



HTML5

TABLE DES MATIÈRES

Chapitre 1 - HTML5.....	5
1.1 - Introduction.....	6
1.1.1 - Présentation.....	6
1.1.2 - Avertissement concernant ce document.....	7
1.1.3 - Quelques nouveautés d'HTML5.....	8
1.1.4 - Références.....	9
1.1.5 - Warning ! Quels navigateurs implémentent HTML5 ?.....	10
1.1.6 - Le nouveau modèle du W3C.....	11
1.1.7 - La structure standard d'un document HTML.....	12
1.1.8 - Le DOCTYPE.....	13
1.1.9 - L'élément HTML.....	14
1.1.10 - Le head.....	15
1.2 - favicon.....	16
1.3 - Les nouveaux éléments de macro-structures.....	17
1.3.1 - Header.....	18
1.3.2 - Footer.....	18
1.3.3 - Nav.....	18
1.3.4 - Section.....	18
1.3.5 - Article.....	18
1.3.6 - Aside.....	18
1.3.7 - Exemple de structuration basique de page.....	20
1.4 - Les éléments pour les medias.....	23
1.4.1 - Image.....	24
1.4.2 - Audio.....	25
1.4.3 - Video.....	26
1.5 - Les nouveaux types input.....	27
1.5.1 - email.....	27
1.5.2 - url.....	27
1.5.3 - number.....	28
1.5.4 - range.....	29
1.5.5 - date pickers.....	30
1.5.6 - color.....	31
1.5.7 - tel.....	33
1.5.8 - search.....	33
1.5.9 - Exemple mixte.....	34
1.5.10 - Quelques écrans d'erreurs.....	39
1.6 - Les inputs date.....	41
1.6.1 - Les sous-types.....	41
1.6.2 - Exemple.....	42
1.6.3 - Initialisation d'un champ date à la date du jour.....	46
1.6.4 - Contrôle de dates et des heures.....	48
1.7 - Les nouveaux éléments de formulaire.....	51
1.7.1 - datalist.....	51
1.7.2 - output.....	53
1.7.3 - mark.....	56
1.7.4 - meter.....	57
1.7.5 - progress.....	59
1.7.6 - time.....	60
1.7.7 - keygen.....	61
1.8 - Les nouveaux attributs des éléments de formulaire.....	63
1.8.1 - Les attributs.....	63
1.8.2 - Exemple.....	64
1.9 - Le Drag and Drop.....	69
1.9.1 - Présentation.....	69
1.9.2 - Syntaxes.....	70
1.9.3 - Code.....	71
1.10 - Le stockage de données côté client.....	74
1.10.1 - Présentation.....	74
1.10.2 - Mise en place.....	75
1.11 - Le stockage local (localStorage).....	76

1.11.1 - Présentation.....	76
1.11.2 - Tester la disponibilité du LocalStorage.....	77
1.11.3 - Syntaxes.....	79
1.11.4 - Exemple de base.....	80
1.11.5 - La console de Chrome.....	80
1.11.6 - Exemple plus	82
1.11.7 - Avec Firefox.....	86
1.11.8 - La console d'Internet Explorer.....	87
1.11.9 - Parser le LocalStorage.....	88
1.12 - Session Storage.....	89
1.12.1 - Présentation.....	89
1.12.2 - Exemple.....	90
1.12.3 - Scripts.....	91
1.12.4 - La console de Chrome.....	95
1.13 - Créer une application offline.....	96
1.13.1 - Principes.....	96
1.13.2 - Démarche.....	97
1.13.3 - Exemple.....	98
1.13.4 - La console de Chrome.....	101
1.13.5 - Exercice : diaporama automatique OnLine et OffLine.....	102
1.14 - Web Database.....	103
1.14.1 - Avertissement.....	103
1.14.2 - Objectif.....	103
1.14.3 - La console de Chrome.....	104
1.14.4 - Syntaxes.....	105
1.14.5 - Quelques particularités de SQLITE.....	106
1.14.6 - Exemple.....	107
1.14.7 - Exercice.....	110
1.15 - Indexed Database (indexedDB).....	111
1.15.1 - Présentation.....	111
1.15.2 - Objectif.....	112
1.15.3 - Syntaxes.....	113
1.15.4 - Exemple.....	121
1.15.5 - Exercices.....	133
1.16 - FileStorage.....	134
1.17 - Tester l'implémentation de HTML5 et agir en conséquence.....	135
1.17.1 - Tester l'implémentation de HTML5.....	135
1.17.2 - Pallier l'absence d'un attribut.....	140
1.17.3 - Pallier l'absence d'un élément.....	146
1.18 - Canvas.....	151
1.18.1 - Définition.....	151
1.18.2 - Quelques méthodes.....	152
1.18.3 - Quelques attributs.....	153
1.18.4 - Un rectangle avec les couleurs par défaut.....	154
1.18.5 - Rectangles en couleurs.....	156
1.18.6 - Lignes.....	157
1.18.7 - Points.....	159
1.18.8 - Triangles.....	161
1.18.9 - Arcs et cercles.....	164
1.18.10 - Ecrire du texte.....	168
1.18.11 - Insérer une image.....	171
1.18.12 - Animation : petite introduction.....	173
1.18.13 - Un histogramme statique.....	182
1.18.14 - Un histogramme dynamique.....	185
1.18.15 - Exercices au choix.....	188
Chapitre 2 - ANNEXES.....	190
2.1 - Le site TP.....	191
2.2 - Les balises HTML.....	192
2.3 - Une page pas à pas.....	195
2.3.1 - Phase 1 : La macro structure et les textes.....	195
2.3.2 - Phase 2 : le style.....	197
2.4 - Les événements HTML.....	199
2.4.1 - Les événements de niveau Page.....	199

2.4.2 - Les événements de niveau formulaire.....	200
2.4.3 - Événements clavier.....	201
2.4.4 - Événements souris.....	202
2.4.5 - Événements media.....	203
2.5 - Divers.....	204
2.5.1 - Redirection HTML.....	204
2.5.2 - Détecter la source de la requête : desktop ou mobile.....	204
2.5.3 - Le maquetage.....	205
2.5.4 - Logiciel de miniaturisation d'images.....	206
2.5.5 - Miniature cliquable.....	207
2.6 - La géolocalisation.....	208
2.7 - HTML5 et quelques navigateurs Linux (Les dernières versions).....	209

CHAPITRE 1 - HTML5

1.1 - INTRODUCTION

1.1.1 - Présentation

HTML5 ... est la dernière version de l'HyperText Markup Language, le langage du World Wide Web.

Pour l'historique cf l'excellent ouvrage de Jeremy Keith aux éditions Dunod.

1.1.2 - Avertissement concernant ce document

Ce document présente quelques nouveautés de HTML5 par rapport à HTML4.

Les balises qui n'ont pas subi de modification ne sont pas présentées.

Donc si vous êtes novice en HTML vous devrez consulter le document HTML.doc.

1.1.3 - Quelques nouveautés d'HTML5

Une syntaxe simplifiée (détachée de XML, l'auto-fermeture n'est pas obligatoire, certaines valeurs d'attribut ne sont pas obligatoires, ...).

Les nouvelles balises sémantiques : header, section, article, nav, aside, footer, ...

Les nouvelles relations des liens : rel='archives', rel='sidebar', rel='tag', ...

Les microdatas : itemprop='name', itemprop='nationality', ...

L'attribut data-.

Une interface plus riche (éléments date, compteurs ou spinners, curseurs ou sliders, ...).

Les nouveaux champs de formulaires : range, autofocus, placeholder, menu, ...

Des éléments de saisie contrôlés (éléments url, email, ...).

Des attributs nouveaux pour de meilleurs contrôles de saisie (min, max, ...).

Les balises média : audio et video.

L'élément Canvas.

Les nouveaux sélecteurs : getElementsByClassName, querySelector, querySelectorAll, ...

Le stockage web ou local : localStorage, sessionStorage, ...

La possibilité de sauvegarder l'état avec un stockage local pour continuer à travailler en offline avec un site (<html manifest="/cache.manifest">).

Les BD Web SQL.

Les BD Web NoSQL.

Les API de cache pour les apps.

Les notifications.

Le drag and drop.

La géolocalisation.

Et puis :

les WebWorkers,

les WebComponents avec les Polymers,

les WebSockets,

le WebGL.

où l'on voit que HTML5 est fortement imbriqué avec JavaScript car, de Canvas jusqu'à la géolocalisation, sans JavaScript point de salut !

1.1.4 - Références

<http://www.w3.org/html/> pour les bases officielles.

<http://www.w3.org/html/wiki/Learn> un wiki officielles pour HTML.

<http://diveintohtml5.org/> pour un document hors normes.

<http://www.w3schools.com/html5/default.asp> pour une documentation d'accès simple.

<http://www.ibm.com/developerworks/library/x-html5/> pour des comparaisons HTML 4 et HTML5.

<http://dev.w3.org/html5/spec/Overview.html> pour les spécifications ... réservée aux pros !

1.1.5 - Warning ! Quels navigateurs implémentent HTML5 ?

Quasiment seul Opera implémente HTML5 (En août 2010).

Google Chrome implémente quelques éléments.

Pour les autres navigateurs (FX, IE, Safari, ...) c'est pschitt!!!

Pour Firefox il faut attendre la version 4.x prévue fin 2010.

Pour IE il faut attendre la version 9.x.

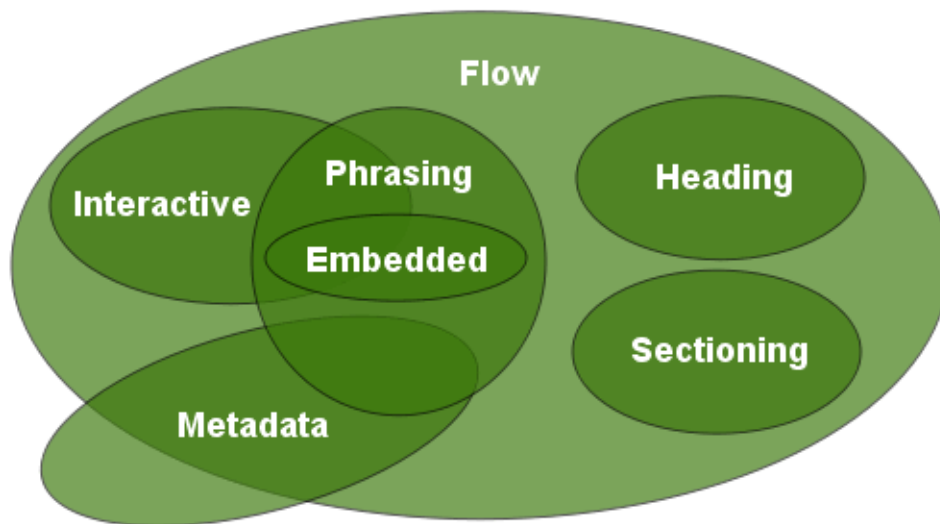
Cf cet article qui compare les différents navigateurs :

[http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(HTML_5\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML_5))

1.1.6 - Le nouveau modèle du W3C

Le modèle HTML 4 était principalement structuré autour de deux types d'éléments : block et inline.

Le nouveau modèle est le suivant :



Presque tous les éléments appartiennent à l'ensemble flow.
Un élément peut appartenir à plusieurs sous-ensembles.

Type	Description
Flow	Modèle de flux : tout texte.
Heading	Les en-têtes de page ou de sections.
Sectioning	L'existence de sections.
Metadata	Tout ce qui n'est pas visible directement : styles, scripts, etc.
Embedded	Incorporation d'objets externes (audio, video, object, ...).
Phrasing	Éléments apparaissant dans le texte : a, li, ...
Interactive	L'interactivité : les événements principalement.

1.1.7 - La structure standard d'un document HTML

Un document HTML est typé.

Un document HTML est composé de 2 parties principales : les en-têtes http (head) et le corps (body).

Le corps est souvent composé au minimum de 6 parties : un en-tête de page (**header**), un menu (**nav**), une partie centrale (une **section** (ancienne div !) avec un **article** et éventuellement un **aside**) et d'un pied de page (**footer**).

```
<!DOCTYPE HTML>

<html>
  <head>
  </head>

  <body>
    <header>
    </header>

    <nav>
    </nav>

    <section>
      <article>
      </article>
      <aside>
      </aside>
    </section>

    <footer>
    </footer>
  </body>
</html>
```

1.1.8 - Le DOCTYPE

Le nouveau DOCTYPE :

```
<!DOCTYPE HTML>
```

Il n'est pas fermé.
Il n'a ni DTD ni schema.
Il est insensible à la casse.

1.1.9 - L'élément HTML

```
<html [lang="langue"] [manifest="fichier.manifest"] >  
  
</html>
```

L'attribut **lang** est utilisé ainsi :

- ✓ Assiste les moteurs de recherche,
- ✓ Assiste les synthétiseurs vocaux,
- ✓ Assiste analyseurs grammaticaux,
- ✓ Aide le "user agent" à sélectionner les glyphes (représentation graphique d'un caractère) d'une typographie,
- ✓ Aide le "user agent" à sélectionner les guillemets,
- ✓ Aide le "user agent" à choisir les espaces, les ligatures (œ, æ, ...) et les coupures de mots.

L'attribut **manifest** sera vu dans le paragraphe sur les applications offline.

1.1.10 - Le head

```
<head>

</head>
```

L'élément `<title>` est hautement recommandé.
Les balises meta, link et script ont été simplifiées.

```
<head>

  <title>Titre de page</title>

  <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />

  <link href="design.css" rel="stylesheet" />

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

</head>
```

De nouvelles valeurs pour l'attribut rel de la balise link ont été créées.

La meta suivante permet d'adapter les dimensions du contenu au contenant :
idéal pour les smartphones ...

```
<meta name="viewport" content="width=device-width"/>
```

En ajoutant quelques attributs on interdit le zoom.

```
<meta name="viewport" content="user-scalable=no, width=device-width,
initial-scale=1.0, maximum-scale=1.0" />
```

1.2 - FAVICON

Le mot favicon est un mot-valise né de la contraction des mots anglais favorite (favori) et icon (icône). C'est l'icône que l'on trouve en haut à gauche de la barre d'url.



Une image de 16x16 ou 32x32 ou 48x48.

Dans le <head>

```
<link rel="shortcut icon" type="image/x-icon" href="favicon.ico" />
<!--[if IE]><link rel="shortcut icon" type="image/x-icon"
href="favicon.ico" /><![endif]-->
```

Les images de type png, jpg, ico et gif sont acceptées par les navigateurs récents.

```
<link rel="icon" type="image/png" href="favicon.png">
```

```
<link rel="icon" type="image/jpeg" href="favicon.jpg">
```

```
<link rel="icon" type="image/gif" href="favicon.gif">
```


1.3 - LES NOUVEAUX ÉLÉMENTS DE MACRO-STRUCTURES

Les macros-structures sont proches des éléments structurant les blogues ou les wikis.

Les principaux macro-éléments sont :

- ✓ Head,
- ✓ Header,
- ✓ Nav,
- ✓ Section,
- ✓ Article,
- ✓ Aside,
- ✓ Footer,
- ✓ Div.

Cf pour la liste complète :

https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5/Liste_des_%C3%A9l%C3%A9ments_HTML5

1.3.1 - Header

L'élément `<header>` représente l'en-tête, la manchette, d'une page ou d'une section.

Il contiendra souvent un logo, un résumé.

En tant qu'en-tête de section il contiendra une introduction et/ou une aide à la navigation (Table des matières, formulaire de recherche, ...).

1.3.2 - Footer

L'élément `<footer>` représente le pied d'une page ou d'une section.

Souvent les informations légales sont affichées dans un footer.

1.3.3 - Nav

L'élément `<nav>` représente une section de liens de navigation.

Les liens seront le plus souvent dans des balises `ul`, `li`.

1.3.4 - Section

L'élément `<section>` permet de définir les différentes sections d'un document comme par exemple les chapitres ou autres sections.

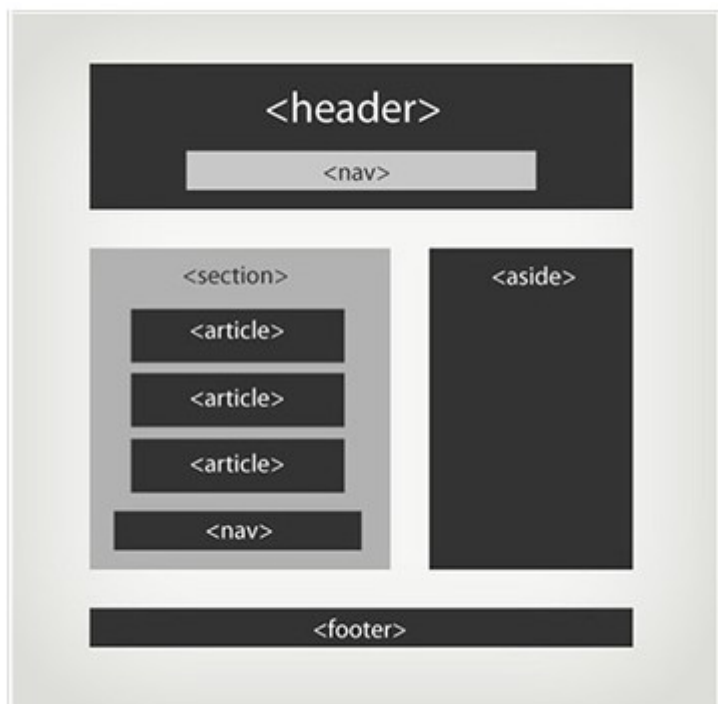
1.3.5 - Article

L'élément `<article>` contient un contenu autonome.

1.3.6 - Aside

Cet élément sert à afficher du contenu additionnel en rapport avec un article, une section ou une page. Il correspond à des notes, des citations, des références mises en exergue dans un encadré.

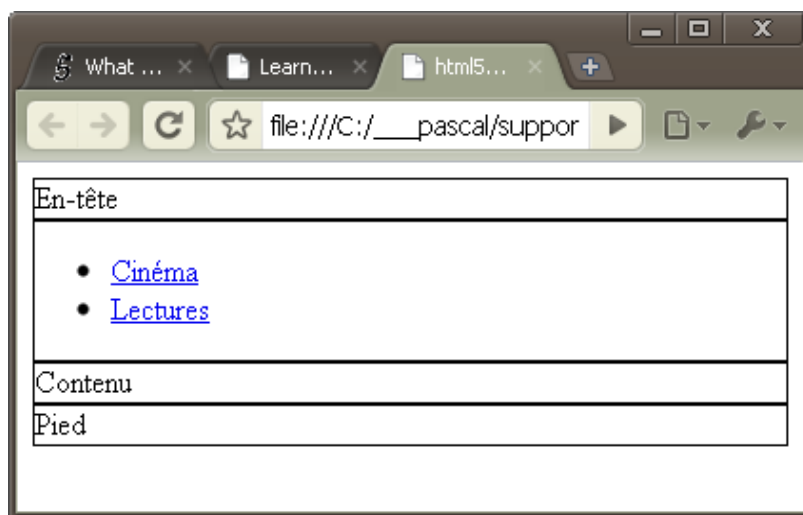
Structure basique !



<http://www.alsacreations.com/article/lire/1376-html5-section-article-nav-header-footer-aside.html>

1.3.7 - Exemple de structuration basique de page

header	
nav	<code><article id="contenu"></code>
	<code></article></code>
footer	



```
<!DOCTYPE HTML>

<html>

  <head>
    <title>Titre</title>
  </head>

  <body>

    <header>En-tête</header>

    <nav>
      <ul>
        <li><a href="#">Cinéma</a></li>
        <li><a href="#">Lectures</a></li>
      </ul>
    </nav>

    <article id="contenu">
      Contenu
    </article>

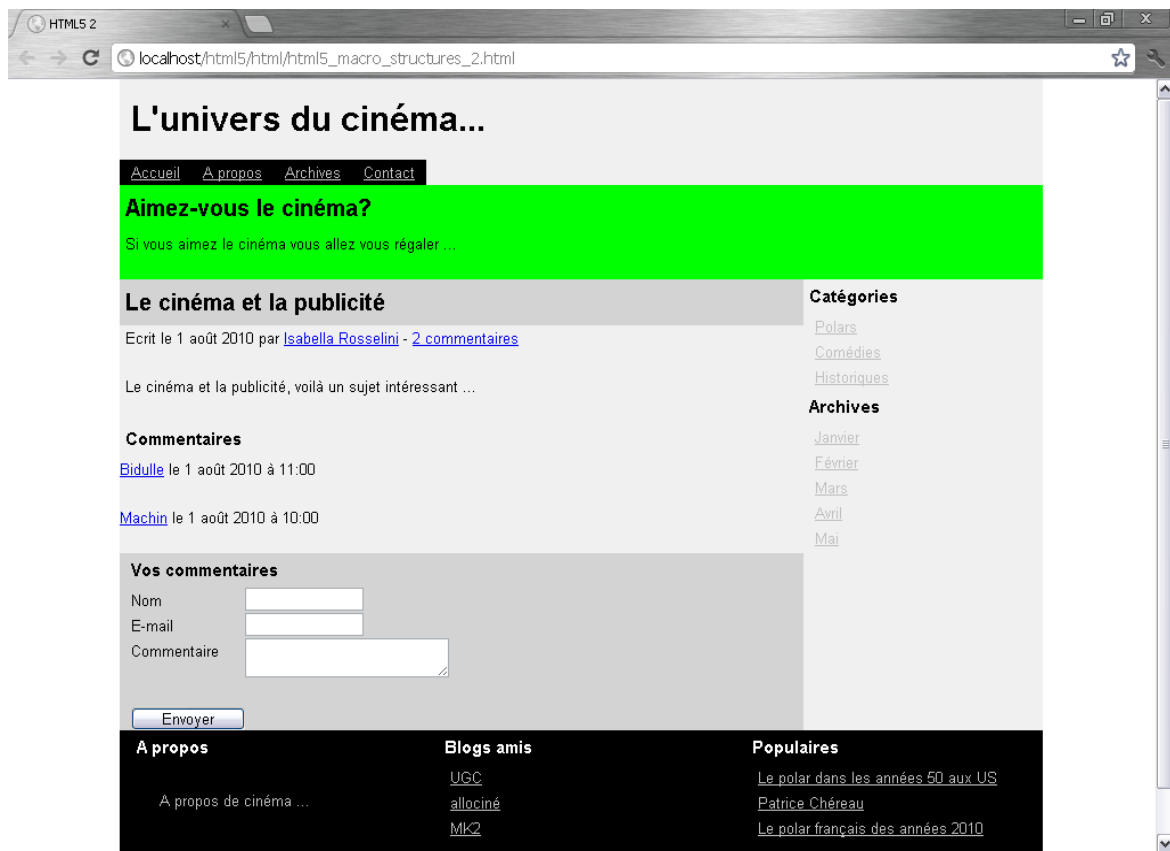
    <footer>
      Pied
    </footer>

  </body>

</html>
```

Exercice

Reproduisez cette page.



