HTML5

Support des navigateurs

L'audio, la vidéo et l'animation ont été traités différemment par les principaux navigateurs. Différents formats ont été pris en charge et certains formats nécessitent des programmes d'assistance supplémentaires (plug-ins) pour fonctionner.

Espérons que cela deviendra de l'histoire ancienne. Le multimédia HTML5 promet un avenir plus facile pour le multimédia.

Les médias HTML5

Formats des vidéos

- MP4 est le format nouveau et à venir pour la vidéo sur Internet.
- MP4 est recommandé par YouTube.
- MP4 est pris en charge par les lecteurs Flash.
- MP4 est pris en charge par HTML5.

Les médias HTML5

Formats des vidéos

Format	Extension	
MPEG-4 or MP4	.mp4	Développé par le Moving Pictures Expert Group. Basé sur QuickTime. Généralement utilisé dans les nouvelles caméras vidéo et le matériel de télévision.
WebM	.webm	Développé par les géants du Web, Mozilla, Opera, Adobe et Google.
Ogg	.ogg	Theora Ogg. Développé par la Fondation Xiph.Org.

Les médias HTML5

Formats audio

Format	Extension	
MP3	.mp3	Les fichiers MP3 sont en fait la partie sonore des fichiers MPEG. MP3 est le format le plus populaire pour les lecteurs de musique. Combine une bonne compression (petits fichiers) avec une qualité élevée. Soutenu par tous les navigateurs.
WAV	.wav	Développé par IBM et Microsoft. Joue bien sur les systèmes d'exploitation Windows, Macintosh et Linux.
Ogg	.ogg	Ogg. Développé par la Fondation Xiph.Org.

Les médias HTML5

Intégration d'une vidéo

Avant HTML5, une vidéo ne pouvait être lue que dans un navigateur avec un plug-in (comme un flash).

L'élément HTML5 <video> spécifie un moyen standard d'intégrer une vidéo dans une page Web.

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogv" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

Les médias HTML5

Intégration d'une vidéo

- L'attribut "controls" ajoute des contrôles vidéo, tels que lecture, pause et volume.
- C'est une bonne idée d'inclure toujours les attributs width et height. Si la hauteur et la largeur ne sont pas définies, la page risque de clignoter pendant le chargement de la vidéo.
- L'élément <source> vous permet de spécifier des fichiers vidéo alternatifs que le navigateur peut choisir. Le navigateur utilisera le premier format reconnu.
- Le texte entre les balises <video> et </video> ne sera affiché que dans les navigateurs qui ne prennent pas en charge l'élément <video>.

Les médias HTML5

Lecture automatique

Pour démarrer une vidéo, utiliser automatiquement l'attribut autoplay.

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

Remarque : L'attribut de lecture automatique ne fonctionne pas sur les appareils mobiles comme l'iPad et l'iPhone.

Les médias HTML5

Maîtriser les contrôles vidéos

HTML5 définit les méthodes, les propriétés et les événements DOM pour l'élément `<video>`.

Cela vous permet de charger, de lire et de mettre en pause des vidéos, ainsi que de définir la durée et le volume.

Il existe également des événements DOM qui peuvent vous avertir quand une vidéo commence à jouer, est en pause, etc.

Les médias HTML5

Maîtriser les contrôles vidéos

HTML :

```
<div style="text-align:center">
  <button onclick="playPause()">Play/Pause</button>
  <button onclick="makeBig()">Big</button>
  <button onclick="makeSmall()">Small</button>
  <button onclick="makeNormal()">Normal</button>
  <br><br>
  <video id="video1" width="420">
    <source src="mov_bbb.mp4" type="video/mp4">
    <source src="mov_bbb.ogv" type="video/ogg">
    Your browser does not support HTML5 video.
  </video>
</div>
```

Les médias HTML5

Maîtriser les contrôles vidéos

JavaScript :

```
var myVideo = document.getElementById("video1");
```

```
function playPause() {  
    if (myVideo.paused)  
        myVideo.play();  
    else  
        myVideo.pause();  
}
```

```
function makeBig() { myVideo.width = 560; }
```

```
function makeSmall() { myVideo.width = 320; }
```

```
function makeNormal() { myVideo.width = 420; }
```

Les médias HTML5

Intégration de l'audio

Avant HTML5, les fichiers audio ne pouvaient être joués que dans un navigateur avec un plug-in (comme le flash).