1.4 - LES ÉLÉMENTS POUR LES MEDIAS

Tous ces éléments sont des éléments embedded (incorporés).

1.4.1 - Image

La balise `` ne change pas.

Note : vous pouvez omettre l'auto-fermeture !

1.4.2 - Audio

Nouvelle balise pour le son : la balise <audio>.

Syntaxe

```
<audio>
  <source src="source" type="type mime" />
  Texte.
</audio>
```

Types possibles : audio/ogg, audio/mpeg, audio/wav.

Attributs

Attribut	Valeur	Description
autoplay	autoplay	Démarrage automatique
controls	controls	Affichage des boutons
loop	loop	Joue en boucle
preload	preload	Chargement au chargement de la page
src	url	URL

Exemple

```
<audio controls="controls">
  <source src="../../sons/candide_01_voltaire.ogg"
type="audio/ogg" />
  <!-- <source src="../../sons/ART_MENGO___LES_PARFUMS_DE_.MP3"
type="audio/mpeg" /> -->
  <!-- <source src="../../sons/wmpaud7.wav" type="audio/wav" /> --
>
  Votre navigateur ne supporte pas la balise audio.
</audio>
```

1.4.3 - Video

Nouvelle balise pour la video : la balise <video>.

Syntaxe

```
<video>
  <source src="source" type="type mime" />
  Texte
</video>
```

Types possibles : video/ogg, video/mp4.

Attributs

Attribut	Valeur	Description
autoplay	autoplay	Démarrage automatique
controls	controls	Affichage des boutons
width	pixels	Largeur
height	pixels	Hauteur
loop	loop	Joue en boucle
preload	preload	Chargement au chargement de la page
src	url	URL

Exemple

```
<video width="320" height="240" controls="controls"
autoplay="autoplay">
  <!-- Ne supporte pas les noms de fichier avec des espaces !!! --
>
  <source src="../../../videos/nottinghill.mp4" type="video/mp4" />
  <source src="../../../videos/Lexorcisme.ogg" type="video/ogg" />
  Votre navigateur ne supporte pas la balise video!!!
</video>
```

1.5 - LES NOUVEAUX TYPES INPUT

Les nouveaux types input sont les suivants :

- ✓ email
- ✓ url
- ✓ number
- ✓ range
- ✓ date pickers (date, month, week, time, datetime, datetime-local),
- ✓ search,
- ✓ color.

1.5.1 - email

Permet de contrôler la bonne formation d'un e-mail. Les contrôles sont effectués au moment de la soumission de la requête (clic sur le bouton submit du formulaire).

```
<input type="email" name="nom" value="valeur"/>
```

```
<input type="email" name="e_mail" value="tintin@free.fr"/>
```

Note : avec Opera cette adresse est correcte : tintin@free !!!

1.5.2 - url

Permet de contrôler la bonne formation d'une URL.

```
<input type="url" name="nom" value="url"/>
```

```
<input type="url" name="url" value="http://tintin.free.fr"/>
```

Note : avec Opera il faut que l'url commence par http:// ...

1.5.3 - number

Présente un spinner (Bouton fléché, compteur) pour saisir un numérique. Permet de préciser une plage de valeurs.

```
<input type="number" name="nom" [id="nom"] [value="valeur"]  
[min="min"] [max="max"] />
```

```
<input type="number" name="salaire" id="salaire" value="3000"  
min="1500" max="5000" />
```

1.5.4 - range

Présente un slider (curseur) pour saisir un numérique avec une borne min et une borne max.

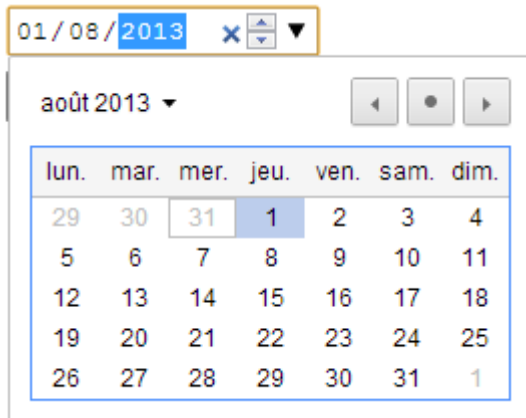


```
<input type="range" [name="nom" id="id" min="min" max="max"
value="valeur"] />
```

```
<input type="range" name="age" id="age" min="7" max="77" value="7"/
>
```

1.5.5 - date pickers

Présente un calendrier (cf plus de détails plus loin).
Si vous voulez initialiser la date elle doit être au format yyyy-mm-dd.



```
<input type="date" name="nom" id="id" value="aaaa-mm-jj"/>
```

```
<input type="date" name="dateNaissance" id="dateNaissance"
value="1936-12-30"/>
```

Note du 1er août 2013 :

Opera 12.16 : calendrier en français mais la date est au format yyyy/mm/dd

IE 10.0.7 : toujours pas de calendrier !

Chrome 28.0.1500.72 : calendrier en français et la date au format jj/mm/aaaa

Firefox 22.0 : toujours pas de calendrier !

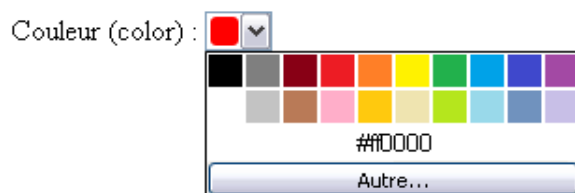
1.5.6 - color

Permet de récupérer une couleur en hexa.

```
<input type="color" name="nom" id="id" value="couleur en hexa précédé de #" />
```

```
<input type="color" name="couleur" id="couleur" value="#FF0000" />
```


Avec Opera :



Avec Chrome



Récupération de la valeur d'une couleur :

Couleur (color) : 

#00ff00

```
// -----  
function getCouleur() {  
    document.getElementById("infos").innerHTML =  
document.getElementById("couleur").value;  
  
    document.getElementById("infos").style.color =  
document.getElementById("couleur").value;  
}  
  
// --- Gestion events  
document.getElementById("couleur").onchange = getCouleur;
```


1.5.7 - tel

Permet de créer un élément de type téléphone.

```
<input type="tel" name="telephone" />
```

L'objectif est de proposer automatiquement un clavier numérique sur les smartphones.

1.5.8 - search

Permet de créer un élément de recherche.

Dès que vous saisissez un caractère une croix s'affiche avec Chrome. Si vous cliquez sur la croix le texte est effacé, et le focus reste dans le champ.

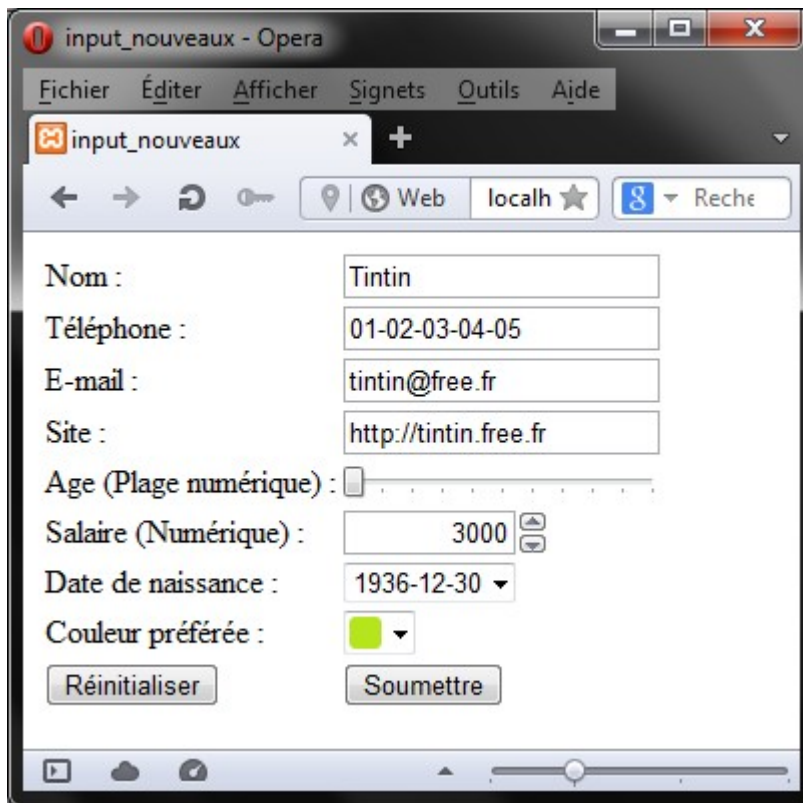
Recherche (search) :



```
<input type="search" name="nom" id="id" />
```

```
<input type="search" name="recherche" id="idRecherche" />
```

1.5.9 - Exemple mixte



The screenshot shows a web browser window titled "input_nouveaux - Opera". The browser's address bar displays "localh" and a search bar contains "Reche". The form itself is titled "input_nouveaux" and contains the following fields and controls:

- Nom :
- Téléphone :
- E-mail :
- Site :
- Age (Plage numérique) :
- Salaire (Numérique) : with up/down arrow buttons
- Date de naissance : with a dropdown arrow
- Couleur préférée : with a dropdown arrow
- Buttons: "Réinitialiser" and "Soumettre"

input_nouveaux.html

```

<!DOCTYPE HTML>
<html lang="fr">

  <head>
    <title>input_nouveaux.html</title>
    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <script src="../../js/input_nouveaux.js"></script>
  </head>

  <body>
    <section id="centre">
      <form>
        <table>
          <tr>
            <td>Nom : </td>
            <td><input type="text" name="nom"
value="Tintin" /></td>
          </tr>
          <tr>
            <td>Téléphone : </td>
            <td><input type="tel" name="tel" value="01-
02-03-04-05" /></td>
          </tr>
          <tr>
            <td>E-mail (email) : </td>
            <td><input type="email" name="eMail"
value="tintin@free.fr" /></td>
          </tr>
          <tr>
            <td>Site (url) : </td>
            <td><input type="url" name="url"
value="http://tintin.free.fr" /></td>
          </tr>
          <tr>
            <td>Age (range : plage numérique) : </td>
            <td>
              <input type="range" name="age" id="age"
min="7" max="77" value="7" />
              <span id="spAge"></span>
            </td>
          </tr>
          <tr>
            <td>Salaire (number : numérique) : </td>
            <td>
              <input type="number" name="salaire"
id="salaire" value="3000" />
              <span id="spSalaire"></span>
            </td>
          </tr>
          <tr>

```

```
        <td>Date (date) : </td>
        <td>
            <input type="date" name="dateNaissance"
id="dateNaissance" value="1936-12-30" />
            <span id="spDateNaissance"></span>
        </td>
    </tr>
    <tr>
        <td>Couleur préférée : </td><td>
            <input type="color" name="couleur"
value="#0000FF"/></td>
        </tr>
    <tr>
        <td><input type="reset"/></td>
        <td><input type="submit"/></td>
    </tr>
</table>
</form>
</section>
</body>
</html>
```

input_nouveaux.js

```
// -----
function getAge() {
    document.getElementById("spAge").innerHTML =
document.getElementById("age").value;
}

// -----
function getSalaire() {
    document.getElementById("spSalaire").innerHTML =
document.getElementById("salaire").value;
}

// -----
function getDateNaissance() {
    document.getElementById("spDateNaissance").innerHTML =
document.getElementById("dateNaissance").value;
}

// -----
function getAujourd'huiYYYYMMDD() {
    var d = new Date();

    var annee = d.getFullYear();
    var mois = d.getMonth() + 1;
    var jour = d.getDate();

    if (mois < 10) {
        mois = "0" + mois;
    }
    if (jour < 10) {
        jour = "0" + jour;
    }

    var aujourd'hui = annee + "-" + mois + "-" + jour;

    return aujourd'hui;
} /// getAujourd'huiYYYYMMDD

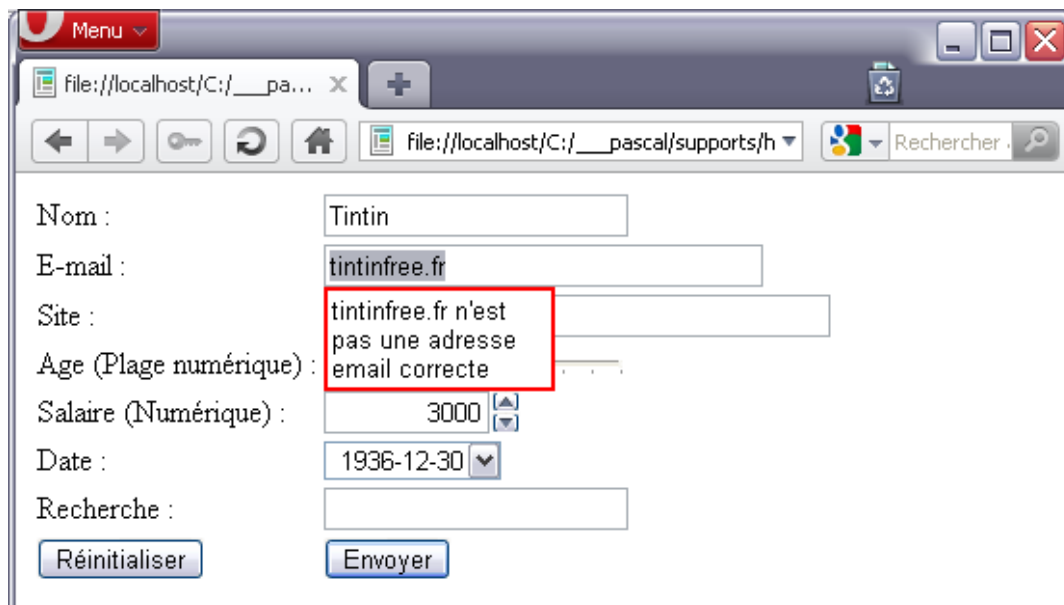
// -----
function init() {
// --- Gestion events
    document.getElementById("dateNaissance").value =
getAujourd'huiYYYYMMDD();
//alert(aujourd'hui);

    document.getElementById("age").onchange = getAge;
    document.getElementById("salaire").onchange = getSalaire;
    document.getElementById("dateNaissance").onchange =
getDateNaissance;
}
```

```
// -----  
window.onload = init;
```

1.5.10 - Quelques écrans d'erreurs

email



A screenshot of a web browser window displaying a form. The browser's address bar shows a file path. The form contains several input fields: 'Nom' (Tintin), 'E-mail' (tintinfree.fr), 'Site' (empty), 'Age (Plage numérique)' (empty), 'Salaire (Numérique)' (3000), 'Date' (1936-12-30), and 'Recherche' (empty). A red box highlights the 'E-mail' field, which contains the text 'tintinfree.fr n'est pas une adresse email correcte'. Below the form are two buttons: 'Réinitialiser' and 'Envoyer'.

Menu

file:///localhost/C:/__pa... x

file:///localhost/C:/__pascal/supports/h

Rechercher

Nom : Tintin

E-mail : tintinfree.fr

Site : tintinfree.fr n'est pas une adresse email correcte

Age (Plage numérique) :

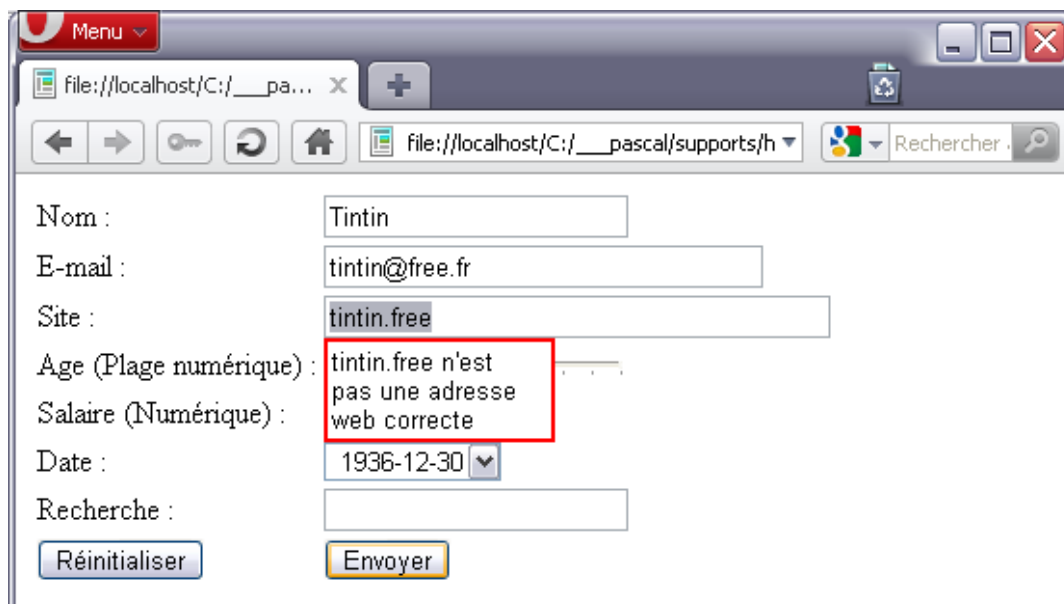
Salaire (Numérique) : 3000

Date : 1936-12-30

Recherche :

Réinitialiser Envoyer

url



A screenshot of a web browser window displaying the same form as above. The 'E-mail' field now contains 'tintin@free.fr'. The 'Site' field contains 'tintin.free'. A red box highlights the 'Site' field, which contains the text 'tintin.free n'est pas une adresse web correcte'. The 'Envoyer' button is highlighted in yellow.

Menu

file:///localhost/C:/__pa... x

file:///localhost/C:/__pascal/supports/h

Rechercher

Nom : Tintin

E-mail : tintin@free.fr

Site : tintin.free

Age (Plage numérique) : tintin.free n'est pas une adresse web correcte

Salaire (Numérique) :

Date : 1936-12-30

Recherche :

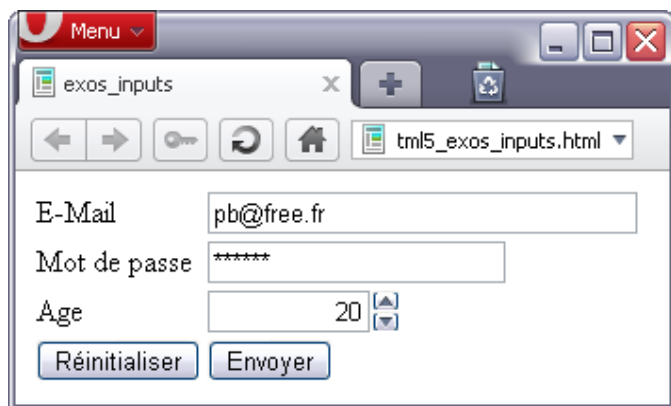
Réinitialiser Envoyer

Exercice

Créez le formulaire suivant avec :

L'e-mail qui doit être valide.

L'âge qui est entre 18 et 25 ans.



The screenshot shows a web browser window with a single tab titled 'exos_inputs'. The address bar displays 'tml5_exos_inputs.html'. The form contains three input fields: 'E-Mail' with the value 'pb@free.fr', 'Mot de passe' with masked characters '*****', and 'Age' with the value '20'. Below the inputs are two buttons: 'Réinitialiser' and 'Envoyer'.

E-Mail	<input type="text" value="pb@free.fr"/>
Mot de passe	<input type="password" value="*****"/>
Age	<input type="text" value="20"/>
<input type="button" value="Réinitialiser"/> <input type="button" value="Envoyer"/>	

1.6 - LES INPUTS DATE

1.6.1 - Les sous-types

```
<input type="date" />
```

```
<input type="month" />
```

```
<input type="week" />
```

```
<input type="time" />
```

```
<input type="datetime" />
```

```
<input type="datetime-local" />
```

Remarques

Initialisation de date : aaaa-mm-dd

Initialisation de month : aaaa-mm

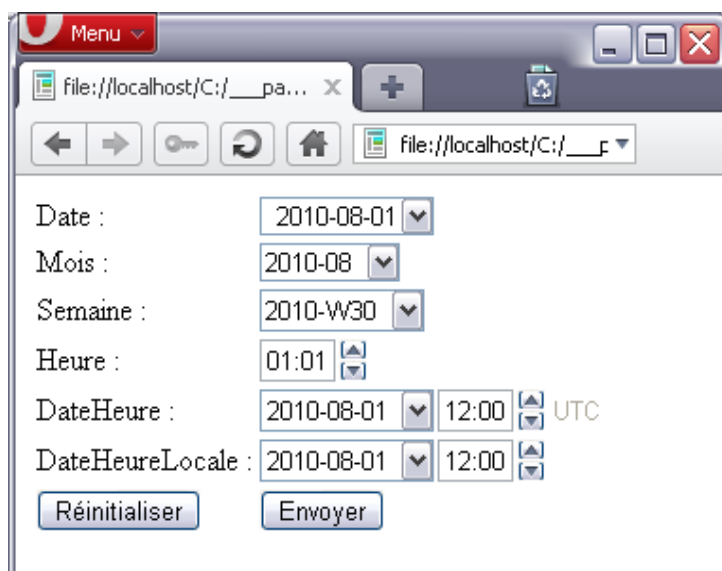
Initialisation de week : aaaa-Wnn

Initialisation de time : hh:mm ou hh:mm:ss

Initialisation de datetime : aaaa-mm-ddThh:mm:ssZ

Initialisation de datetime-local : même chose sans le Z

1.6.2 - Exemple



The screenshot shows a web browser window with a single tab titled "file:///localhost/C:/__pa...". The address bar contains "file:///localhost/C:/__f". The page content is a form with the following fields and controls:

- Date : 2010-08-01 (dropdown)
- Mois : 2010-08 (dropdown)
- Semaine : 2010-W30 (dropdown)
- Heure : 01:01 (spinners)
- DateHeure : 2010-08-01 (dropdown) 12:00 (spinners) UTC
- DateHeureLocale : 2010-08-01 (dropdown) 12:00 (spinners)
- Buttons: Réinitialiser, Envoyer

type="date"

2010-08-01 ▾

◀ Aout ▶ 2010 ⌵

Lun	Mar	Mer	Jeu	Ven	Sam	Dim
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Aujourd'hui

type="month"

2010-08 ▾

◀ Aout ▶ 2010 ⌵

Lun	Mar	Mer	Jeu	Ven	Sam	Dim
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Aujourd'hui

type="week"

2010-W30 ▾

◀ juillet ▶ 2010 ⌵

Semaine	Lun	Mar	Mer	Jeu	Ven	Sam	Dim
26	28	29	30	1	2	3	4
27	5	6	7	8	9	10	11
28	12	13	14	15	16	17	18
29	19	20	21	22	23	24	25
30	26	27	28	29	30	31	1
31	2	3	4	5	6	7	8

Aujourd'hui

```

<!DOCTYPE HTML>
<html>
<!-- html5_input_date.html -->
<head>
    <title>html5_input_date.html</title>
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
</head>

<body>

<form>
<table>
<tr>
<td>Date : </td>
<td><input type="date" name="date_naissance" value="2010-08-01"
/></td>
</tr>
<tr>
<td>Mois : </td>
<td><input type="month" name="mois" id="mois" value="2010-08"
/></td>
</tr>
<tr>
<td>Semaine : </td>
<td><input type="week" name="semaine" id="semaine" value="2010-W30"
/></td>
</tr>
<tr>
<td>Heure : </td>
<td><input type="time" name="heure" id="heure" value="08:00"
/></td>
</tr>
<tr>
<td>DateHeure : </td>
<td><input type="datetime" name="date_heure" id="date_heure"
value="2010-08-01T08:00:00Z" /></td>
</tr>
<tr>
<td>DateHeureLocale : </td>
<td><input type="datetime-local" name="date_heure_locale"
id="date_heure_locale" value="1996-12-19T16:39:57" /></td>
</tr>

<tr>
<td><input type="reset" /></td>
<td><input type="submit" /></td>
</tr>
</table>
</form>

</body>
</html>

```


L'URL

file:///localhost/C:/___pascal/supports/html_supports/site_html5/
html5_input_date.html?date_naissance=2010-08-01&mois=2010-
08&semaine=2010-W30&heure=01%3A01%3A00&date_heure=2010-08-
01T12%3A00Z&date_heure_locale=2010-08-01T12%3A00

1.6.3 - Initialisation d'un champ date à la date du jour

Avec Chromium :



et ça sur Nexus-Android :



Le champ de saisie :

```
<input type="date" name="itDate" id="itDate" value="" />
```

Le script :

```
<script>
    var d = new Date();

    var jour = new String(d.getDate()); // quantieme

    if (jour.length === 1) {
        jour = "0" + jour;
    }

    var mois = new String(d.getMonth() + 1); // commence a 0

    if (mois.length === 1) {
        mois = "0" + mois;
    }

    var annee = d.getFullYear();

    var aujourd'hui = annee + "-" + mois + "-" + jour;

    $("#itDate").val(aujourd'hui);
</script>
```

Notez qu'il faut passer par la classe String pour calculer la longueur et que le format d'initialisation est YYYY-MM-DD même si ensuite l'affichage est dans la locale.

1.6.4 - Contrôle de dates et des heures

Objectif

Contrôler plus finement la saisie d'une date ou d'une heure.
En ajoutant les attributs min, max et step.

Date contrôlée :

Heure contrôlée :

Semaine	Lun	Mar	Mer	Jeu	Ven	Sam	Dim
30	26	27	28	29	30	31	1
31	2	3	4	5	6	7	8
32	9	10	11	12	13	14	15
33	16	17	18	19	20	21	22
34	23	24	25	26	27	28	29
35	30	31	1	2	3	4	5

Heure contrôlée :

Vous remarquerez que les boutons pour les mois sont grisés ainsi que tous les jours du mois sauf les dimanches.

Exemple

Les livraisons sont effectuées tous les dimanches du mois d'août 2010 de 11 heures à 23 heures de ¼ d'heure en ¼ d'heure (900 secondes).

```

<tr>
  <td>Date contrôlée : </td>
  <td><input type="date" name="date_livraison" value="2010-08-01"
min="2010-08-01" max="2010-08-30" step="7" /></td>
</tr>

<tr>
  <td>Heure contrôlée : </td>
  <td><input type="time" name="heure_livraison" value="11:00"
min="11:00" max="23:00" step="900" /></td>
</tr>

```

Pour l'heure :



De 09:00 à 19:00 par suat de 30 minutes (1800 secondes).

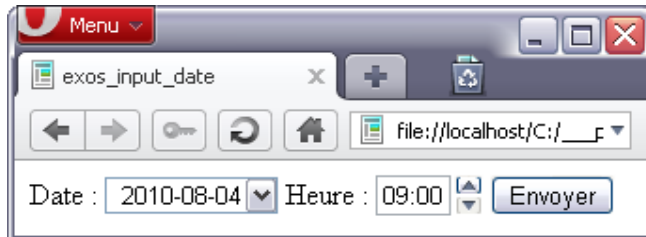
```
<input type="time" name="itHeure" id="itHeure" value="09:00"  
min="09:00" max="19:00" step="1800"/>
```

Avec le Nexus-Android ça ne fonctionne pas !!! Enfin le pas ne fonctionne pas. Le min et max fonctionnent.



Exercice

Créez un formulaire pour la saisie de la date et de l'heure de rendez-vous du 4 janvier 2010 au 31 décembre 2010 de 9 heures à 13 heures toutes les demi-heure.

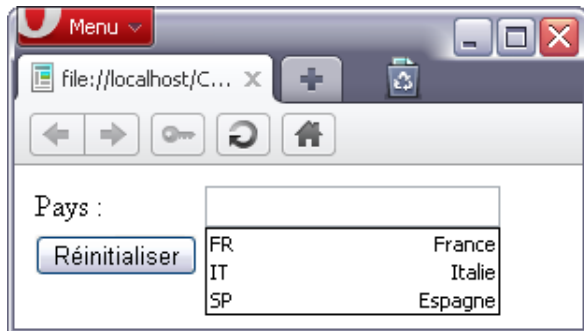


The screenshot shows a web browser window with a single tab titled 'exos_input_date'. The address bar displays 'file:///localhost/C:/...'. The page content includes a 'Menu' dropdown menu, a toolbar with navigation icons (back, forward, search, refresh, home), and a form with two dropdown menus: 'Date' (set to '2010-08-04') and 'Heure' (set to '09:00'). An 'Envoyer' button is located to the right of the 'Heure' dropdown.

1.7 - LES NOUVEAUX ÉLÉMENTS DE FORMULAIRE

1.7.1 - datalist

Permet de présenter une liste lorsque l'élément de saisie prend le focus.
L'élément en question comportera l'attribut list.



Syntaxes

```
<input type="text" name="nom" list="idListe" />
```

```
<datalist id="idList">  
<option label="Valeur affichée" value="Valeur récupérée" />  
<option label="Valeur affichée" value="Valeur récupérée" />  
...  
</datalist>
```

Exemple

```
<!DOCTYPE HTML>
<html>
<!-- html5_datalist.html -->
<head>
  <title></title>
  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />
</head>

<body>

<form>

<table>

<tr>
<td>Pays : </td>
<td><input type="text" name="pays" list="liste_pays" value=""
/></td>
</tr>

<datalist id="liste_pays">
<option label="France" value="FR" />
<option label="Italie" value="IT" />
<option label="Espagne" value="SP" />
</datalist>

<tr>
<td><input type="reset" /></td>
<td><input type="submit" /></td>
</tr>

</table>

</form>

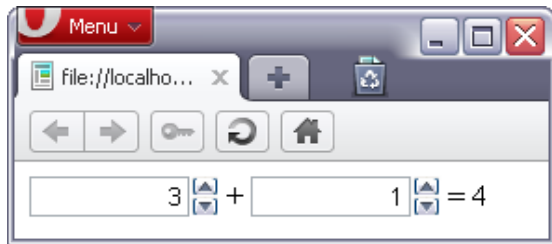
</body>

</html>
```

1.7.2 - output

Objectif

L'élément output représente le résultat d'un calcul effectué dans un formulaire web sans que le formulaire soit soumis.



Syntaxes

Le calcul est effectué sur l'événement onforminput.
Dès qu'une saisie est réalisée dans le formulaire l'événement est déclenché.

```
<form oninput="nom_de_l_output.value=résultat du calcul">
```

```
<output name="nom_de_l_output"></output>
```

Exemple

Calculer et afficher au fur et à mesure de la "frappe" le résultat d'une addition de deux éléments de type number.

```
<!DOCTYPE HTML>

<html>
<!-- html5_output.html -->

<head>
  <title></title>
  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />
</head>

<body>
  <form oninput="result.value=parseInt(a.value) +
parseInt(b.value)">

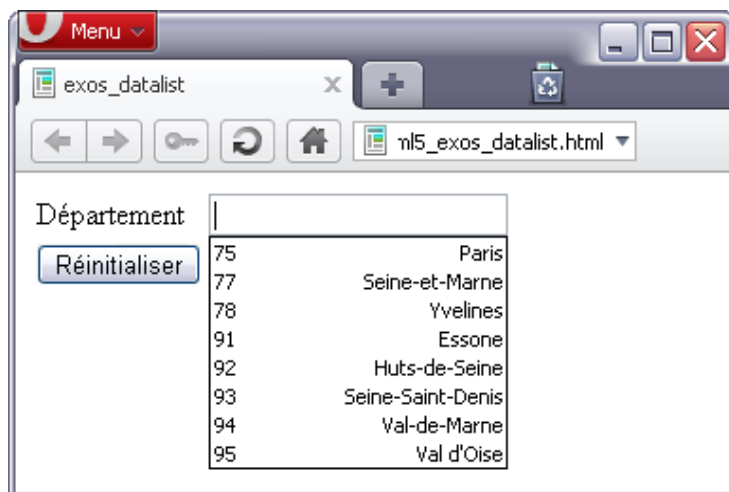
    <input name="a" type="number" step="any" value="0" /> +
    <input name="b" type="number" step="any" value="0" /> =

    <output name="result"></output>

  </form>
</body>
</html>
```

Exercice sur la datalist

Créez un formulaire avec une liste de suggestions pour la saisie des codes des départements de l'Ile-de-France.



The screenshot shows a web browser window with the title 'exos_datalist'. The address bar displays 'nl5_exos_datalist.html'. The page content includes a label 'Département' next to a text input field. Below the input field is a button labeled 'Réinitialiser'. A dropdown menu is open, showing a list of department codes and names for the Ile-de-France region:

75	Paris
77	Seine-et-Marne
78	Yvelines
91	Essone
92	Hauts-de-Seine
93	Seine-Saint-Denis
94	Val-de-Marne
95	Val d'Oise

1.7.3 - mark

Cette balise permet de mettre un texte en exergue. Un peu comme le faisait l'élément ``. La plupart des navigateurs surlignent.

Mark : HTLM5

```
<mark>Texte</mark>
```

```
<span>Mark : </span>  
<mark>HTLM5</mark>
```

1.7.4 - meter

La balise `<meter>` permet d'afficher une jauge représentant un état. Les bornes sont au choix du développeur. Elles peuvent être négatives.

Visuel :

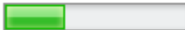



Syntaxe


```
<meter min="min" max="max" value="valeur" [low="bas"
optimum="optimum" high="haut"]>valeur</meter>
```

Les attributs `low`, `optimum` et `high` ont pour effet de modifier la couleur en fonction de la valeur et de son rapport avec les attributs en question.

Exemples

Pourcentage : 33% (Entre 0 et 100) 

Note : 12 (Entre 0 et 20) 

Température : 17 (Entre -50 et 50) 

```
<p>
  <label>Pourcentage : 33% (Entre 0 et 100)</label>
  <meter min="0" max="100" value="33">33</meter>
  <br/><br/>

  <label>Note : 12 (Entre 0 et 20)</label>
  <meter min="0" max="20" value="12">12/10</meter>
  <br/><br/>

  <label>Température : 17 (Entre -50 et 50)</label>
  <meter min="-50" max="50" value="17">17</meter>
  <br/><br/>
</p>
```

Exemple avec JS

On saisit une note, on valide, ça "jauge"!!!

Saisir la note : 

Saisir la note : 

```
<p>
  <label>Saisir la note : </label>
  <input type="text" name="itNote" id="itNote" />
  <input type="button" id="btNote" value="Valider" />
  <meter min="0" max="20" value="0" id="meterNote"></meter>
  <br/>
</p>
```

```
// -----
function getMeter() {
    document.getElementById("meterNote").value =
document.getElementById("itNote").value;
}

// --- Gestion events
document.getElementById("btNote").onclick = getMeter;
```

Exercice

Saisir la note avec un `<input type="number" />`.

Saisir la note : 

1.7.5 - progress

La balise `<progress>` permet d'afficher une jauge de progression.
Les valeurs sont comprises entre 0 et 100 (%).

Jauge Progressive : 

Syntaxe

```
<progress id="id" max="max" value="valeur"></progress>
```

Si max n'est pas défini max est égal à 1.
La valeur peut être décimale.

Exemple

```
<progress id="jaugeProgressive" max="100" value="50"></progress>
```

A utiliser de façon statique cela ne présente aucun intérêt.

Exercice

Sablier sur 1 minute ...

Jauge Progressive : 

Jauge Progressive : 

1.7.6 - time

<http://www.alsacreations.com/article/lire/1386-html5-element-time.html>

<http://html5doctor.com/the-time-element/>

1.7.7 - keygen

`<keygen>` permet de générer un jeu de clefs pour le cryptage et le décryptage d'informations.

Le jeu équivaut à une paire de clés, l'une dite publique, l'autre dite privée.

La clé privée est stockée localement, tandis que la clé publique est envoyée sur le serveur.

Les attributs sont :

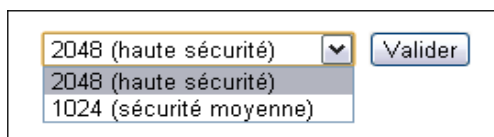
Keytype : le type d'encryptage (Seulement RSA actuellement).

Challenge : une phrase envoyée avec la clé publique.

```
<!DOCTYPE HTML>
<html>
<head>
  <title>html5_keygen</title>
  <meta charset="utf-8" />
</head>

<body>
  <form>
    <keygen name="cle" challenge="challenge" keytype="rsa" />
    <input type="submit"/>
  </form>
</body>
</html>
```

Ecran généré avec Chrome :



A screenshot of a web form. It features a dropdown menu with three options: '2048 (haute sécurité)', '2048 (haute sécurité)', and '1024 (sécurité moyenne)'. The first two options are identical. To the right of the dropdown is a button labeled 'Valider'.

Clé générée :

http://localhost/html5/html/html5_keygen.html?cle=MIICSTCCATEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCgBQtB6woOyrGRTL45mnjfa%2BR%2BBnrZI48r9vEfmA9YftgzsFDpiZG%2FOfSSiHHHZaIW%2FVyDMkQF4uIh4G%2BN1p2eAzuoed7RjheQITbbFfHhErIkNITUgdNYbjJykyw2kpvJ7eL%2Bg9cYgftiro53zwp14MOIu8%2FEKkEbT1cnyISRQFagimoEufg%2BeITxA1zrsMMOC3A5kwPNsiN%2FsAftBUmZ%2BN4gZAQB%2B7J7rw6ZjKU2LmbVALOxwWQUfsVf6ZvJ6BVS52foZBb0OjxAE56Jbxyp8SOPwTfQcu1qnjUafkU0YmhgeeZcj8DOLNF84KyjJ12g5dGvJziPqNqmFUZNKytAgMBAAEWcWNoYWxsZW5nZTANBkgqhkiG9w0BAQQFAAOCAQEAa0So%2FfLUF56FefRjBXKcVG6NIE%2BT92ZZ70DDROL37ClwFi%2BLg4fsvtjbi6sDiei5GauPRaVpRwnRcE7yANWyDBew8yjdPx3GEIxf6D8QgKcfN%2FAudiiSkwRZo04uYvPZA2ScPxHX9g9sv%2BFJmOrarWHECl4AYmFkSBQNzyhHDLxc98Ij88%2BoGF35JXHDe9TT9USK6iTjevoJOzc6miReklQrX3I740KUn3yxeEdqbVg6QTFD%2FX%2BSLCuyackw1Yho0pPxjj3YF%2BpG8NnMzdLAMN4ZW%2F9cdSWnAiuJNpKNABNCEqXGSmrDxz%2B%2B3wnVLheSfThUGVDVNFkbmaZ5mNHpQ%3D%3D

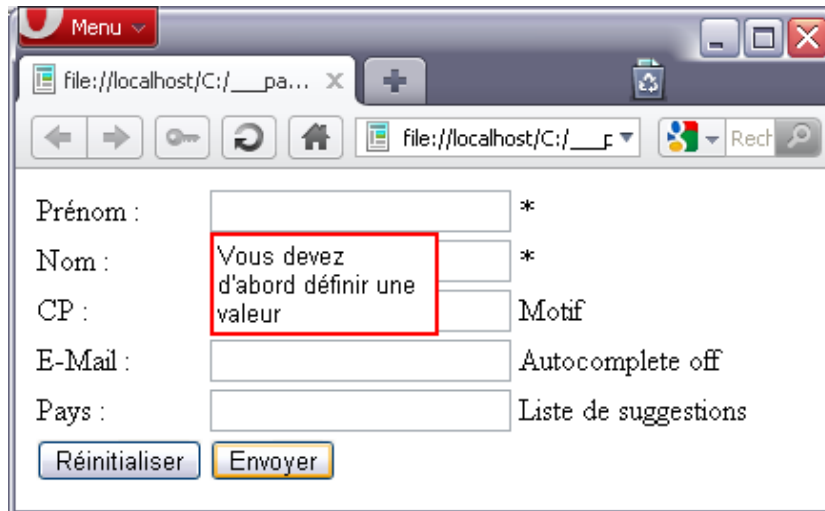
1.8 - LES NOUVEAUX ATTRIBUTS DES ÉLÉMENTS DE FORMULAIRE

1.8.1 - Les attributs

Attribut	Valeur	Description
autocomplete	on off	Autocompletion. Quand autocomplete est à On sur la balise <form> il est possible de l'annuler sur un des <input> du formulaire avec autocomplete à off.
autofocus	autofocus	Prend le focus automatiquement lors du chargement de la page.
form		Référence d'un élément à un formulaire alors que l'élément est en dehors de l'élément <form>.
form overrides		
height and width		Hauteur et largeur de l'image source de l' <input type="image" / >.
list		Liste de suggestions.
min, max and step		Minimum, maximum et pas.
multiple		Sélection multiple.
novalidate	True	Ne valide pas automatiquement.
pattern	Motif	Motif d'une expression régulière.
placeholder	Un texte	Affiche la valeur de cet attribut en gris pâle. La valeur d'initialisation disparaît dès que l'élément reçoit le focus.
required	Required	Obligatoire. Si le champ est vide lors de la validation du formulaire avec Opera affiche un message et avec Chrome met le focus sur le champ sans message.

1.8.2 - Exemple

Prénom : * pour placeholder ...



A screenshot of a web browser window displaying a registration form. The browser's address bar shows a file path. The form contains several input fields with placeholder text. A red box highlights the 'Nom' field, which contains the text 'Vous devez d'abord définir une valeur'. The form includes buttons for 'Réinitialiser' and 'Envoyer'.

Menu ▾

file:///localhost/C:/__pa... x +

file:///localhost/C:/__f ▾ Rect 🔍

Prénom : *

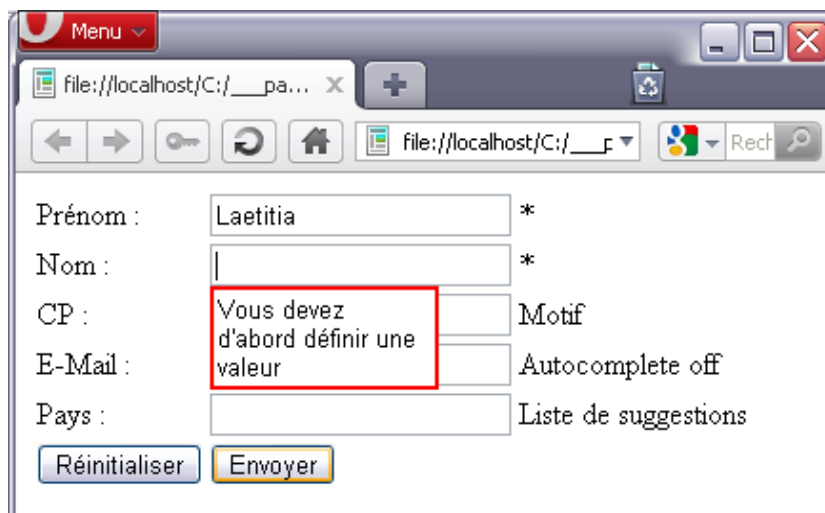
Nom : *

CP : Motif

E-Mail : Autocomplete off

Pays : Liste de suggestions

Réinitialiser Envoyer



A second screenshot of the same web browser window, showing the registration form after the 'Prénom' field has been filled with the name 'Laetitia'. The 'Nom' field still contains the placeholder text. The 'Réinitialiser' and 'Envoyer' buttons remain at the bottom.