L'élément HTML5 `<audio>` spécifie un moyen standard d'intégrer du contenu audio dans une page Web.

Les médias HTML5

Intégration de l'audio

Exemple :

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

Les médias HTML5

Intégration de l'audio

L'attribut controls ajoute des contrôles audio, tels que lecture, pause et volume.

L'élément <source> vous permet de spécifier des fichiers audio alternatifs que le navigateur peut choisir. Le navigateur utilisera le premier format reconnu.

Le texte entre les balises <audio> et </ audio> ne sera affiché que dans les navigateurs qui ne prennent pas en charge l'élément <audio>.

Les médias HTML5

Plugins

Les applications d'assistance (plug-ins) sont des programmes informatiques qui étendent la fonctionnalité standard d'un navigateur Web.

- Des exemples de plug-ins bien connus sont les applets Java.
- Des plug-ins peuvent être ajoutés aux pages Web avec la balise `<object>` ou la balise `<embed>`.
- Les plug-ins peuvent être utilisés à de nombreuses fins: afficher des cartes, rechercher des virus, vérifier votre identifiant bancaire, etc.

Les médias HTML5

L'élément `<object>`

L'élément `<object>` est pris en charge par tous les navigateurs.

L'élément `<object>` définit un objet incorporé dans un document HTML.

Il est utilisé pour intégrer des plug-ins (comme des applets Java, des lecteurs PDF, des lecteurs Flash) dans des pages Web.

Les médias HTML5

L'élément <object>

Exemple :

```
<object width="400" height="50" data="bookmark.swf"></object>
```

```
<object width="100%" height="500px" data="snippet.html"></object>
```

```
<object data="audi.jpeg"></object>
```

Les médias HTML5

L'élément `<embed>`

L'élément `<embed>` est pris en charge dans tous les principaux navigateurs.

L'élément `<embed>` définit également un objet incorporé dans un document HTML.

Les navigateurs Web ont pris en charge l'élément `<embed>` pendant une longue période. Toutefois, il n'a pas été une partie de la spécification HTML avant HTML5

Les médias HTML5

L'élément <embed>

Exemple :

```
<embed width="400" height="50" src="bookmark.swf">
```

```
<embed width="100%" height="500px" src="snippet.html">
```

```
<embed src="audi.jpeg">
```

Les médias HTML5

Youtube

La façon la plus simple de lire des vidéos en HTML consiste à utiliser YouTube.

- La conversion de vidéos en différents formats peut s'avérer difficile et fastidieuse.
- Une solution plus facile est de laisser YouTube lire les vidéos dans votre page Web.

Les médias HTML5

Youtube

- Transférer la vidéo sur YouTube
- Prendre note de l'identifiant vidéo
- Définir un élément `<iframe>` dans votre page Web
- Laissez l'attribut `src` pointer vers l'URL de la vidéo
- Utilisez les attributs `largeur` et `hauteur` pour spécifier la dimension du lecteur
- Ajoutez d'autres paramètres à l'URL

Les médias HTML5

Intégration Youtube

```
<iframe width="420" height="315"  
src="https://www.youtube.com/embed/XGSy3_Czz8k">  
</iframe>
```

Les médias HTML5

Intégration Youtube

```
<iframe width="420" height="315"  
src="https://www.youtube.com/embed/XGSy3_Czz8k">  
</iframe>
```

Les médias HTML5

Youtube - Autoload

Vous pouvez faire démarrer votre vidéo automatiquement lorsqu'un utilisateur visite cette page en ajoutant un paramètre simple à votre URL YouTube.