Menu ▾

file:///localhost/C:/__pa... x +

file:///localhost/C:/__f ▾ Rect 🔍

Prénom : *

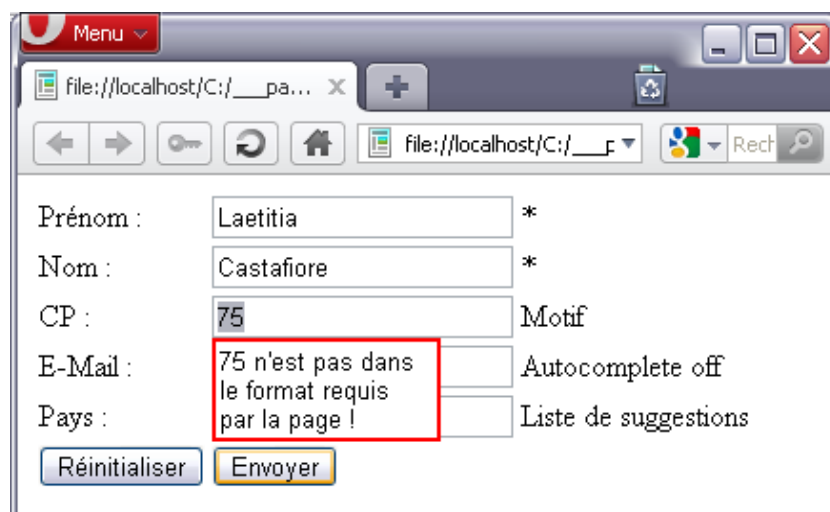
Nom : *

CP : Motif

E-Mail : Autocomplete off

Pays : Liste de suggestions

Réinitialiser Envoyer



A screenshot of a web browser window displaying a form. The browser's address bar shows a file path. The form contains several input fields: 'Prénom' with 'Laetitia', 'Nom' with 'Castafore', 'CP' with '75', 'E-Mail' with '75 n'est pas dans le format requis', and 'Pays' with 'par la page !'. The 'E-Mail' and 'Pays' fields are highlighted with a red border. To the right of the 'CP' field is the label 'Motif'. To the right of the 'E-Mail' field is the text 'Autocomplete off'. To the right of the 'Pays' field is the text 'Liste de suggestions'. At the bottom left are two buttons: 'Réinitialiser' and 'Envoyer'.

Menu

file:///localhost/C:/__pa... x

file:///localhost/C:/__f

Prénom : Laetitia *

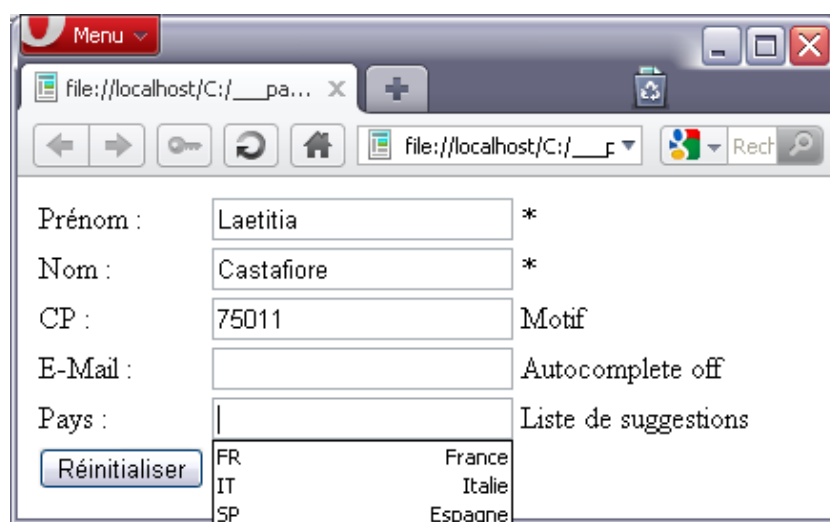
Nom : Castafore *

CP : 75 Motif

E-Mail : 75 n'est pas dans le format requis Autocomplete off

Pays : par la page ! Liste de suggestions

Réinitialiser Envoyer



A screenshot of a web browser window displaying the same form as above. The 'CP' field now contains '75011'. The 'E-Mail' field is empty. The 'Pays' field is empty, and a dropdown menu is visible below it, showing the following options: FR France, IT Italie, and SP Espagne. The 'Autocomplete off' and 'Liste de suggestions' text remains to the right of the 'E-Mail' and 'Pays' fields respectively. The 'Réinitialiser' and 'Envoyer' buttons are still at the bottom left.

Menu

file:///localhost/C:/__pa... x

file:///localhost/C:/__f

Prénom : Laetitia *

Nom : Castafore *

CP : 75011 Motif

E-Mail : Autocomplete off

Pays : Liste de suggestions

Réinitialiser

FR France

IT Italie

SP Espagne

```

<!DOCTYPE HTML>
<html>
<!-- html5_input_nouveaux_attributs.html -->

<head>
    <title></title>
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
</head>

<body>
<form autocomplete="on">

<table>
<tr>
<td>Prénom : </td>
<td><input type="text" name="prenom" placeholder="Saisissez ..."
required="required" /></td>
<td>*</td>
</tr>

<tr>
<td>Nom : </td>
<td><input type="text" name="nom" value="Castafiore"
required="required" autofocus="autofocus" /></td>
<td>*</td>
</tr>

<tr>
<td>CP : </td>
<td><input type="text" name="cp" pattern="[0-9]{5}" value="75011" /
></td>
<td>Motif</td>
</tr>

<tr>
<td>E-Mail : </td>
<td><input type="text" name="e_mail" value="" autocomplete="off"
/></td>
<td>Autocomplete off</td>
</tr>

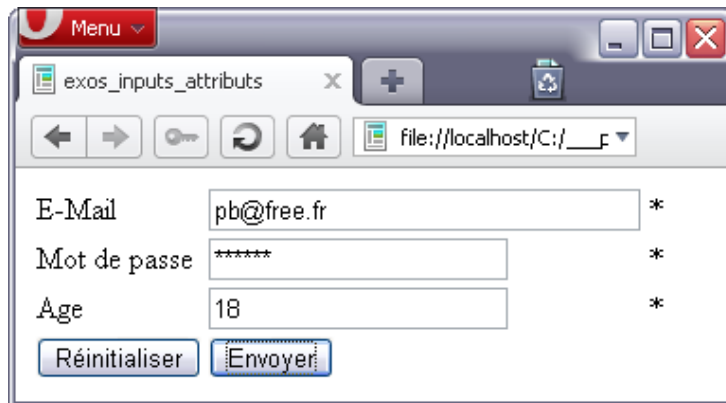
<tr>
<td>Pays : </td>
<td><input type="text" name="pays" list="liste_pays"
value=""/></td>
<td>Liste de suggestions</td>
</tr>
<b><datalist id="liste_pays">
<option label="France" value="FR">
<option label="Italie" value="IT">
<option label="Espagne" value="SP">
</datalist>

```

```
<tr>
<td><input type="reset" /></td>
<td><input type="submit" /></td>
<td>&nbsp;</td>
</tr>
</table>

</form>
</body>

</html>
```

Exercice

The screenshot shows a web browser window with a single tab titled 'exos_inputs_attributs'. The address bar displays 'file:///localhost/C:/_f'. The page content includes a form with three input fields: 'E-Mail' containing 'pb@free.fr', 'Mot de passe' containing seven asterisks, and 'Age' containing '18'. Each field has an asterisk to its right, indicating it is required. Below the fields are two buttons: 'Réinitialiser' and 'Envoyer'.

Tous les champs sont obligatoires.

Le mot de passe est au minimum de 5 caractères. Les caractères autorisés sont les lettres sans accent, les chiffres et le _.

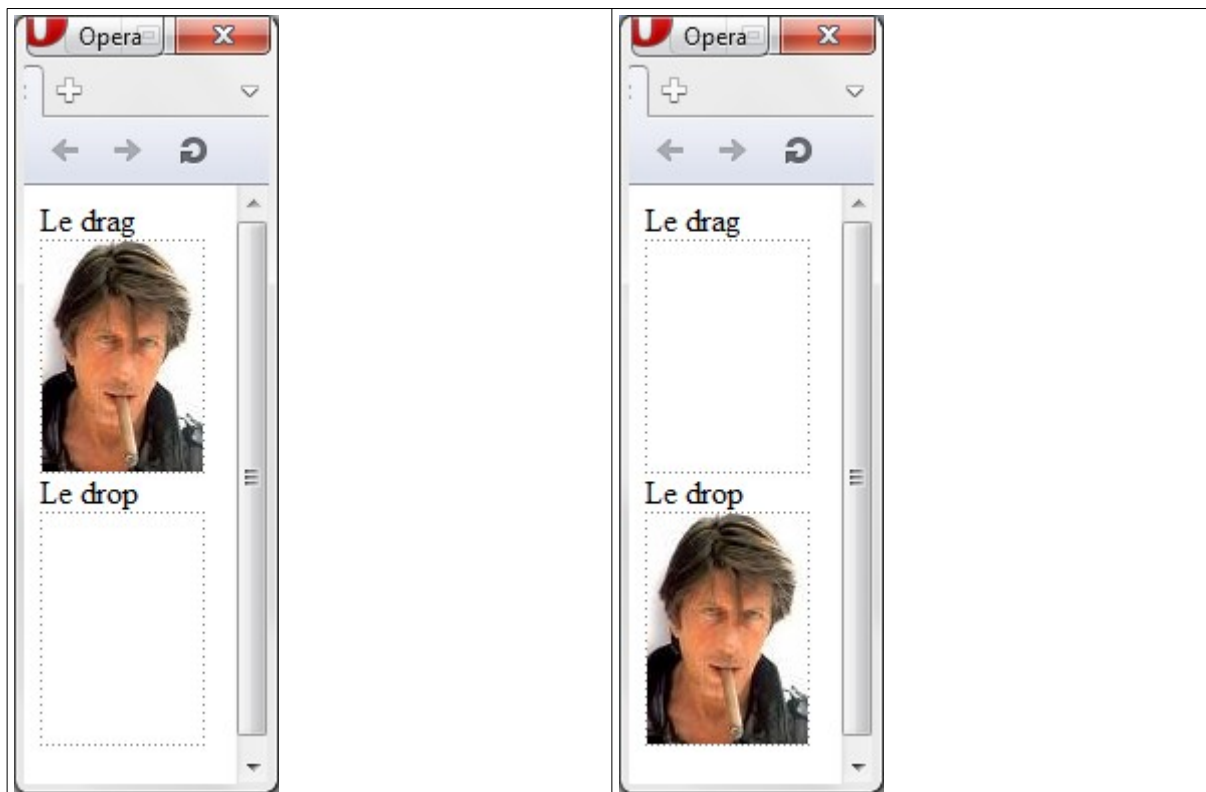
L'âge est composé de 1 à 3 chiffres.

1.9 - LE DRAG AND DROP

1.9.1 - Présentation

Draguer/Larguer ou Glisser/Déposer

ie déplacer un élément ...



1.9.2 - Syntaxes

Attribut de la zone de drag :

`draggable="true"`

Fonction à solliciter lorsque le DRAG commence

```
source.ondragstart = fonction;
```

Fonction à solliciter lorsque l'on entre dans la zone de DROP

```
cible.ondragover = fonction;
```

Fonction à solliciter lors du DROP

```
cible.ondrop = fonction;
```

Autres événements :

`ondrag`, `ondragleave`, `ondrageend`, `ondragenter`.

1.9.3 - Code

```
<!DOCTYPE html>

<!-- Pour l'instant ... 1/3/2013 -->
<!-- OK avec Chrome 25.0.1364.172 et Opera 12.14 -->
<!-- KO avec IE 9.8112.x -->
<!-- KO FX 19.x.x car ouvre une nouvelle fenetre !!! -->

<html>
  <head>
    <title>DragAndDrop.html</title>
    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

    <style>
      div{ border: 1px dotted gray; width: 80px; height:
115px;}
    </style>

  </head>

  <body>
    <label>Zone de drag</label><br/>
    <div>
      <br/>
    </div>

    <label>Zone de drop</label><br/>
    <div id="cible"></div><br/>
  </body>
</html>
```


DragAndDrop.js

```

/*
 * KO avec des input type text ?
 * OK avec <div>, <span>, <img>
 */

// Declaration des variables
var source, cible;

// -----
function init() {
    source = document.getElementById("source");
    cible = document.getElementById("cible");

    // --- Debut du drag
    source.ondragstart = glisser;
    // --- Arrivee dans la zone de drop
    cible.ondragover = autoriserDrop;
    // --- Drop
    cible.ondrop = déposer;
    //ondrag, ondragleave, ondrageend, ondragenter.
}

// drag()
//function glisser(target, e) {
function glisser(e) {
    e = e || window.event;
    console.log(e);
    console.log("DRAG");
    e.dataTransfer.setData("idElementSource", e.target.id);
} /// glisser

//function autoriserDrop(target, e) {
function autoriserDrop(e) {
    //console.log(e);
    e = e || window.event;
    //console.log(e);
    e.preventDefault();
} /// autoriserDrop

// drop()
//function déposer(target, e) {
function déposer(e) {
    //e = e || window.event;
    //e = window.event;
    var id = e.dataTransfer.getData("idElementSource");
    e.target.appendChild(document.getElementById(id));
    console.log("DROP");
    console.log("e : " + e);
    console.log("id : " + id);
} /// déposer

/*

```

```
* INIT
*/
window.onload = init;
```

1.10 - LE STOCKAGE DE DONNÉES CÔTÉ CLIENT

https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API

1.10.1 - Présentation

Avec HTML4 seuls les cookies permettent de faire du stockage local. Ce stockage peut être persistant au-delà de la session.

Les limites de ce système sont les suivantes :

- ✓ L'internaute doit autoriser les cookies,
- ✓ Le nombre et la taille des cookies sont limités (4 ko par cookie, 50 cookies par domaine, 300 au maximum),
- ✓ Les cookies transitent sur le réseau.

Avec HTML5 et JavaScript il est possible de stocker localement des valeurs. Le stockage est réalisé au moyen de :

- ✓ localStorage,
- ✓ sessionStorage,
- ✓ Web SQL Database (hors standard depuis novembre 2010),
- ✓ IndexedDB,
- ✓ fileStorage.

La limite du stockage est de 5 mo, pour la plupart des navigateurs, par domaine. Le transit sur le réseau est nul.

Les paragraphes suivant examineront ces différents moyens, à l'exception du fileStorage.

1.10.2 - Mise en place

Technique	Application
LocalStorage	Préférences des utilisateurs
SessionStorage	Authentification
Web Database	No future !
IndexedDB	Gestion des élèves
FileStorage	TODO

1.11 - LE STOCKAGE LOCAL (LOCALSTORAGE)

1.11.1 - Présentation

<http://www.w3.org/TR/webstorage/>

L'API Web Storage dispose d'une interface Storage et de deux attributs : sessionStorage et localStorage.

Le stockage est réalisé dans des paires key/value.
Par défaut 5 Mo de données peuvent être stockés.

Avec Opera le cache est géré dans Outils/Avancé/, dans Outils/Préférences et dans Affichage/Outils de développement/

Avec Firefox le cache est géré dans Options/Vie privée

Avec Chrome le cache est géré dans Settings/Paramètres/Paramètres avancés/Paramètres de contenu ou dans Settings/Historique/ et dans Settings/Outils/Outils de développement/Ressources

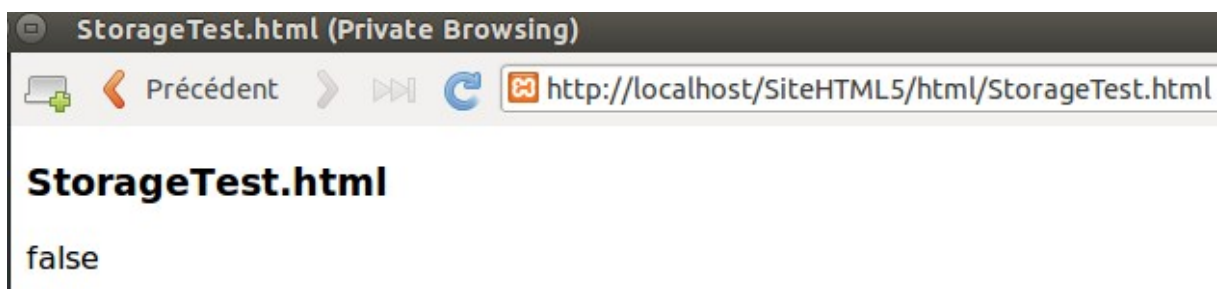
Avec IE le cache est géré dans ...

1.11.2 - Tester la disponibilité du LocalStorage

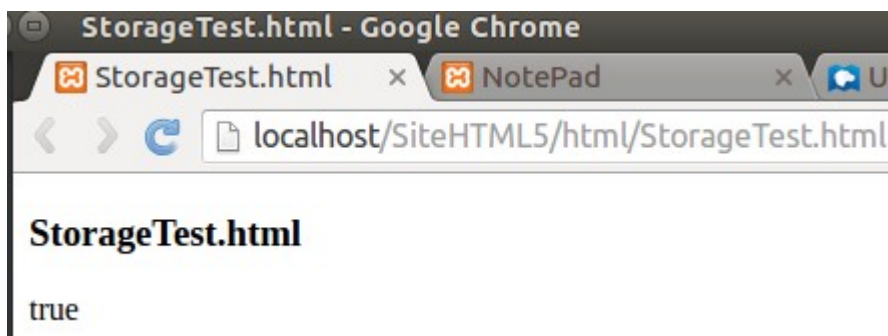
https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API

« Un navigateur peut supporter localStorage, mais pas le rendre disponible pour les scripts sur la page. Un exemple de cela est Safari, qui en mode de navigation privée nous donne un objet localStorage vide avec un quota de zéro, de manière efficace le rendant inutilisable. »

Exemple avec le navigateur Mitori.



Exemple avec Chrome



```
<!DOCTYPE html>
<!--
StorageTest.html
-->
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <title>StorageTest.html</title>
  </head>

  <body>
    <div>
      <h3>StorageTest.html</h3>
      <label id="lblMessage"></label>
    </div>
    <script>
      function storageAvailable(type) {
        try {
          // Ici sessionStorage avec comme valeur
          __storage_test__ ...
          var storage = window[type], x =
            '__storage_test__';
          storage.setItem(x, x);
          storage.removeItem(x);
          return true;
        }
        catch (e) {
          return false;
        }
      }

      var lbStorage = storageAvailable('sessionStorage');
      document.getElementById("lblMessage").innerHTML =
lbStorage;
    </script>
  </body>
</html>
```

1.11.3 - Syntaxes

Stockage d'une variable.

```
localStorage.setItem("nom de variable", "valeur");  
ou  
localStorage["clé"] = "valeur";
```

Tester l'existence d'une variable.

```
if(localStorage.getItem("nom de variable") === null) {
```

Récupération de la valeur d'une variable.

```
localStorage.getItem("nom de variable");  
ou  
localStorage["clé"];
```

Suppression d'une variable.

```
localStorage.removeItem("nom de variable");
```

Suppression de toutes les variables.

```
localStorage.clear();
```

Récupération du nombre de variables

```
localStorage.length;
```

Récupération du nom de la clé.

```
localStorage.key(n); // Commence à 0
```


1.11.4 - Exemple de base

Stocker prénom et nom ...

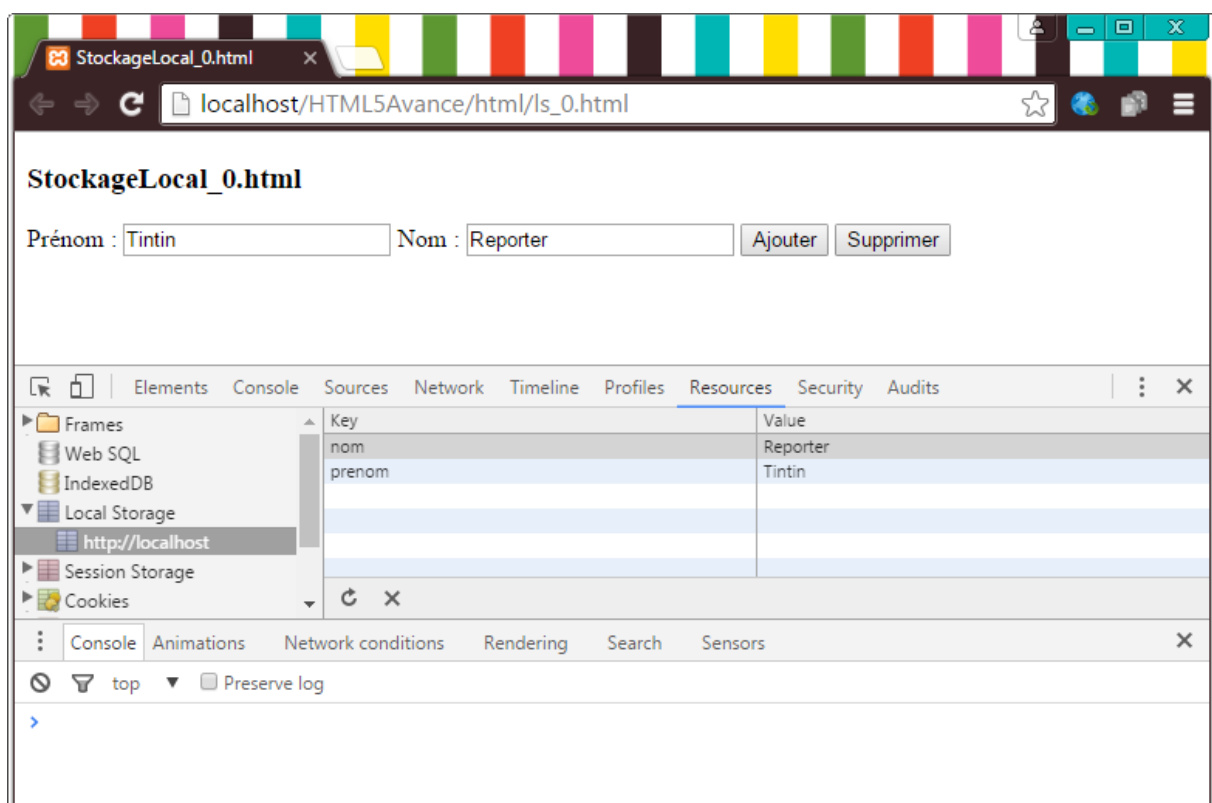
StockageLocal_0.html

Prénom : Nom :

1.11.5 - La console de Chrome

La console : F12 ou Maj/CTRL/i.