Non-recommandé dans la plupart des cas

```
<iframe width="420" height="315"  
src="https://www.youtube.com/embed/XGSy3_Czz8k?autoplay=1">  
</iframe>
```

Valeur	Description
0 (default)	La vidéo ne sera pas lue automatiquement lorsque le lecteur est chargé.
1	La vidéo est lue automatiquement lorsque le lecteur est chargé.

Les médias HTML5

Youtube - Boucle

→ Playlist

→ Boucle

```
<iframe width="420" height="315"  
src="https://www.youtube.com/embed/XGSy3_Czz8k?playlist=XGSy3_Czz8k&loop=1">  
</iframe>
```

Valeur	Description
0 (default)	La vidéo est lue une seule fois.
1	La vidéo sera en boucle (pour toujours).

Les médias HTML5

Youtube - Contrôles

```
<iframe width="420" height="315"  
src="https://www.youtube.com/embed/XGSy3_Czz8k?controls=0">  
</iframe>
```

Valeur	Description
0	Les contrôles du lecteur ne s'affichent pas.
1 (default)	Les commandes du lecteur s'affichent.

11.

Les APIS HTML5

Géolocalisation

Drag & Drop

Stockage local

Cache d'application

WebWorker

Les APIS HTML5

Géolocalisation - Présentation

L'API de géolocalisation HTML est utilisée pour obtenir la position géographique d'un utilisateur.

Comme cela peut compromettre la confidentialité, la position n'est pas disponible à moins que l'utilisateur ne l'approuve.

Remarque: Géolocalisation est la plus précise pour les appareils avec GPS, comme l'iPhone.

Les APIS HTML5

Géolocalisation - Exemple

La méthode `getCurrentPosition()` est utilisée pour renvoyer la position de l'utilisateur.

L'exemple ci-dessous renvoie la latitude et la longitude de la position de l'utilisateur:

```
<script>
var x = document.getElementById("demo");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}
</script>
```

Les APIS HTML5

Géolocalisation - Gestion des erreurs

Le second paramètre de la méthode `getCurrentPosition ()` est utilisé pour traiter les erreurs. Il spécifie une fonction à exécuter si elle ne parvient pas à obtenir l'emplacement de l'utilisateur:

→ `getCurrentPosition(showPosition, showError);`

Les APIS HTML5

Géolocalisation - Gestion des erreurs

```
function showError(error) {  
    switch(error.code) {  
        case error.PERMISSION_DENIED:  
            x.innerHTML = "User denied the request for  
Geolocation."  
            break;  
        case error.POSITION_UNAVAILABLE:  
            x.innerHTML = "Location information is unavailable."  
            break;  
        case error.TIMEOUT:  
            x.innerHTML = "The request to get user location timed  
out."  
            break;  
        case error.UNKNOWN_ERROR:  
            x.innerHTML = "An unknown error occurred."  
            break;  
    }  
}
```

Les APIS HTML5

Géolocalisation - Affichage Google Maps

Pour afficher le résultat dans une carte, vous devez avoir accès à un service de carte, comme Google Maps.

Dans l'exemple ci-dessous, la latitude et la longitude retournées sont utilisées pour afficher l'emplacement dans une carte Google Map (à l'aide d'une image statique):

```
function showPosition(position) {  
    var latlon = position.coords.latitude + "," + position.coords.longitude;  
  
    var img_url = "https://maps.googleapis.com/maps/api/staticmap?center=  
        "+latlon+"&zoom=14&size=400x300&sensor=false&key=YOUR_:KEY;"  
  
    document.getElementById("mapholder").innerHTML = "<img  
src='"+img_url+"'>";  
}
```

Les APIS HTML5

Géolocalisation - Propriétés “getCurrentPosition()”

L'objet de retour de la fonction “getCurrentPosition()” possède plusieurs propriétés intéressantes :

Propriété	Description
coords.latitude	La latitude sous la forme d'un nombre décimal (toujours renvoyé)
coords.longitude	La longitude sous la forme d'un nombre décimal (toujours renvoyé)
coords.accuracy	La précision de la position (toujours retournée)
coords.altitude	L'altitude en mètres au-dessus du niveau moyen de la mer (retournée si disponible)
coords.altitudeAccuracy	La précision d'altitude de la position (retournée si disponible)
coords.heading	Le titre en degrés dans le sens des aiguilles d'une montre à partir du nord (retourné si disponible)
coords.speed	La vitesse en mètres par seconde (retournée si disponible)
timestamp	La date et l'heure de la réponse (retournée si disponible)

Les APIS HTML5

Géolocalisation - Méthodes temps réelles

L'objet Geolocation a également d'autres méthodes intéressantes:

- `WatchPosition ()` - Renvoie la position actuelle de l'utilisateur et continue de retourner la position mise à jour lorsque l'utilisateur se déplace (comme le GPS dans une voiture)
- `ClearWatch ()` - Arrête la méthode `watchPosition ()`.
L'exemple ci-dessous montre la méthode `watchPosition ()`. Vous avez besoin d'un appareil GPS précis pour tester ceci (comme l'iPhone)

Vous avez besoin d'un appareil GPS précis pour tester ceci (comme l'iPhone).

Les APIS HTML5

Géolocalisation - Exemple temps réel

Exemple avec la méthode `watchPosition()`. Vous avez besoin d'un appareil GPS précis pour tester ceci (comme l'iPhone)

```
var x = document.getElementById("demo");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.watchPosition(showPosition);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
        "<br>Longitude: " + position.coords.longitude;
}
```

Les APIS HTML5

Drag & Drop - Présentation

Le glisser-déposer est une caractéristique très courante. C'est lorsque vous "saisissez" un objet et le faites glisser vers un autre emplacement.

Dans HTML5, glisser-déposer fait partie de la norme: tout élément peut être déplaçable.

Les APIs HTML5

Drag & Drop - Exemple

Exemple

Côté HTML :

```
<div id="div1" ondrop="drop(event) "  
ondragover="allowDrop(event) "></div>
```

```

```

Les APIS HTML5

Drag & Drop - Exemple

Exemple, côté JavaScript :

```
function allowDrop(ev) {  
    ev.preventDefault();  
}
```

```
function drag(ev) {  
    ev.dataTransfer.setData("text", ev.target.id);  
}
```

```
function drop(ev) {  
    ev.preventDefault();  
    var data = ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(data));  
}
```

Les APIS HTML5

Drag & Drop - Explications

→ Rendre un élément Draggable

Tout d'abord: Pour rendre un élément draggable, définissez l'attribut draggable sur true:

```
<img draggable="true">
```

→ Que faire glisser - ondragstart et setData ()

Indiquez ensuite ce qui doit se produire lorsque l'élément est déplacé.

- ◆ L'attribut ondragstart appelle une fonction, drag (event), qui spécifie les données à déplacer.
- ◆ La méthode dataTransfer.setData () définit le type de données et la valeur des données glissées

```
function drag(ev) {  
    ev.dataTransfer.setData("text", ev.target.id);  
}
```

Les APIS HTML5

Drag & Drop - Explications

→ Où déposer - ondragover

L'événement ondragover spécifie l'endroit où les données glissées peuvent être supprimées.

Par défaut, les données / éléments ne peuvent pas être supprimés dans d'autres éléments. Pour permettre un déplacement, nous devons empêcher la manipulation par défaut de l'élément.

Cela se fait en appelant la méthode `event.preventDefault()` pour l'événement ondragover:

```
event.preventDefault()
```

Les APIS HTML5

Drag & Drop - Explications

→ Faites le Drop - ondrop

Lorsque les données glissées sont supprimées, un événement de “drop” se produit.