Puis Application/Local Storage/http://localhost



```

<!DOCTYPE html>
<!--
StockageLocal_0.html
-->
<html>
  <head>
    <title>StockageLocal_0.html</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <div>
      <h3>StockageLocal_0.html</h3>

      <label>Prénom : </label>
      <input type="text" id="prenom" value="Tintin" />
      <label>Nom : </label>
      <input type="text" id="nom" value="Reporter" />
      <input type="button" value="Ajouter" id="btAjouter" />
      <input type="button" value="Supprimer" id="btSupprimer"

    />

    <script src="../../js/StockageLocal_0.js"></script>
  </div>
</body>
</html>

```

```

/*
 * StockageLocal_0.js
 */

function init() {
  document.getElementById("btAjouter").onclick = function() {
    localStorage.setItem("nom",
document.getElementById("nom").value);
    localStorage.setItem("prenom",
document.getElementById("prenom").value);
  };

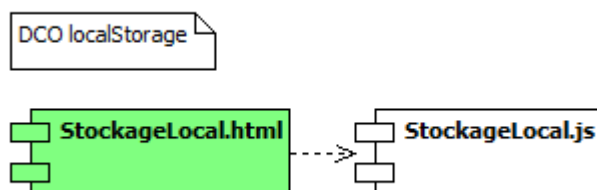
  document.getElementById("btSupprimer").onclick = function() {
    localStorage.removeItem("nom");
    localStorage.removeItem("prenom");
  };
} /// init

// -----
window.onload = init;

```

1.11.6 - Exemple plus ...

Conserver les préférences couleurs de l'internaute via le localStorage.



Le fichier StockageLocal.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>StockageLocal</title>

    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <link href="../../css/formulaire.css" rel="stylesheet" />

    <script src="../../js/StockageLocal.js"></script>
  </head>

  <body>
    <div>
      <label class="etiquette">Couleur du fond : </label>
      <input type="color" id="fond" />
      <br class="nettoyeur"/>

      <label class="etiquette">Couleur du texte : </label>
      <input type="color" id="texte" />
      <br class="nettoyeur"/>

      <p id="message"></p>
    </div>
  </body>
</html>
```

Le fichier StockageLocal.js

```

/*
 * StockageLocal.js
 */

// -----
function ajouter(t) {
    // On ajoute la variable ... via une boucle parce qu'on ne
    // connait pas la clef
    for (var cle in t) {
        localStorage[cle] = t[cle];
    }
} /// ajouter

// -----
function changerCouleurs() {
    // Modifie la couleur de l'element input (ie le rectangle)
    document.getElementById("fond").value = localStorage["fond"];
    // Colorise le body
    document.getElementsByTagName("body")[0].style.backgroundColor
= localStorage["fond"];
    // Modifie la couleur de l'element input (ie le rectangle)
    document.getElementById("texte").value = localStorage["texte"];
    // Colorise le texte du body
    document.getElementsByTagName("body")[0].style.color =
localStorage["texte"];
}

// -----
function init() {
    // Si les variables existent
    if (localStorage["fond"] !== undefined && localStorage["texte"]
!== undefined) {
        changerCouleurs();
        console.log(localStorage["fond"]);
        console.log(localStorage["texte"]);
    }

    // Lorsque au retour de la boîte de dialogue ...
    document.getElementById("fond").onchange = function() {
        var t = new Array();

        t["fond"] = document.getElementById("fond").value;
        // Modification du storage
        ajouter(t);
        // Modification de l'interface
        changerCouleurs();
    };

    document.getElementById("texte").onchange = function() {
        var t = new Array();

        t["texte"] = document.getElementById("texte").value;

```

```
        ajouter(t);
        changerCouleurs();
    };

} /// init

// --- Chargement de la page
window.onload = init;
```

Exercices

Ajoutez le code pour supprimer les variables.

Lorsque le paragraphe "Créer une application OffLine" aura été vu rendez cette page OffLine (obsolète).

1.11.7 - Avec Firefox

La console : F12.

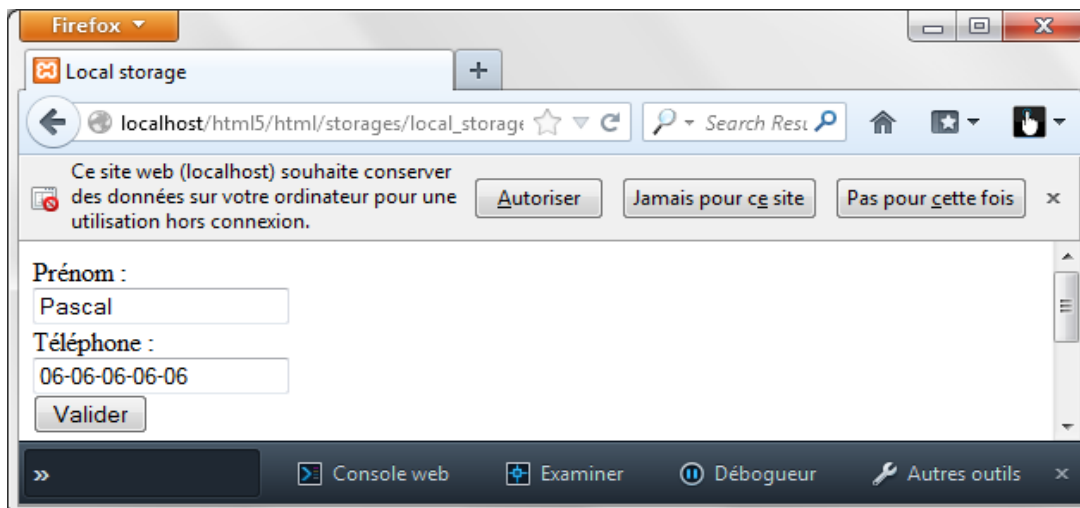
Pour visualiser le localStorage, il faut installer FireStorage Plus !

<https://addons.mozilla.org/en-US/firefox/addon/firestorage-plus/>

Pour le cache :

La première fois que la page est chargée si le cache manifest est activé (<html manifest="StockageLocal.manifest">).

Dans tous les cas, à la différence de Chrome, les valeurs sont conservées lorsque l'on a quitté le navigateur puis l'on est revenu sur la page après l'avoir relancé.



1.11.8 - La console d'Internet Explorer

La console : F12

1.11.9 - Parser le LocalStorage

Objectif :

Récupérer toutes les variables du LocalStorage (les noms et les valeurs).

LocalStorageParser

ActualiteXML : 1#La promotion du mois#-20 % sur toutes les prestation

Texte1 : Ligne1 Ligne2 Ligne3

Texte2 : aaa bbb ccc

Texte3 : aaa bbb qqg

favorite_tables : {"dc76e9f0c0006e8f919e0c515c66dbba3982f785":[]}

ip : localhost

plateforme : Desktop

Le script :

```
var lsVariablesLocalStorage = "";

for (var i = 0; i < localStorage.length; i++) {
    lsVariablesLocalStorage += localStorage.key(i) + " : " +
    localStorage.getItem(localStorage.key(i)) + "<br />";
}

document.getElementById("resultats").innerHTML =
lsVariablesLocalStorage;
```

1.12 - SESSION STORAGE

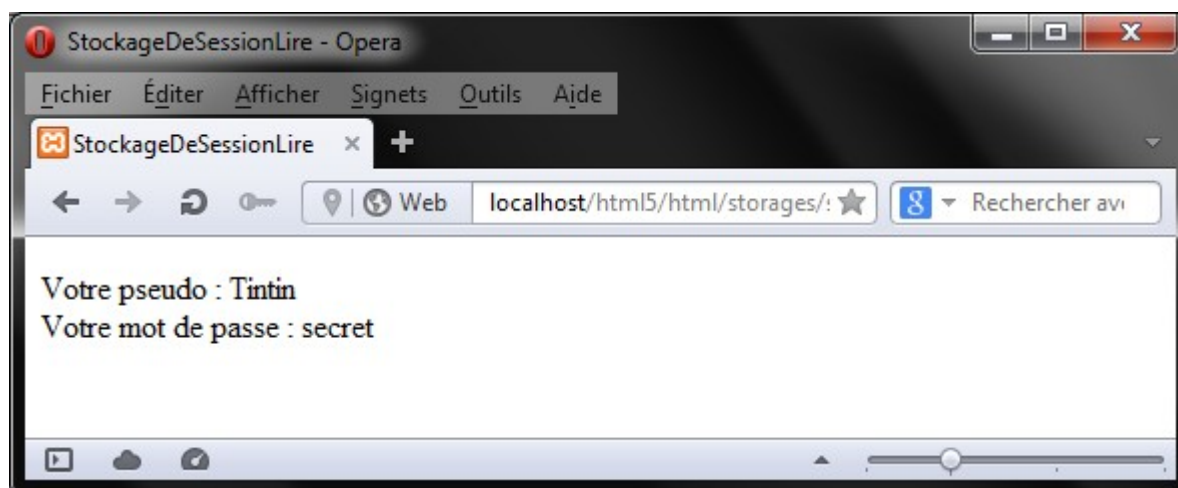
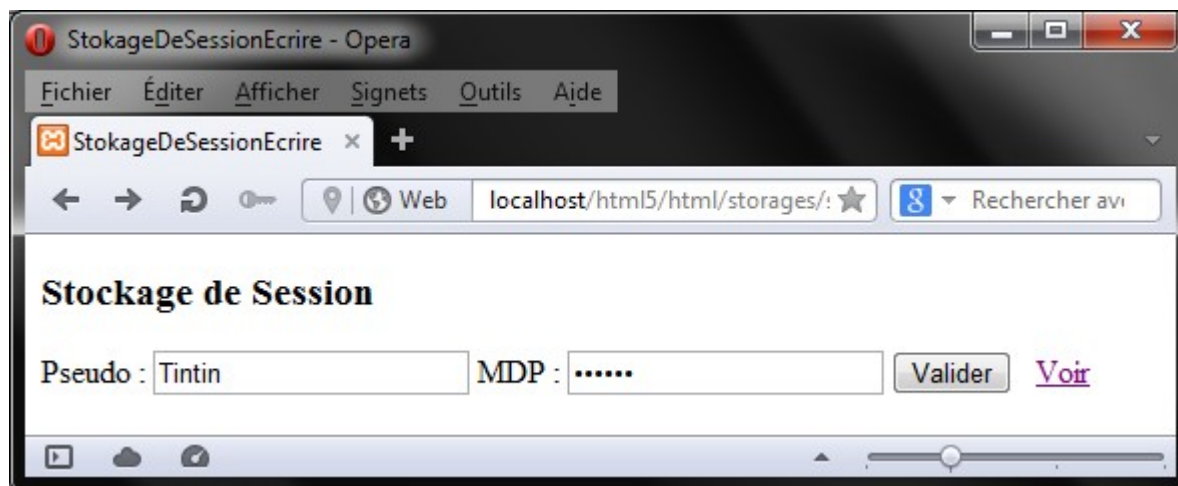
1.12.1 - Présentation

<http://www.w3.org/TR/webstorage/>

Le Session Storage (sessionStorage) dépend de la même interface (Storage) que le Local Storage (localStorage).
Donc les méthodes sont les mêmes.

Cependant le fonctionnement est différent puisque le stockage est valable dans le cadre d'une session. Et là une session est un espace très restreint. Ce n'est pas seulement la fermeture du navigateur qui marque la fin de session. Le passage à un autre onglet par exemple implique de changer d'application et donc les variables de session ne sont pas disponibles. Donc pour tester vous devez créer des liens.

1.12.2 - Exemple



1.12.3 - Scripts

StockageDeSessionEcrire.html

Authentification ... le début ...

```
<!DOCTYPE html>
<html>
  <head>
    <title>StockageDeSessionEcrire</title>
    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>

  <body>
    <h3>Stockage de Session</h3>
    <p>
      <label>Pseudo : </label>
      <input type="text" id="pseudo" value="Tintin" />
      <label>MDP : </label>
      <input type="password" id="mdp" value="secret" />
      <input type="button" value="Ecrire" id="btEcrire"/>
      &nbsp;
      <a href="StockageDeSessionLire.html">Voir</a>
    </p>
    <script src="../../js/StockageDeSessionEcrire.js"></script>
  </body>
</html>
```

StockageDeSessionLire.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>StockageDeSessionLire</title>
    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>

  <body>
    <p>
      <label>Votre pseudo : </label>
      <span id="pseudo"></span>
      <br/>
      <label>Votre mot de passe : </label>
      <span id="mdp"></span>
    </p>
    <script src="../../js/StockageDeSessionLire.js"></script>
  </body>
</html>
```

StockageDeSessionEcrire.js

```
/*
 * StockageDeSessionEcrire.js
 */

document.getElementById("btEcrire").onclick = function() {
    sessionStorage.setItem("pseudo",
document.getElementById("pseudo").value);
    sessionStorage.setItem("mdp",
document.getElementById("mdp").value);
};
```

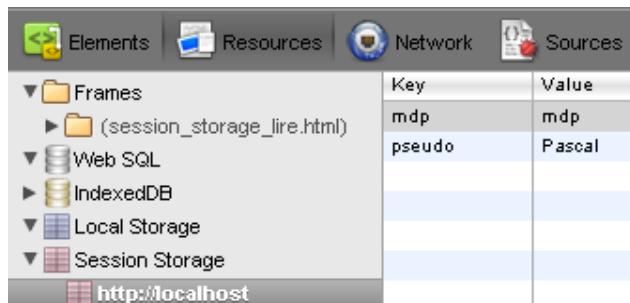
StockageDeSessionLire.js

```
/*
 * StockageDeSessionLire.js
 */

document.getElementById("pseudo").innerHTML =
sessionStorage.getItem("pseudo");
document.getElementById("mdp").innerHTML =
sessionStorage.getItem("mdp");
```

1.12.4 - La console de Chrome

Application/Session Storage/http://localhost



1.12.5 - Exercice : l'authentification

Authentification

Pseudo : MDP :

Authentification OK

[Gérer mon compte](#)

suite à un clic sur le lien ...

Mon Compte

Pseudo : MDP :

ou

Authentification

Pseudo : MDP :

Authentification ratée

[Gérer mon compte](#)

suite à un clic sur le lien ...

Authentification

Pseudo : MDP :

[Vous devez être authentifié](#)

[Gérer mon compte](#)

1.13- CRÉER UNE APPLICATION OFFLINE

```
| sudo /opt/lampp/lampp start
```

```
| sudo /opt/lampp/lampp stop
```

1.13.1 - Principes

Le principe est de rendre une application, un site plus exactement, consultable même lorsque l'internaute est hors-ligne.

https://developer.mozilla.org/fr/docs/Utiliser_Application_Cache



Obsolète

Cette fonctionnalité a été supprimée des standards du Web. Bien que quelques navigateurs puissent encore la supporter, elle est en cours d'éradication. Ne l'utilisez ni dans d'anciens projets, ni dans de nouveaux. Les pages et applications Web l'utilisant peuvent cesser de fonctionner à tout moment.



L'utilisation de la fonction de *mise en cache d'application* décrite ici est actuellement fortement déconseillée; cette fonctionnalité est [en train d'être retiré de la plate-forme Web](#). Utiliser [Service Workers](#) à la place.

<http://www.html5-css3.fr/html5/tutoriel-application-web-offline-html5-cache-manifest>

Avec du code pour mettre à jour le cache via JavaScript.

<http://blog.xebia.fr/2010/12/02/application-hors-ligne-avec-html5-le-manifest/>

Idem.

Comme cela est obsolète il est conseillé d'utiliser les Service Workers qui sont expérimentaux.

https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers

1.13.2 - Démarche

1 - Ajouter l'attribut `manifest="fichier.manifest"` à la balise `<html>`.

2 - Créer un fichier manifest.

Il contient ceci en première ligne : `CACHE MANIFEST`
Ensuite une version en commentaire : `# Version 0.1`
Enfin la liste des fichiers à mettre dans le cache.

3 - Créer un fichier `.htaccess` qui contient cette ligne : `AddType text/cache-manifest manifest`.

Facultatif avec Chrome.

Remarques :

A partir du moment où les éléments (pages, images, scripts, ...) sont dans le cache, donc stockées chez le client, les modifications de ces éléments sur le serveur ne sont pas prises en compte automatiquement par le client.
Il existe divers moyens pour que la mise à jour du cache soit effectuée :

- ✓ modification sur le serveur du fichier `.manifest`,
- ✓ suppression des données de navigation par l'internaute.

1.13.3 - Exemple

La page html : **offline.html**

```
<!DOCTYPE html>
<html manifest="site.manifest">

<head>
  <title>offline</title>
  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />
</head>

<body>
  <div id="main">
    <h1>Ceci est une application offline</h1>
    
    
  </div>
</body>

</html>
```

Le fichier manifest : **site.manifest**

```
CACHE MANIFEST

# Version 0.1

offline.html
../images/jacques_dutronc.jpg
```

Note : seule la photo de Jacques Dutronc est mise dans le cache.

Ajouter ceci au fichier manifest pour que lors de la reconnexion au serveur la deuxième photo soit affichée. Il est possible d'utiliser *.

```
NETWORK:
../images/francoise_hardy.jpg
```

Le fichier .htaccess

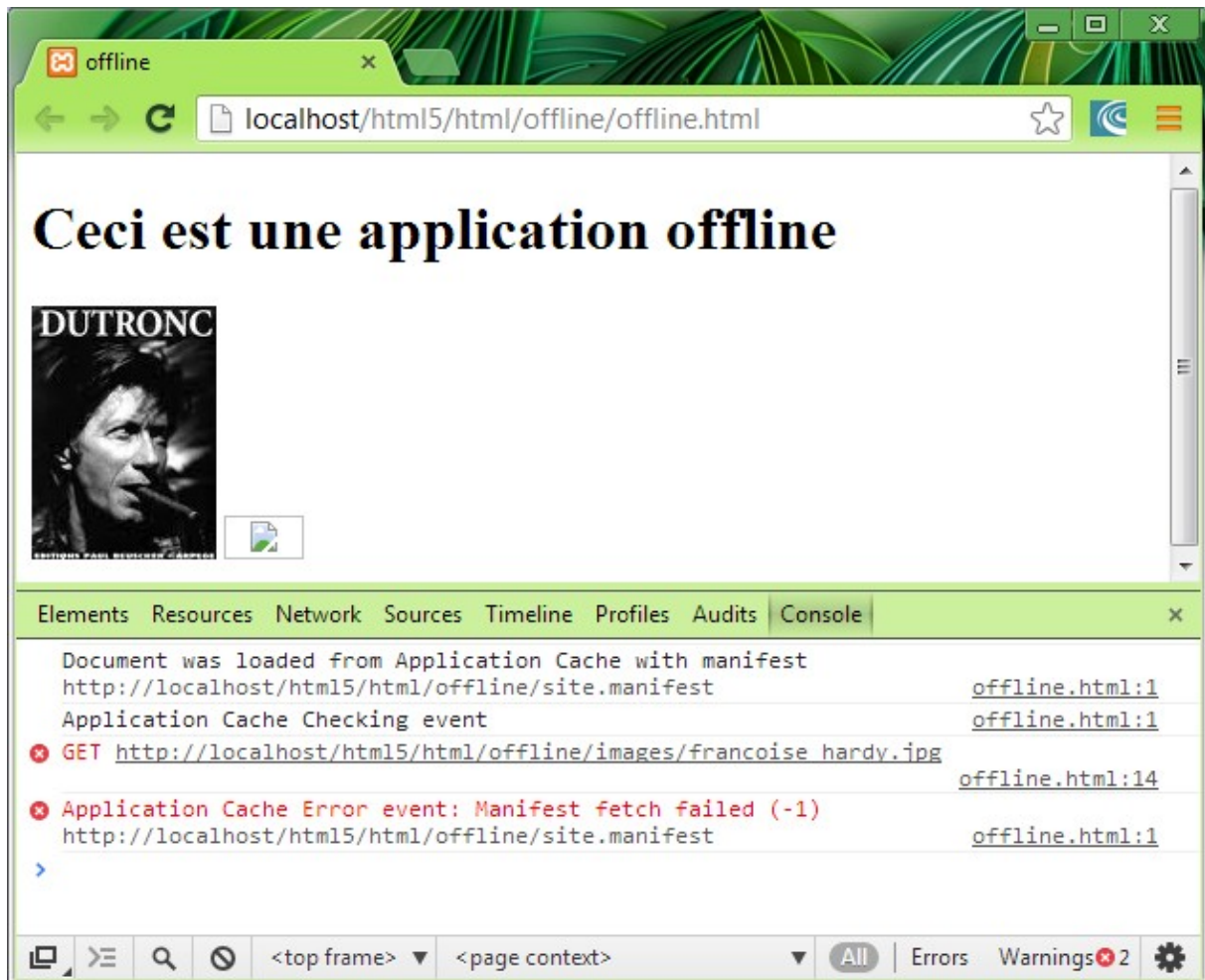
```
AddType text/cache-manifest manifest
```

Ecran lors de la première requête avec la mise en cache.



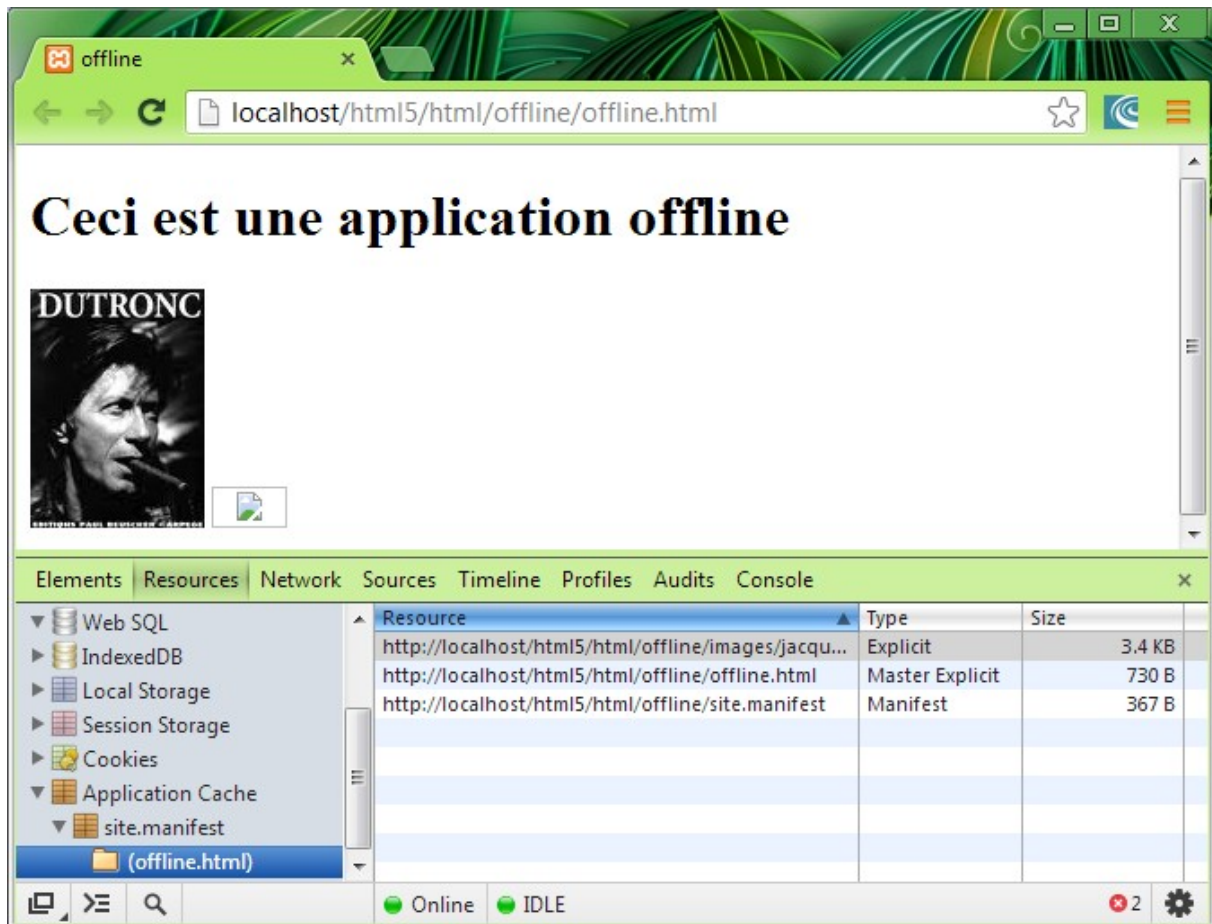
Note : avec l'exemple de code ci-dessus la photo de François Hardy apparaît aussi.

Ecran lors d'une requête alors que l'on est déconnecté du serveur.



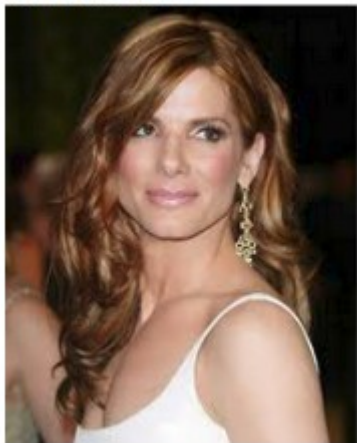
1.13.4 - La console de Chrome

Resources/Application Cache/...



1.13.5 - Exercice : diaporama automatique OnLine et OffLine

Galerie On Line



cf SiteHTML5 et GalerieOnLine.html et GalerieOffLine.html.

1.14 - WEB DATABASE

1.14.1 - Avertissement

<http://www.w3.org/TR/webdatabase/>

Status of This Document

Beware. This specification is no longer in active maintenance and the Web Applications Working Group does not intend to maintain it further.

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the most recently formally published revision of this technical report can be found in the [W3C technical reports index](http://www.w3.org/TR/) at <http://www.w3.org/TR/>.

This document is the 18 November 2010 Working Group Note of Web SQL Database. Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress. The W3C [Web Applications Working Group](#) is the W3C working group responsible for this document.

This document was on the W3C Recommendation track but specification work has stopped. The specification reached an impasse: all interested implementors have used the same SQL backend (Sqlite), but we need multiple independent implementations to proceed along a standardisation path.

The Web Applications Working Group continues work on two other storage-related specifications: [Web Storage](#) and [Indexed Database API](#).

This document is the 18 November 2010 Working Group Note of Web SQL Database.

This document was on the W3C Recommendation track but specification work has stopped. The specification reached an impasse: all interested implementors have used the same SQL backend (Sqlite), but we need multiple independent implementations to proceed along a standardisation path.

Cependant la plupart des navigateurs de bureaux et smartphones continuent de supporter cette API.

1.14.2 - Objectif

Gérer une BD SQL stockée par le client.

Créer ou ouvrir une BD.

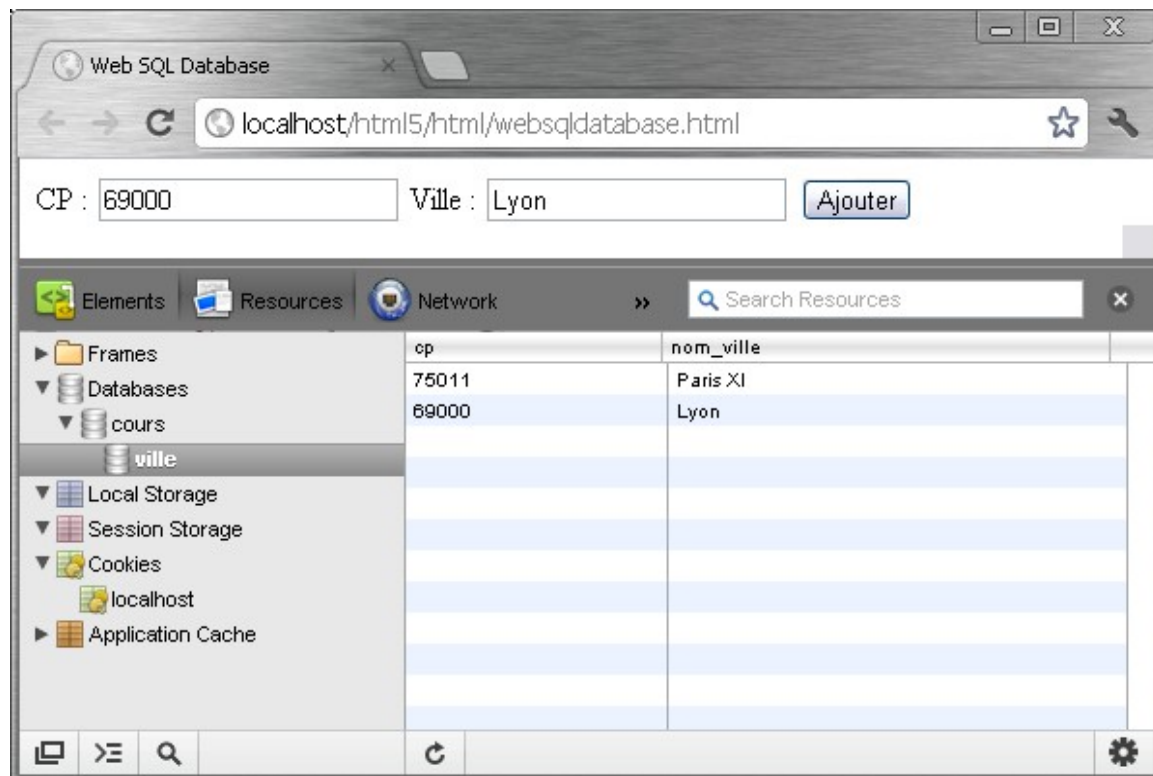
Créer – éventuellement - une table.

Insérer un enregistrement.

Lire les enregistrements.

1.14.3 - La console de Chrome

Resources/Databases/...



1.14.4 - Syntaxes

Ouvrir ou créer une BD

```
db = openDatabase("nom de la bd", "version de la bd",  
"Description", taille de la BD);
```

Exécuter un ordre SQL (LDD ou LMD)

```
db.transaction(function(transaction) {  
    transaction.executeSql("Ordre SQL" [, paramètres [, fonction  
de callback]]);  
});
```

Note : le SQL est standard.

Exemples

Création d'une table (1 argument)

```
tx.executeSql(  
    "CREATE TABLE IF NOT EXISTS ville(" +  
    "cp TEXT NOT NULL, " +  
    "nom_ville TEXT);");
```

Insertion dans une table (2 arguments)

```
tx.executeSql(  
    "INSERT INTO ville(cp, nom_ville) VALUES (?,?);",  
    [asCp, asNomVille]
```

Visualisation d'une table (3 arguments)

```
tx.executeSql("SELECT * FROM ville", [], afficherVilles);
```

1.14.5 - Quelques particularités de SQLITE

1.14.5.1 - Les types de colonnes

TEXT,
INT,
NUM,
REAL,
NONE.

1.14.5.2 - Création d'une table avec une PK auto-incrémentée

```
CREATE TABLE nomDeTable(clePrimaire INTEGER PRIMARY KEY ASC, col1, col2);
```

Pour plus de détails cf SQLITE.doc.

1.14.6 - Exemple

Objectif: l'insertion d'un enregistrement dans la table Villes.



WebSQLDatabase.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Web SQL Database</title>

    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <link href="../../css/formulaire.css" rel="stylesheet" /
  >

  <script src="WebSQLDatabaseDAO.js"></script>
  <script src="WebSQLDatabase.js"></script>
</head>

<body>
  <div>
    <label class="etiquette">CP : </label>
    <input type="text" id="cp" value="69000" />
    <br class="nettoyeur"/>

    <label class="etiquette">Ville : </label>
    <input type="text" id="ville" value="Lyon" />
    <br class="nettoyeur"/>

    <button id="btAjouter">Ajouter</button><br/>
    <button id="btVider">Vider</button><br/>
    <button id="btLire">Lire</button><br/>
  </div>
</body>
</html>
```

WebSQLDatabase.js

```
/*
 * WebSQLDatabase.js
 */
// -----
function init() {
    var db = openORcreateBD("cours", "1.0", "BD exemple", 100 *
1024);
    creerTableVille(db);

    document.getElementById("btAjouter").onclick = function() {
        insererVille(db, document.getElementById("cp").value,
document.getElementById("ville").value);
    };

    document.getElementById("btVider").onclick = function() {
        viderVille(db);
    };

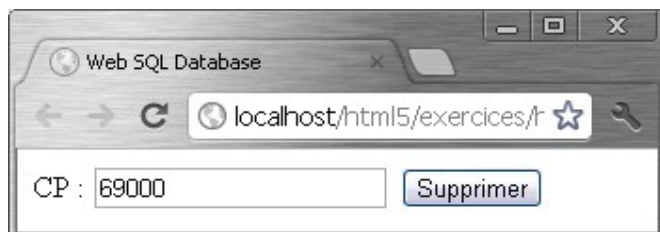
    document.getElementById("btLire").onclick = function() {
        var t = lireVille(db);
    };
} /// init

// --- INIT
window.onload = init;
```

WebSQLDatabaseDAO.js

1.14.7 - Exercice

Supprimer un enregistrement



1.15- INDEXED DATABASE (INDEXEDDB)

1.15.1 - Présentation

<http://www.w3.org/TR/IndexedDB/>

https://developer.mozilla.org/en-US/docs/IndexedDB/Basic_Concepts_Behind_IndexedDB

L'API IndexedDB permet de gérer une BD NoSQL stockée par le client.

La BD est persistante donc utilisable d'une session à l'autre et en mode déconnecté (offline).

L'API utilise l'API Asynchronous – elle est donc non bloquante – et est transactionnelle (aussi bien en écriture qu'en lecture).

Quelques définitions du W3C

Une BD est un entrepôt de stockage d'objets :

"A database comprises one or more object stores which hold the data stored in the database."

Note : les objets sont des objets JSON.

Composants

Les objets : BD → Objets de stockage → Structure (cle + champs) → Données stockées

Equivalents SQL : BD → Tables → Colonnes → Lignes

BD {objets de stockage}

Par exemple : coursIDB { eleves, profs, ...}

Equivalent SQL : BDCours.

Objet de stockage {key, champ1, champ2}

Par exemple : eleves(**id**, prenom, age, email}

Equivalent SQL : table eleves.

Objet de stockage {données stockées}

Equivalent SQL : lignes de la table eleves.

1.15.2 - Objectif

L'objectif de ce paragraphe est de :

- ✓ créer ou ouvrir une BD,
- ✓ créer – éventuellement – un objet de stockage,
- ✓ gérer des transactions,
- ✓ insérer un enregistrement,
- ✓ visualiser un ou plusieurs enregistrements,
- ✓ supprimer un enregistrement,
- ✓ détruire une BD.

Les exercices porteront sur la suppression de tous les enregistrements et la modification d'un enregistrement.

1.15.3 - Syntaxes

1.15.3.1 - Fabriques d'objets BD, connexion, transaction

```
// Factory IBDS
var IDB; // Surtout pas indexedDB en dehors d'une fonction

// Objet BD : une connexion a la BD
var bd;

// Factory Transactions
var IDBTrans; // Surtout pas IDBTransaction en dehors d'une
fonction
```

```
// Factory IBDS
IDB = window.indexedDB || window.webkitIndexedDB ||
window.mozIndexedDB || window.msIndexedDB;
```

L'objet IDB correspond à une Interface vers la gestion d'une IndexedDB.

1.15.3.2 - Ouverture ou création d'une BD

```
// --- Ouverture (Connexion) ou creation de la BD : open("nom de
la BD", version de la BD)
var requeteCreationOuverture = IDB.open(nomDeLaBD, 1);

// --- success and error handlers
// --- Si erreur
requeteCreationOuverture.onerror = function(evt) {
    // Code de gestion d'erreur
};

// --- Si OK
requeteCreationOuverture.onsuccess = function(evt) {
    // Code de connexion
}
```

L'objet IDB est une interface vers la gestion de BD de type IndexedDB.

1.15.3.3 - Connexion à la BD

```
bd = requeteCreationOuverture.result;
```

L'objet bd est de type IDBDatabase.

Il correspond à une connexion à la base de données.

1.15.3.4 - Déconnexion de la BD

```
bd.close();
```

L'objet bd est de type IDBDatabase.
Il correspond à une connexion à la base de données.

1.15.3.5 - Suppression d'une BD

```
var requeteDeSuppressionDeBD = IDB.deleteDatabase(nomDeLaBD);
```

Il faut être déconnecté de la BD au préalable.

1.15.3.6 - Création d'un objet de stockage

Il s'agit d'un document NoSQL, semblable à une table SQL. Une liste de structures.

```
var objetStoque =  
evt.currentTarget.result.createObjectStore("objetDeStockage",  
{ keyPath: "id", autoIncrement: true });
```

1.15.3.7 - Création d'index

Index non unique : permettra un accès indexé avec la méthode get("").

```
objetStoque.createIndex("nomIndex", "nomDuChampClef", { unique:  
false });
```

Index unique

```
objetStoque.createIndex("nomIndex", "nomDuChampClef", { unique:  
true });
```

Note : cf aussi deleteIndex("nomIndex").

1.15.3.8 - Ajout de données

Initialisation d'une transaction.
La valeur par défaut est "readwrite".

```
var transaction = bd.transaction(objetDeStockage, "readwrite");
```

Note : pour les transactions cf aussi les événements onabort, oncomplete, onerror.

Récupération d'un objet de la BD

```
var objetStocke = transaction.objectStore(objetDeStockage);
```

Ajout de données dans un objet

```
var requeteDAjout = objetStocke.add(donnees);
```

```
var requeteDAjout = objetStocke.add({ prenom: "un prénom", age: "un  
âge", email: "un email" });
```

ou

```
var data = {};  
data.prenom = document.getElementById("itPrenom").value;  
data.age = document.getElementById("itAge").value;  
data.email = document.getElementById("itEmail").value;  
  
var requeteDAjout = objetStocke.add(data);
```

1.15.3.9 - Visualisation de données

Récupération d'un enregistrement via l'id :

```
var requeteLecture = objetStocke.get(id);
```

En cas de succès

```
requeteLecture.onsuccess = function(evt) {
```

Dans le cadre de la fonction anonyme

```
var valeur = evt.target.result.nomDuChamp;
```

Récupération d'un enregistrement via un index :

choix de l'index utilisé

```
var idx = objetStocke.index("nom de l'index");
```

récupération

```
var requeteLecture = objetStocke.get("valeur dans l'index");
```

Gestion d'un curseur :

Ouverture du curseur :

```
var requeteLecture = objetStocke.openCursor();
```

Récupération du curseur :

```
var curseur = evt.target.result;
```

Récupération d'un champ clé :

```
curseur.key
```

Récupération d'un champ non clé :

```
curseur.value.nomDuChamp
```

Fetch :

```
curseur.continue();
```

Note : cf aussi `cursor.advance(n)`, `cursor.delete()`, `cursor.update()`.

1.15.3.10 - Suppression de données

La méthode delete()

```
var requeteSuppression = objetStoque.delete(id);
```

1.15.4 - Exemple

1.15.4.1 - L'écran

Le CRUD :

Lycée-Elèves

Id :

Prénom :

Age :

E-mail :

Créer ou ouvrir IDB

Ajouter

Voir

Voir Un

Supprimer Un

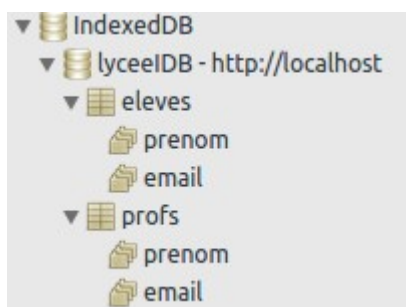
Supprimer

Modifier

Supprimer IDB

id [1] Henri a 18 ans et son e-mail est henri@free.fr
 id [2] Uomo a 17 ans et son e-mail est uomo@free.fr
 id [3] Donna a 16 ans et son e-mail est donna@free.it
 Fin de la partie!!!

1.15.4.2 - La BD



1.15.4.3 - bdIndexee.html

```

<!DOCTYPE html>
<html>
  <head>
    <title>bdIndexee</title>
    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <link href="formulaire.css" rel="stylesheet" />
  </head>
  <body>
    <div>
      <h3>Lycée-Elèves</h3>
      <label class="etiquette">Id : </label>
      <input type="text" id="itIdEleve" value="" />
      <br class="nettoyeur"/>

      <label class="etiquette">Prénom : </label>
      <input type="text" id="itPrenom" value="" />
      <br class="nettoyeur"/>

      <label class="etiquette">Age : </label>
      <input type="text" id="itAge" value="" />
      <br class="nettoyeur"/>

      <label class="etiquette">E-mail : </label>
      <input type="text" id="itEmail" value="" />
      <br class="nettoyeur"/>

      <button id="btCreerOuvrirIDB">Créer ou ouvrir
IDB</button><br/>
      <button id="btAjouter">Ajouter</button><br/>
      <button id="btVoir">Voir</button><br/>
      <button id="btVoirUn">Voir Un</button><br/>
      <button id="btVoirViaIndex">Voir Un Via
Index</button><br/>
      <button id="btSupprimerUn">Supprimer Un</button><br/>
      <button id="btSupprimer">Supprimer</button><br/>
      <button id="btModifierUn">Modifier</button><br/>
      <button id="btSupprimerIDB">Supprimer IDB</button><br/>

      <p id="pResultats"></p>
    </div>

    <script src="bdIndexee.js"></script>
  </body>
</html>

```

1.15.4.4 - formulaire.css

```
/*  
    formulaire.css  
*/  
.etiquette{float: left; display: block; width: 150px;}  
.nettoyeur{clear: both;}  
button{width: 300px;}
```

1.15.4.5 - bdIndexee.js

WARNING : rien ne s'affichera car mode Asynchrone et donc lsMessage = chaîne vide. Il faudrait afficher dans les méthodes.