Dans l'exemple ci-dessus, l'attribut ondrop appelle une fonction, drop (event):

```
function drop(ev) {  
    ev.preventDefault();  
    var data = ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(data));  
}
```

Les APIS HTML5

Drag & Drop - Explications

- Appelez `preventDefault ()` pour empêcher la manipulation par défaut du navigateur des données (par défaut est ouvert comme lien à la baisse)
- Obtenez les données glissées avec la méthode `dataTransfer.getData ()`. Cette méthode renverra toutes les données qui ont été définies au même type dans la méthode `setData ()`
- Les données traînées sont l'id de l'élément traîné ("drag1")
- Ajouter l'élément glissé dans l'élément de dépôt

Les APIS HTML5

Stockage local - Présentation

Avec le stockage local, les applications Web peuvent stocker des données localement dans le navigateur de l'utilisateur.

- Contrairement aux cookies, la limite de stockage est beaucoup plus grande (au moins 5 Mo) et les informations ne sont jamais transférées sur le serveur.
- Le stockage local est par origine (par domaine et protocole). Toutes les pages, d'une origine, peuvent stocker et accéder aux mêmes données.

Les APIS HTML5

Stockage local - Différents stockages

Le stockage local HTML fournit deux objets pour stocker des données sur le client:

- `Window.localStorage` - stocke les données sans date d'expiration
- `Window.sessionStorage` - stocke les données d'une session (les données sont perdues lorsque l'onglet du navigateur est fermé)

Avant d'utiliser le stockage local, vérifiez le support du navigateur pour `localStorage` et `sessionStorage`:

```
if (typeof(Storage) !== "undefined") {  
    // Code for localStorage/sessionStorage.  
} else {  
    // Sorry! No Web Storage support..  
}
```


Les APIS HTML5

Stockage local - localStorage

L'objet localStorage stocke les données sans date d'expiration. Les données ne seront pas supprimées lorsque le navigateur est fermé et sera disponible le jour, la semaine ou l'année suivant.

```
// Enregistrement
```

```
localStorage.setItem("lastname", "Smith");
```

ou

```
localStorage.lastname = "Smith";
```

```
// Récupération
```

```
document.getElementById("result").innerHTML =
```

```
localStorage.getItem("lastname");
```

ou

```
document.getElementById("result").innerHTML =
```

```
localStorage.lastname;
```

Les APIS HTML5

Stockage local

Remarque:

Les paires nom / valeur sont toujours stockées sous forme de chaînes. N'oubliez pas de les convertir à un autre format si nécessaire!

Les APIS HTML5

Stockage local - Compteur de clic

Exemple :

```
if (localStorage.clickcount) {  
    localStorage.clickcount = Number(localStorage.clickcount) + 1;  
} else {  
    localStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the button " +  
localStorage.clickcount + " time(s).";
```

Les APIS HTML5

Stockage local - sessionStorage

L'objet sessionStorage est égal à l'objet localStorage, sauf qu'il stocke les données pour une seule session. Les données sont supprimées lorsque l'utilisateur ferme l'onglet spécifique du navigateur.

Exemple précédent (stockage session) :

```
if (sessionStorage.clickcount) {  
    sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;  
} else {  
    sessionStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the button " +  
sessionStorage.clickcount + " time(s) in this session.";
```

Les APIS HTML5

Cache d'application - Présentation

HTML5 introduit le cache d'application, ce qui signifie qu'une application Web est mise en cache et accessible sans connexion Internet.

Le cache d'application offre à l'application trois avantages:

- Navigation hors ligne : les utilisateurs peuvent utiliser l'application lorsqu'ils sont hors ligne
- Vitesse : les ressources mises en cache sont chargées plus rapidement
- Charge de serveur réduite : le navigateur ne télécharge que les ressources mises à jour / modifiées depuis le serveur

Les APIS HTML5

Cache d'application - Attribut <html>

Pour activer le cache d'application, incluez l'attribut manifeste dans la balise <html> du document:

```
<!DOCTYPE HTML>
```

```
<html manifest="demo.appcache">
```

```
...
```

```
</html>
```

Chaque page avec l'attribut manifeste spécifié sera mise en cache lorsque l'utilisateur le visite. Si l'attribut manifest n'est pas spécifié, la page ne sera pas mise en cache.

L'extension de fichier recommandée pour les fichiers de manifeste est: ".appcache"

Les APIS HTML5

Cache d'application - Fichier manifest

Le fichier manifeste est un fichier texte simple, qui indique au navigateur ce qu'il faut mettre en cache (et ce qu'il ne faut jamais mettre en cache).

Le fichier manifeste comporte trois sections:

- **CACHE MANIFEST** - Les fichiers listés sous cet en-tête seront mis en cache après leur premier téléchargement
- **NETWORK** - Les fichiers répertoriés sous cet en-tête nécessitent une connexion au serveur et ne seront jamais mis en cache
- **FALLBACK** - Les fichiers répertoriés sous cet en-tête spécifient les pages de secours si une page est inaccessible

Les APIS HTML5

Cache d'application - CACHE MANIFEST

La première ligne, CACHE MANIFEST, est requise:

```
CACHE MANIFEST
```

```
/theme.css
```

```
/logo.gif
```

```
/main.js
```

Le fichier manifeste ci-dessus répertorie trois ressources: un fichier CSS, une image GIF et un fichier JavaScript. Lorsque le fichier manifeste est chargé, le navigateur téléchargera les trois fichiers à partir du répertoire racine du site Web.

Ensuite, chaque fois que l'utilisateur n'est pas connecté à Internet, les ressources seront toujours disponibles.

Les APIS HTML5

Cache d'application - NETWORK

La section NETWORK ci-dessous spécifie que le fichier "login.asp" ne doit jamais être mis en cache et ne sera pas disponible en mode hors connexion:

NETWORK:

Login.asp

Un astérisque peut être utilisé pour indiquer que toutes les autres ressources/fichiers nécessitent une connexion Internet:

NETWORK:

*

Les APIS HTML5

Cache d'application - FALLBACK

La section FALLBACK ci-dessous spécifie que "offline.html" sera servi à la place de tous les fichiers placés dans le dossier "/html/", au cas où une connexion Internet ne peut pas être établie:

FALLBACK:

/html/ /offline.html

Les APIS HTML5

Cache d'application - Mise à jour

Une fois qu'une application est mise en cache, elle reste mise en cache jusqu'à l'apparition d'une des situations suivantes:

- L'utilisateur efface le cache du navigateur
- Le fichier manifeste est modifié
- Le cache d'application est mis à jour par programme

Les APIS HTML5

Cache d'application - Mise à jour

Exemple - Fichier complet de manifeste de cache

CACHE MANIFEST

2012-02-21 v1.0.0

/theme.css

/logo.gif

/main.js

NETWORK:

login.asp

FALLBACK:

/html/ /offline.html

Les APIS HTML5

Cache d'application - Mise à jour

Astuce:

- Les lignes commençant par un "#" sont des lignes de commentaire, mais peuvent également servir un autre but.
- Le cache d'une application n'est mis à jour que lorsque son fichier manifeste est modifié. Si vous modifiez une image ou modifiez une fonction JavaScript, ces modifications ne seront pas remises en cache.
- La mise à jour de la date et de la version dans une ligne de commentaire est une façon de rendre le navigateur ré-cache vos fichiers.

Les APIS HTML5

Web Worker - Présentation

Qu'est-ce qu'un Web Worker?

Lors de l'exécution de scripts dans une page HTML, la page ne répond plus jusqu'à ce que le script soit terminé.