```

/*
 * bdIndexee.js
 */

/*
 * La BD : "lyceeIDB"
 * La "table" ... un objet : "eleves"
 * La cle : id-NUMBER-AutoIncrement
 * Les champs : prenom, age, email
 * Les index : prenom, email (unique)
 */

/*
 * Variables globales
 */
var itIdEleve, itPrenom, itAge, itEmail;
var pResultats;

var gsNomDeLaBD = "lyceeIDB";

// Factory IBDs
var IDB; // Surtout pas indexedDB en dehors d'une fonction
// Objet BD : une connexion a la BD
var bd;
// Factory Transactions
var IDBTrans; // Surtout pas IDBTransaction en dehors d'une
fonction

var IDBCles;

// --- Donnees pour l'exemple
var tDonnees = [
    {prenom: "Uomo", age: 17, email: "uomo@free.fr"},
    {prenom: "Donna", age: 16, email: "donna@free.it"}
];

// -----
function init() {
    /*
     * Les elements d'interface
     */
    itIdEleve = document.getElementById("itIdEleve");
    itPrenom = document.getElementById("itPrenom");
    itAge = document.getElementById("itAge");
    itEmail = document.getElementById("itEmail");

    pResultats = document.getElementById("pResultats");

```

```
/*
 * Les evenements
 */
document.getElementById("btCreerOuvrirIDB").onclick =
function() {
    pResultats.innerHTML = creerOuvrirBD(gsNomDeLaBD);
};

document.getElementById("btAjouter").onclick = function() {
    //{ prenom: lsPrenom, age: lsAge, email: lsEmail }
    var data = {};

    data.prenom = itPrenom.value;
    data.age = itAge.value;
    data.email = itEmail.value;

    pResultats.innerHTML = ajouter(bd, "eleves", data);
};

document.getElementById("btVoir").onclick = function() {
    pResultats.innerHTML = voir(bd, "eleves");
};

document.getElementById("btVoirUn").onclick = function() {
    pResultats.innerHTML = voirUn(bd, "eleves",
itIdEleve.value);
};

document.getElementById("btVoirViaIndex").onclick = function()
{
    pResultats.innerHTML = voirViaIndex(bd, "eleves",
itPrenom.value);
};

document.getElementById("btSupprimerUn").onclick = function() {
    pResultats.innerHTML = supprimerUn(bd, "eleves",
itIdEleve.value);
};

document.getElementById("btSupprimer").onclick = function() {
    pResultats.innerHTML = supprimer(bd, "eleves");
};

document.getElementById("btModifierUn").onclick = function() {
    var data = {};

    data.prenom = itPrenom.value;
    data.age = itAge.value;
    data.email = itEmail.value;

    pResultats.innerHTML = modifierUn(bd, "eleves",
itIdEleve.value, data);
};

document.getElementById("btSupprimerIDB").onclick = function()
{
```

```

        pResultats.innerHTML = supprimerBD(gsNomDeLaBD, IDB, bd);
    };
} /// init

/*
 * Creation
 * ou Mise a jour d'une BD
 * ou Ouverture de la BD
 */
function creerOuvrirBD(psNomDeLaBD) {
    var lsMessage = "Message";
    // --- Creation d'un Objet indexedDB
    // IDB = window.webkitIndexedDB; // Pour Chrome
    // --- Pour Standard ou Chrome ou FX ou IE ou MS
    // Factory IBDS
    IDB = window.indexedDB || window.webkitIndexedDB ||
window.mozIndexedDB || window.msIndexedDB;

    // --- Ouverture (Connexion) ou creation de la BD : open("nom
de la BD", version de la BD)
    var requeteCreationOuverture = IDB.open(psNomDeLaBD, 1);

    // --- success and error handlers
    // --- Si erreur
    requeteCreationOuverture.onerror = function(evt) {
        lsMessage = "Code erreur BD : " + evt.target.errorCode;
    }; /// create onerror

    // --- Si OK
    requeteCreationOuverture.onsuccess = function(evt) {
        lsMessage = "Création/Ouverture de la BD réussie-" +
evt.toString();

        // --- La variable global bd pointe vers la BD
        bd = requeteCreationOuverture.result;
    }; /// create onsuccess

    // --- Si la version de la BD a change ou si creation de la BD
    requeteCreationOuverture.onupgradeneeded = function(evt) {
        lsMessage = "Création de la BD réussie";

        // Creation d'"une" table (un ObjectStore) nommee [eleves]
avec une PK de nom "id" auto_incrementee
        // ObjectStore = Objet de stockage
        var objetStoque =
evt.currentTarget.result.createObjectStore("eleves", {keyPath:
"id", autoIncrement: true});

        // Creation des index sur les champs (facultatif)
        // createIndex(nomIndex, cle, contrainte)
        // --- Index non unique sur le prenom; permettra un acces
indexe via get("un prenom")
        objetStoque.createIndex("idx_prenom", "prenom", {unique:
false});

```

```

        // --- Index unique sur l'email
        objetStocke.createIndex("idx_email", "email", {unique:
true});

        // --- Ajout d'un enregistrement
        // --- En fait les donnees sont des objets donc a cle
        objetStocke.add({prenom: "Henri", age: 18, email:
"henri@free.fr"});

        // --- Ajouts de plusieurs enregistrements
        for (i in tDonnees) {
            objetStocke.add(tDonnees[i]);
        }

        // --- Creation d'un objet de stockage nomme [profs]
        var objetStockeProfs =
evt.currentTarget.result.createObjectStore("profs", {keyPath: "id",
autoIncrement: true});

        // Creation des index sur les champs (facultatif)
        // createIndex(nomIndex, cle, contrainte)
        // --- Index non unique sur le prenom; permettra un acces
indexe get("")
        objetStockeProfs.createIndex("idx_prenom", "prenom",
{unique: false});
        // --- Index unique sur l'email
        objetStockeProfs.createIndex("idx_email", "email", {unique:
true});

        // --- Ajout d'un enregistrement
        // --- En fait les donnees sont des objets donc a cle
        objetStockeProfs.add({prenom: "Pascal", email:
"pascal@free.fr"});
        objetStockeProfs.add({prenom: "Sophie", email:
"sophie@free.fr"});

        lsMessage = "Ajouts de l'objet et des index et des données
réussis : " + evt.toString();
    };
    return lsMessage;
} /// create onupgradeneeded

/*
 * Suppression d'une BD
 * La BD doit etre fermee
 */
function supprimerBD(psNomDeLaBD, IDB, bd) {
    var lsMessage = "Message";
    try {
        if (IDB === undefined) {
            // --- Creation d'un objet d'accès a la BD
            IDB = window.indexedDB || window.webkitIndexedDB ||
window.mozIndexedDB || window.msIndexedDB;
        }
    }

```



```

        else {
            // Fermeture de la BD
            lsMessage = "Fermeture de la BD réussie";
            bd.close();
        }

        // --- Le nom de la BD est sensible a la casse
        // Suppression effective de la BD
        var requeteDeSuppression = IDB.deleteDatabase(psNomDeLaBD);

        // --- Erreur lors de la suppression
        requeteDeSuppression.onerror = function(evt) {
            lsMessage = "Erreur de suppression - " +
(evt.target.error ? evt.target.error : evt.target.errorCode);
        };

        // --- Suppression réussie
        requeteDeSuppression.onsuccess = function() {
            lsMessage = "Suppression de la BD réussie";
        };
    }
    catch (ex) {
        lsMessage = "Exception lors de la destruction de la BD - "
+ ex.message;
    }
    return lsMessage;
} /// supprimerBD

/*
 * Ajout des donnees
 */
function ajouter(bd, objetDeStockage, donnees) {
    var lsMessage = "Message";
    try {
        // --- Creation d'un objet de gestion de transaction
        var transaction = bd.transaction(objetDeStockage,
"readwrite");

        // --- Creation d'un objet de stockage
        var objetStoque = transaction.objectStore(objetDeStockage);

        // --- Ajout d'un enregistrement
        //var requeteDAjout = objetStoque.add({ prenom: lsPrenom,
age: lsAge, email: lsEmail });
        var requeteDAjout = objetStoque.add(donnees);

        // --- Si erreur d'ajout
        requeteDAjout.onerror = function(evt) {
            lsMessage = "Ajout dans la BD raté : " +
evt.target.error.name;
        };
        // --- Si OK
        requeteDAjout.onsuccess = function(evt) {

```

```

        lsMessage = "Ajout dans la BD réussi : " +
evt.toString();
    }; /// success
} /// try
catch (ex) {
    lsMessage = "Exception lors de l'ajout de données - " +
ex.message;
}
    return lsMessage;
} /// ajouter

/*
 * Suppression d'un enregistrement
 * via delete(cle)
 */
function supprimerUn(bd, objetDeStockage, id) {
    var lsMessage = "Message";
    try {
        id = parseInt(id);
        // WARNING : le parametre pour get est fortement type : "1"
est different de 1
        // ici c'est 1 ou 2 ... car id est number auto-increment

        // --- Transaction en lecture/ecriture
        var transaction = bd.transaction(objetDeStockage,
"readwrite");

        var objetStocke = transaction.objectStore(objetDeStockage);

        // --- Suppression d'un enregistrement avec delete(cle)
        var requeteSuppression = objetStocke.delete(id);

        requeteSuppression.onerror = function(evt) {
            lsMessage = "Suppression dans la BD ratée : " +
evt.target.error.id;
        }; /// error suppression

        requeteSuppression.onsuccess = function(evt) {
            lsMessage = "Enregistrement supprimé";
        }; /// success suppression
    } /// try

    catch (ex) {
        lsMessage = "Exception lors de la suppression d'un
enregistrement - " + ex.message;
    }
    return lsMessage;
} /// supprimerUn

/*
 * Voir un enregistrement
 */

```

```

function voirUn(bd, objetDeStockage, id) {
    var lsMessage = "Message";
    try {
        id = parseInt(id);
        // WARNING : le parametre pour get est fortement type : "1"
        // est different de 1
        // ici c'est 1 ou 2 ... car id est number auto-increment

        // --- Transaction en lecture seule
        var transaction = bd.transaction(objetDeStockage);

        var objetStocke = transaction.objectStore(objetDeStockage);

        // --- Recupere l'enregistrement ayant l'id = id
        var requeteLecture = objetStocke.get(id);

        requeteLecture.onerror = function(evt) {
            lsMessage = "Lecture dans la BD ratée : " +
            evt.target.error.name;
        }; /// error lecture

        requeteLecture.onsuccess = function(evt) {
            // --- Si l'enregistrement existe
            if (evt.target.result != undefined) {
                lsMessage = "Prénom : " + evt.target.result.prenom;

                lsMessage = "<br/>id [" + id + "] " +
                evt.target.result.prenom + " a " + evt.target.result.age + " ans et
                son e-mail est " + evt.target.result.email;
            }
            else {
                lsMessage = "<br/>id [" + id + "] n'existe pas";
            }
        }; /// success Lecture
    } /// try

    catch (ex) {
        lsMessage = "Exception lors de la visualisation d'une
        donnée - " + ex.message;
    }
    return lsMessage;
} /// voirUn

/*
 * Voir un enregistrement via Index
 */
function voirViaIndex(bd, objetDeStockage, prenom) {
    var lsMessage = "Message";
    try {
        // --- Transaction en lecture seule
        var transaction = bd.transaction(objetDeStockage);

        var objetStocke = transaction.objectStore(objetDeStockage);
        var idx = objetStocke.index("prenom");

        // --- Recupere l'enregistrement ayant l'id = id
    }

```

```

    var requeteLecture = idx.get(prenom);

    requeteLecture.onerror = function(evt) {
        lsMessage = "Lecture dans la BD ratée : " +
evt.target.error.name;
    }; /// error lecture

    requeteLecture.onsuccess = function(evt) {
        // --- Si l'enregistrement existe
        if (evt.target.result != undefined) {
            lsMessage = "<br/>" + evt.target.result.prenom + "
a " + evt.target.result.age + " ans et son e-mail est " +
evt.target.result.email;
        }
        else {
            lsMessage = "<br/>[" + prenom + "] n'existe pas";
        }
    }; /// success Lecture
} /// try

catch (ex) {
    lsMessage = "Exception lors de la visualisation d'une
donnée - " + ex.message;
}
return lsMessage;
} /// voirViaIndex

/*
 * Voir tous les enregistrements
 */
function voir(bd, objetDeStockage) {
    var lsMessage = "";
    try {
        // --- Transaction en lecture seule
        var transaction = bd.transaction(objetDeStockage);

        var objetStocke = transaction.objectStore(objetDeStockage);

        // Ouvre un curseur
        var requeteLecture = objetStocke.openCursor();

        requeteLecture.onerror = function(evt) {
            lsMessage = "Lecture dans la BD ratée : " +
evt.target.error.name;
        }; /// error lecture

        requeteLecture.onsuccess = function(evt) {
            var curseur = evt.target.result;
            if (curseur) {
                lsMessage += "<br/>id [" + curseur.key + "] " +
curseur.value.prenom + " a " + curseur.value.age + " ans et son e-
mail est " + curseur.value.email;
                // On fetch a nouveau
                curseur.continue();
            }
        }
    }
}

```

```
        else {
            lsMessage += "<br/>Fin de la partie!!!";
        }
    }; /// success lecture
} /// try

catch (ex) {
    lsMessage = "Exception lors de la visualisation des données
- " + ex.message;
}
return lsMessage;
} /// voir

/*
 *
 */
window.onload = init;
```

1.15.5 - Exercices

- 1) Supprimer tous les enregistrements.
- 2) Modifier un enregistrement.
- 3) A faire après la géolocalisation, récupérer des coordonnées de villes à stocker dans une IndexedDB



Pays : Ville :

Message :

1.16 - FILESTORAGE

https://developer.mozilla.org/en-US/docs/Using_files_from_web_applications

https://developer.mozilla.org/en-US/docs/Using_files_from_web_applications

Heads up!

In April 2014, it was announced on public-webapps that the Filesystem API spec is not being considered by other browsers. For now, the API is Chrome-specific and it's unlikely to be implemented by other browsers and is no longer being standardized with the W3C.

1.17 - TESTER L'IMPLÉMENTATION DE HTML5 ET AGIR EN CONSÉQUENCE

Référence : <http://diveintohtml5.org/detect.html>

1.17.1 - Tester l'implémentation de HTML5

Objectif

Nous avons déjà vu quelques éléments de testing pour HTML5 et CSS3. Mais il existe une bibliothèque JS qui permet de le faire : MODERNIZR à <http://www.modernizr.com/>.

Au 1^{er} juillet 2015 la version est la 2.8.3.

Elle teste, en autres, les types et attributs suivants :

Input attributes : autocomplete, autofocus, list, placeholder, max, min, multiple, pattern, required, step.

Types : search, tel, url, email, datetime, date, month, week, time, datetime-local, number, range, color.

Éléments : output, datalist, **optgroup**, keygen, progress, meter.

Syntaxes de modernizr

```
Modernizr.élément
Modernizr.inputtypes.type
Modernizr.input.attribut
```

Avec comme exemples de valeurs :

Modernizr.applicationcache, Modernizr.geolocation, Modernizr.localstorage, Modernizr.video, Modernizr.canvas, ...

Modernizr.inputtypes.date, Modernizr.inputtypes.time, ...

Modernizr.input.autofocus, Modernizr.input.required, ...

Si Modernizr ne reconnaît pas un type ou un attribut il est toujours possible de tester ainsi :

Pour un élément

```
var element = document.createElement("output");
if (element != "[object HTMLOutputElement]") {
    console.log("output inexistant !!!");
}
```

où vous adaptez l'HTML element.

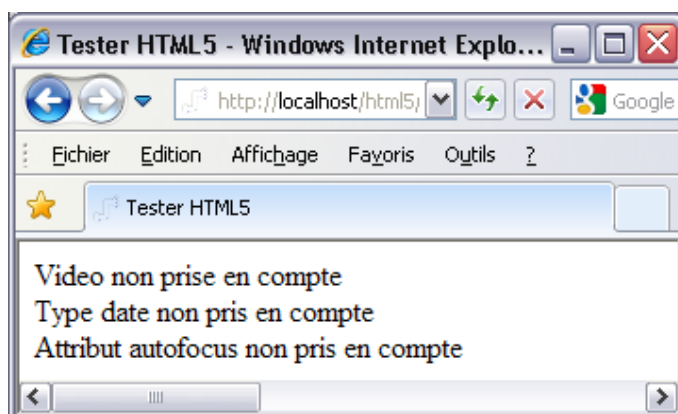
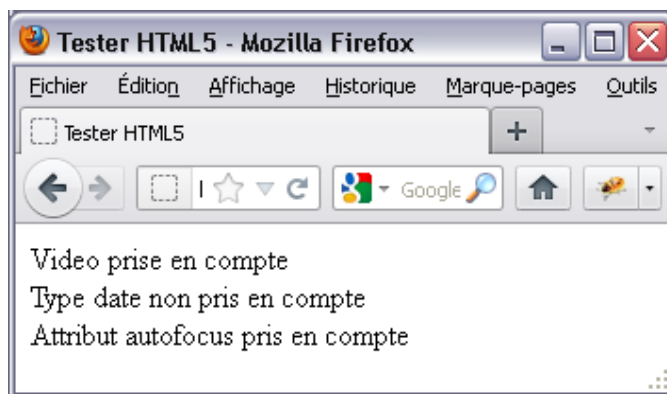
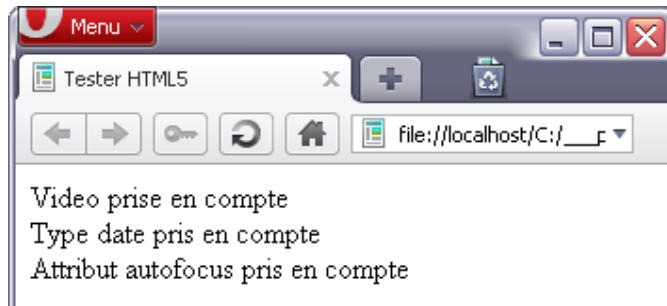
Pour un type d'input

Pour un attribut

```
if (!("min" in document.createElement("input"))) {
    console.log("L'attribut min est inexistant !!!");
}
```

Exemples de tests (Septembre 2012)

Le navigateur supporte-t-il les éléments video, date et l'attribut autofocus ?
(Opera 12.x, Firefox 15.0.x, Chrome 22.x et IE 8).



Code

```

<!DOCTYPE HTML>
<html lang="fr">
<!-- html5_tester.html -->
<head>
  <title>Tester HTML5</title>
  <meta charset="UTF-8" />
</head>

<body>
  <label id="lbl_msg"></label>

  <script src="../../../js/modernizr.js"></script>

  <script>
    // -----
    function testerHTML5() {
      var lblMsg = document.getElementById("lbl_msg");

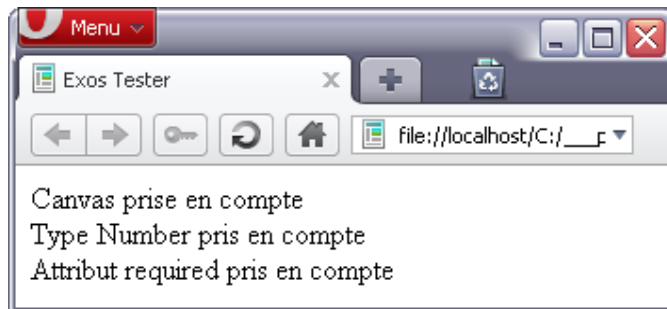
      // --- Test la prise en compte d'un element
      if (Modernizr.video) {
        lblMsg.innerHTML += "Video prise en compte<br/>";
      }
      else {
        lblMsg.innerHTML += "Video non prise en
compte<br/>";
      }
      // --- Test la prise en compte d'un type input
      if (Modernizr.inputtypes.date) {
        lblMsg.innerHTML += "Type date pris en
compte<br/>";
      }
      else {
        lblMsg.innerHTML += "Type date non pris en
compte<br/>";
      }
      // --- Test la prise en compte d'un attribut d'element
      if (Modernizr.input.autofocus) {
        lblMsg.innerHTML += "Attribut autofocus pris en
compte<br/>";
      }
      else {
        lblMsg.innerHTML += "Attribut autofocus non pris en
compte<br/>";
      }
    }

    window.onload = testerHTML5;
  </script>
</body>
</html>

```

Exercice

Vos navigateurs supportent-ils les canvas, les numbers et l'attribut required ?



1.17.2 - Pallier l'absence d'un attribut

Objectif

Pallier l'absence d'un attribut d'un élément lorsque le navigateur ne supporte pas les nouveautés de HTML5.

Démarche

Tester le support de l'attribut.
Modifier les attributs de l'élément si besoin est ou solliciter une méthode remplaçante.

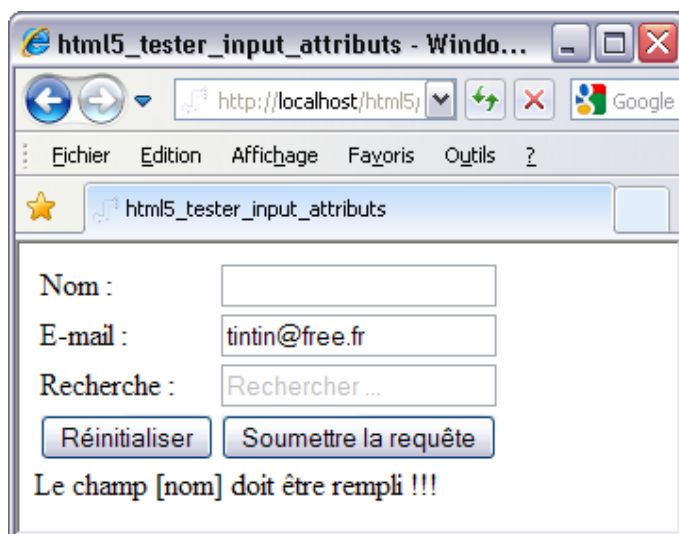
Exemple

Les attributs `required`, `autofocus` et `placeholder` (texte grisé à l'affichage du formulaire et effacé dès que l'élément prend le focus) sont testés.

Si l'attribut `required` n'existe pas on ajoute un code qui teste la saisie dans le champ en question.

Si l'attribut `autofocus` n'existe pas on utilise la méthode `focus()`.

Si l'attribut `placeholder` n'existe pas on modifie les attributs `value`, `style` et ajoute un code pour l'événement `onfocus`.



Code

```

<!DOCTYPE HTML>
<html lang="fr">
<!-- html5_tester_input_attributs.html -->
<!--
L'objectif est de tester certains attributs.
Quand il n'existe pas on ajoute un script pour l'element
-->
<head>
  <title>html5_tester_input_attributs</title>
  <meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1" />
  <style type="text/css">
    .gris{color: silver;}
    .noir{color: black;}
  </style>
</head>

<body>
  <form action="" method="get" onsubmit="return soumettre();">
    <table>
      <tr>
        <td>Nom : </td>
        <td><input type="text" name="nom" id="nom" value="Tintin"
required="required" /></td>
      </tr>
      <tr>
        <td>E-mail : </td>
        <td><input type="email" name="e_mail" id="e_mail"
value="tintin@free.fr" autofocus="autofocus" /></td>
      </tr>
      <tr>
        <td>Recherche : </td>
        <td><input type="search" name="recherche" id="recherche"
placeholder="Saisissez ..." /></td>
      </tr>
      <tr>
        <td><input type="reset"/></td>
        <td><input type="submit"/></td>
      </tr>
    </table>
    </form>