Un web worker est un JavaScript qui s'exécute en arrière-plan, indépendamment des autres scripts, sans affecter les performances de la page. Vous pouvez continuer à faire ce que vous voulez: en cliquant, en sélectionnant les choses, etc., pendant que le Web worker fonctionne en arrière-plan.

Les APIS HTML5

Web Worker - Vérification

Avant de créer un travailleur Web, vérifiez si le navigateur de l'utilisateur le prend en charge:

```
if (typeof(Worker) !== "undefined") {  
    // Yes! Web worker support!  
    // Some code.....  
} else {  
    // Sorry! No Web Worker support..  
}
```

Les APIS HTML5

Web Worker - Fichier Web Worker

Maintenant, créons notre web worker dans un JavaScript externe. Ici, nous créons un script qui compte. Le script est stocké dans le fichier “demo_workers.js” :

```
var i = 0;
function timedCount() {
    i = i + 1;
    postMessage(i);
    setTimeout("timedCount()",500);
}
timedCount();
```

- La partie importante du code ci-dessus est la méthode `postMessage()` - qui est utilisée pour envoyer un message à la page HTML.
- Normalement, les utilisateurs Web ne sont pas utilisés pour de tels scripts simples, mais pour des tâches plus intensives en CPU.

Les APIS HTML5

Web Worker - Objet Web Worker

Maintenant que nous avons le fichier Web worker, nous devons l'appeler à partir d'une page HTML.

Les lignes suivantes vérifient si le travailleur existe déjà, sinon - il crée un nouvel objet Web worker et exécute le code dans "demo_workers.js":

```
if (typeof(w) == "undefined") {  
    w = new Worker("demo_workers.js");  
}
```

Les APIs HTML5

Web Worker - Objet Web Worker

Ensuite, nous pouvons envoyer et recevoir des messages du Web worker.

Ajoutez un écouteur d'événement "onmessage" au travailleur Web.

```
w.onmessage = function(event){  
    document.getElementById("result").innerHTML = event.data;  
};
```

Lorsque le travailleur Web publie un message, le code dans l'écouteur d'événements est exécuté. Les données du web worker sont stockées dans event.data.

Les APIS HTML5

Web Worker - Objet Web Worker

Lorsqu'un objet de travail Web est créé, il continuera à écouter les messages (même après que le script externe est terminé) jusqu'à ce qu'il soit terminé.

Pour mettre fin à un travailleur Web et aux ressources du navigateur/ordinateur, utilisez la méthode `terminate()`:

```
w.terminate();
```

Les APIS HTML5

Web Worker - Objet Web Worker

Lorsqu'un objet de travail Web est créé, il continuera à écouter les messages (même après que le script externe est terminé) jusqu'à ce qu'il soit terminé.

Pour mettre fin à un travailleur Web et aux ressources du navigateur/ordinateur, utilisez la méthode `terminate()`:

```
w.terminate();
```

Les APIS HTML5

Web Worker - Exemple (côté HTML)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>Count numbers: <output id="result"></output></p>
```

```
<button onclick="startWorker()">Start Worker</button>
```

```
<button onclick="stopWorker()">Stop Worker</button>
```

```
</body>
```

```
</html>
```

Les APIS HTML5

Web Worker - Exemple (côté JavaScript interne)

```
<script>
var w;

function startWorker() {
  if(typeof(Worker) !== "undefined") {
    if(typeof(w) == "undefined") {
      w = new Worker("demo_workers.js");
    }
    w.onmessage = function(event) {
      document.getElementById("result").innerHTML = event.data;
    };
  } else {
    document.getElementById("result").innerHTML = "Sorry! No Web Worker support.";
  }
}

function stopWorker() {
  w.terminate();
  w = undefined;
}
</script>
```

Les APIS HTML5

Web Worker

Puisque les agents Web sont dans des fichiers externes, ils n'ont pas accès aux objets JavaScript suivants:

- L'objet fenêtre
- L'objet de document
- L'objet parent

Merci de votre attention !

=)