    <label id="lbl_msg"></label>

    <script src="../js/modernizr.js"></script>

    <script>
      var lblMsg;
      var obligatoire = false;

```

```

// -----
function init() {
    lblMsg = document.getElementById("lbl_msg");
    // --- Si la propriete required n'existe pas pour un
    element de type input
    //if (!("required" in document.createElement("input")))
{
    if (!Modernizr.input.required) {
        obligatoire = true;
        lblMsg.innerHTML += "required non reconnu<br/>";
    }
    else {
        lblMsg.innerHTML += "required reconnu<br/>";
    }

    // --- Si la propriete autofocus n'existe pas pour un
    element de type input
    //if (!("autofocus" in
document.createElement("input"))) {
        if (!Modernizr.input.autofocus) {
            document.getElementById("e_mail").focus();
            lblMsg.innerHTML += "autofocus non reconnu<br/>";
        }
        else {
            lblMsg.innerHTML += "autofocus reconnu<br/>";
        }

        // --- Si la propriete placeholder n'existe pas pour un
        element de type input
        //if (!("placeholder" in
document.createElement("input"))) {
            if (!Modernizr.input.placeholder) {
                document.getElementById("recherche").value =
"Rechercher ...";

document.getElementById("recherche").setAttribute("class", "gris");
// --- OK

document.getElementById("recherche").setAttribute("onfocus",
"effacerTexte();"); // --- OK
                lblMsg.innerHTML += "placeholder non reconnu<br/>";
            }
            else {
                lblMsg.innerHTML += "placeholder reconnu<br/>";
            }
        } // init
    }

// -----
function soumettre() {
    if(obligatoire) {
        if(document.getElementById("nom").value == "") {
            lblMsg.innerHTML += "Le champ [nom] doit être
rempli !!!<br/>";
            return false;
        }
    }
}

```



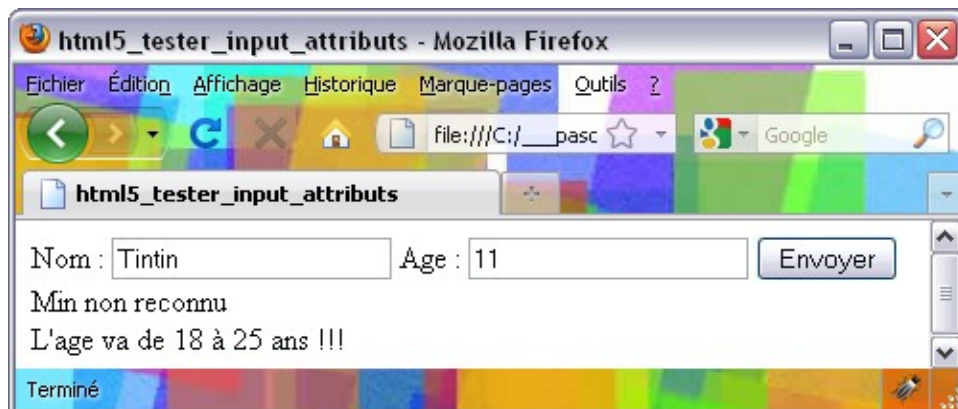
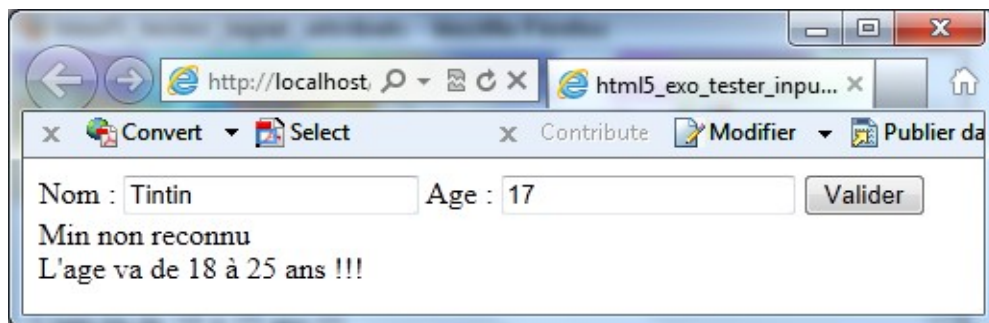
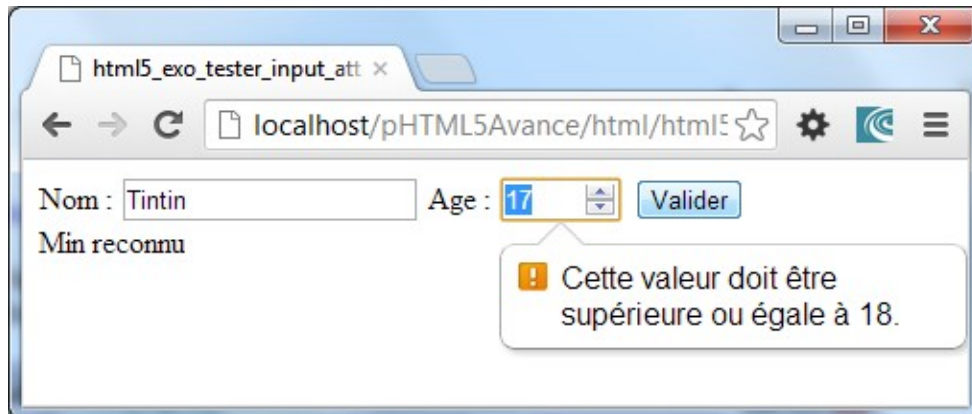
```
        } /// soumettre

        // -----
        function effacerTexte() {
            document.getElementById("recherche").value = "";
document.getElementById("recherche").setAttribute("class", "noir");
        } /// effacerTexte

        // --- INIT
        window.onload = init;
    </script>
</body>
</html>
```

Exercice

Testez la reconnaissance de l'attribut min pour la saisie de l'âge d'une personne. L'âge doit être compris entre 18 et 25 ans.



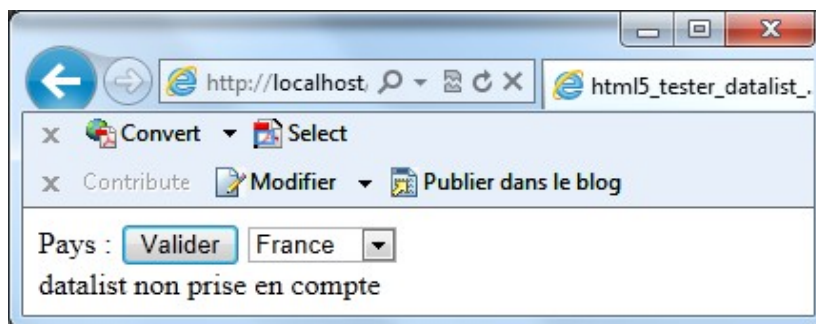
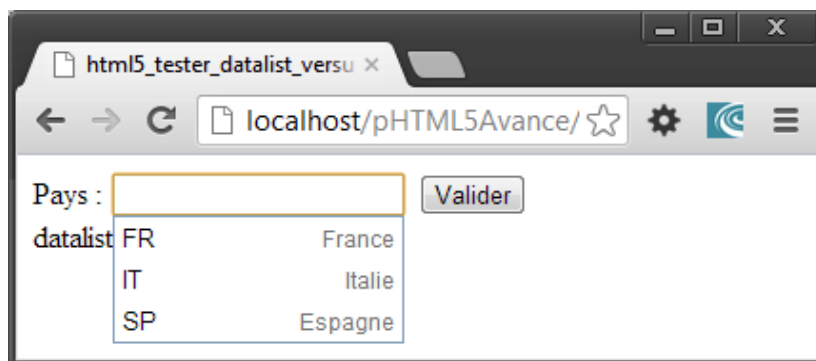
1.17.3 - Pallier l'absence d'un élément

Principe

Si l'élément n'est pas reconnu par le navigateur, il faut le remplacer par un autre. Soit par un élément HTML basique, soit par un élément plus sophistiqué disponible dans un framework ou une bibliothèque JavaScript (Dojo, jQuery, ...).

Exemple basique avec datalist

Remplacer un `<input type="text" />` qui utilise une `<datalist>` par une balise `<select>`.



Modernizr.datalist non reconnu (version 2.5.2).

```

<!DOCTYPE HTML>
<html lang="fr">
<!-- html5_tester_datalist_versus_select.html -->
<!--
L'objectif est de tester datalist.
Quand il n'existe pas on ajoute l'element <select> par script
et on supprime l'input type text
-->
<head>
  <title>html5_tester_datalist_versus_select</title>
  <meta charset="UTF-8" />
</head>

<body>
  <form action="" method="get" id="formulaire">
    <label>Pays : </label>
    <input type="text" name="pays" id="itPays"
list="liste_pays" value="" />

    <datalist id="liste_pays">
      <option label="France" value="FR">
      <option label="Italie" value="IT">
      <option label="Espagne" value="SP">
    </datalist>

    <input type="submit" value="Valider" />
  </form>

  <label id="lbl_msg"></label>

  <script>
    // -----
    function testerHTML5() {
      var formulaire = document.getElementById("formulaire");
      var lblMsg = document.getElementById("lbl_msg");

      // Le test sans Modernizr
      // On tente d'ajouter un element datalist
      var dl = document.createElement("datalist");

      // Si l'élément ajouté n'est pas un objet datalist
      if(dl !== "[object HTMLDataListElement]") {
        lblMsg.innerHTML = "datalist non prise en compte";
        formulaire.removeChild(itPays);

        var liste = document.createElement("select");

        var op = document.createElement("option");
        op.setAttribute("value", "FR");
        op.innerHTML = "France";

```

```
        liste.appendChild(op);

        var op = document.createElement("option");
        op.setAttribute("value", "IT");
        op.innerHTML = "Italie";
        liste.appendChild(op);

        var op = document.createElement("option");
        op.setAttribute("value", "SP");
        op.innerHTML = "Espagne";
        liste.appendChild(op);

        liste.setAttribute("id", "pays");
        liste.setAttribute("name", "pays");
        formulaire.appendChild(liste);
    }
    else {
        lblMsg.innerHTML = "datalist pris en compte";
    }
} /// testerHTML5

// --- INIT
window.onload = testerHTML5;
</script>
</body>
</html>
```

Exercices

1) Gérez l'absence du type date (qui affiche un calendrier).

IE8

Date de livraison :

Input type date not supported

Août 2010						
L	M	M	J	V	S	D
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Chrome 20.x

Date de livraison :

Input type date pris en charge

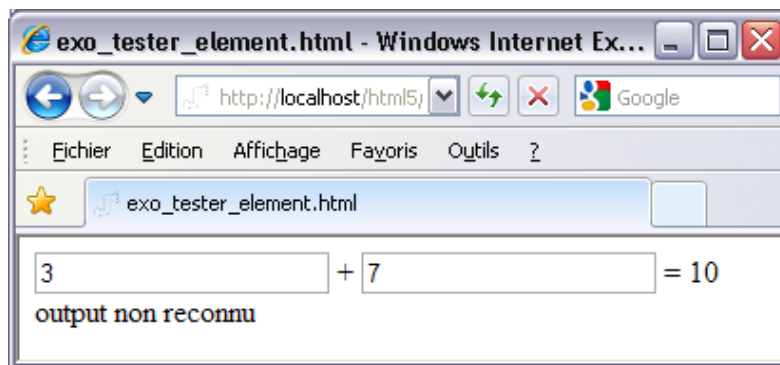
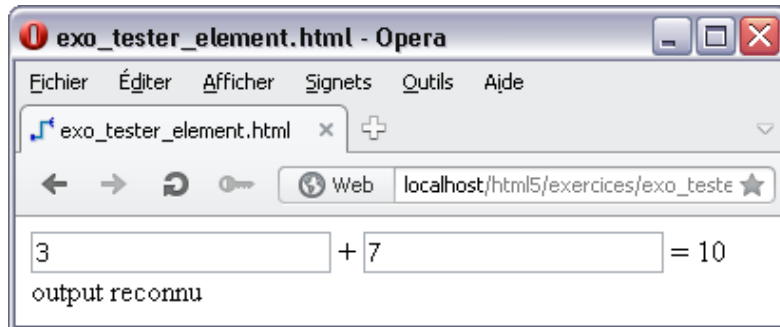
août 2010 ▼						
lun.	mar.	mer.	jeu.	ven.	sam.	dim.
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

2) Remplacez l'élément `<output>` si nécessaire.

modernizr.js (version `< 2.x`) ne reconnaît pas `output`.

Opera : l'élément `output` existe.

IE8 : l'élément `output` n'existe pas.





1.18 - CANVAS

1.18.1 - Définition

L'élément `<canvas>` est utilisé pour créer des éléments 2D et 3D à la volée avec du JavaScript.

Cet élément est proche de l'élément `` par certains côtés mais ne possède que les attributs `width` et `height`.

Pour les dernières versions les principaux navigateurs supporte cet élément.

			
X	X	X	X

Syntaxe HTML

```
<canvas id="canevas" width="largeur" height="hauteur">  
Message en cas d'erreur  
</canvas>
```

Il est préférable d'identifier l'élément canvas.

Le dimensionnement peut être spécifié en HTML et en CSS (sauf qu'en CSS ça bogue avec Chrome) ! Par défaut l'élément `<canvas>` est de 300 px sur 150px. Ensuite il faut coder en JavaScript.

Contenu alternatif : pour les navigateurs qui n'implémentent pas l'élément `<canvas>` il faut fournir un contenu alternatif. Il suffit de placer un texte ou même une image entre la balise ouvrante et la balise fermante.

Pour pouvoir dessiner il faut préciser le contexte (actuellement 2D) grâce à la méthode `getContext("2d")` de l'objet `<canvas>`.

Références :

https://developer.mozilla.org/fr/Tutoriel_canvas

<http://dev.opera.com/articles/view/html-5-canvas-the-basics/>

<http://caimansys.com/painter/> : un exemple d'interface permettant de créer des dessins.

<http://www.benjoffe.com/code/demos/canvascape/> : un exemple de jeu.

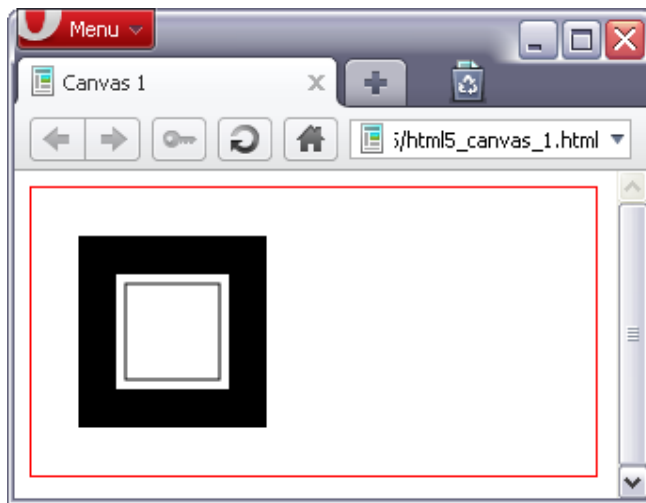
1.18.2 - Quelques méthodes

Méthode	Fonctionnalité
Contexte = canvas.getContext('2d')	Crée un contexte qui permet de dessiner.
Context.beginPath()	Ouvre le chemin pour dessiner.
Context.closePath()	Ferme le chemin.
Context.fillRect()	Dessine un rectangle plein.
Context.strokeRect()	Dessine un rectangle vide.
Context.clearRect()	Efface un rectangle.
Context.moveTo()	Déplace le pointeur.
Context.lineTo()	Trace une ligne.
Context.arc()	Dessine un arc.
Context.fillText()	Ecrit un texte.
Context.drawImage()	Dessine une image.

1.18.3 - Quelques attributs

Attribut	Description
Context.strokeStyle	Définit la couleur d'une ligne ou d'une bordure. La couleur peut être définie par un nom, une valeur hexadécimale, la fonction rgb(R,G,B) ou encore la fonction rgba(R,G,B, transparence).
Context.fillStyle	Définit la couleur de remplissage.
Context.lineWidth	Définit l'épaisseur d'une ligne.
Context.lineJoin	" miter " (pointu), "round" (arrondi), "bevel" (droit) . Permet de créer une pointe de flèche par exemple.
Canvas.width	Récupère la largeur du canevas.
Canvas.height	Récupère la hauteur du canevas.
Canvas.font	La police de caractères.

1.18.4 - Un rectangle avec les couleurs par défaut



La seule forme primitive de canvas est le rectangle.

Syntaxes

Les coordonnées 0,0 sont en haut à gauche.

Création du contexte pour dessiner via la récupération du contexte 2D; cela correspond à l'API utilisée.

```
var contexte = canvas.getContext('2d');
```

Dessin d'un rectangle plein

```
contexte.fillRect(x,y,largeur,hauteur);
```

Dessin d'un rectangle

```
contexte.strokeRect(x,y,largeur,hauteur);
```

Effacement d'un rectangle

```
contexte.clearRect(x,y,largeur,hauteur);
```

```
<!DOCTYPE HTML>

<html>
<!-- html5_canvas_1.html -->
<!-- Rectangles, couleurs de base -->

<head>
  <title>Canvas 1</title>

  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />

  <script>
    // -----
    function dessiner() {
      var canvas = document.getElementById('canevas');

      if(canvas.getContext) {
        var contexte = canvas.getContext('2d');

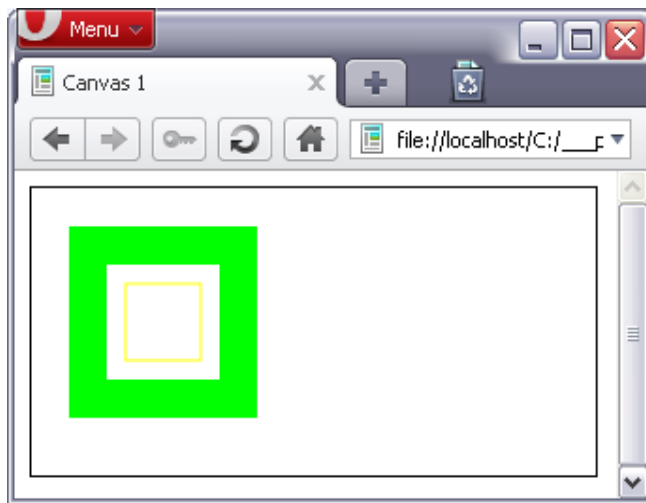
        // --- xxxRect(x,y,largeur,hauteur)
        contexte.fillRect(20,20,100,100); // --- Dessine un
rectangle plein noir
        contexte.clearRect(40,40,60,60); // --- Efface un
rectangle
        contexte.strokeRect(50,50,40,40); // --- Dessine un
rectangle
      }
    }
    windows.onload = dessiner;
  </script>

  <style type="text/css">
    canvas{ border: 1px solid red; }
  </style>
</head>

<body>
  <canvas id="canevas" width="300" height="150">
    Votre navigateur ne supporte pas l'élément canvas
  </canvas>
</body>

</html>
```

1.18.5 - Rectangles en couleurs



Syntaxes

Couleur du remplissage (Par défaut noir)

```
Contexte.fillStyle = couleur
```

Couleur du trait (Par défaut gris clair)

```
Contexte.strokeStyle = couleur
```

Largeur du trait (Par défaut 1px)

```
Contexte.lineWidth = pixels
```

Code

Placez ces lignes après avoir récupéré le contexte et avant de dessiner les rectangles

```
contexte.fillStyle    = '#0F0'; // Vert  
contexte.strokeStyle  = 'yellow'; // Jaune  
contexte.lineWidth    = 1;
```

1.18.6 - Lignes

Objectif et démarche



Canvas ne permet de créer qu'une forme primitive - le rectangle. Toutes les autres formes sont créées en combinant un ou plusieurs chemins.

Il faut au départ créer un chemin.

Ensuite il faut tracer une ligne d'un point A à un point B.

Et puis dessiner.

Et enfin fermer le chemin.

Syntaxes

Créer un chemin

```
| contexte.beginPath()
```

Fermer un chemin

```
| contexte.closePath()
```

Se positionner

```
| contexte.moveTo(x, y);
```

Définir une ligne

```
| contexte.lineTo(x, y);
```

Dessiner

```
| contexte.stroke()
```

Code

```

<!DOCTYPE html>
<html>
  <head>
    <title>html5_canvas_droite</title>
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <style type="text/css">
      #canevas{ border: 1px solid black;}
    </style>
  </head>

  <body>
    <div>
      <canvas id="canevas">
        Votre navigateur ne supporte pas l'élément canvas
      </canvas>
    </div>

    <script>
      /*
       * init
       */
      function init() {
        var canevas = document.getElementById("canevas");
        if (canevas.getContext) {
          var contexte = canevas.getContext('2d');
          dessinerDroite(contexte);
        }
      } /// init

      /*
       * dessinerDroite
       */
      function dessinerDroite(contexte) {
        contexte.beginPath();
        contexte.strokeStyle = 'black';
        contexte.lineWidth = 0.5;
        contexte.moveTo(20,100);
        contexte.lineTo(100,10);
        contexte.stroke();
        contexte.closePath();
      } /// dessinerDroite

      // -----
      window.onload = init;
    </script>

  </body>
</html>

```

1.18.7 - Points

Un point est pour canvas une ligne de 1,1 ou un rectangle (1,1) ou plus ... (2,2) par exemple.

```
<!DOCTYPE html>
<html>
  <head>
    <title>html5_canvas_point</title>

    <meta charset="UTF-8">
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />

    <style type="text/css">
      #canevas{ border: 1px solid black;}
    </style>
  </head>

  <body>
    <div>
      <canvas id="canevas">
        Votre navigateur ne supporte pas l'élément canvas
      </canvas>
    </div>

    <script>
      // -----
      function init() {
        var canevas = document.getElementById("canevas");
        if (canevas.getContext) {
          var contexte = canevas.getContext('2d');
          dessinerPoint(contexte);
        }
      } // init

      // -----
      function dessinerPoint(contexte) {
        contexte.strokeStyle = 'red';
        contexte.lineWidth = 1;

        contexte.beginPath();

        // --- Avec une ligne
        // x,y
        contexte.moveTo(20, 100);
        contexte.lineTo(21, 101);

        // --- Avec un rectangle
        contexte.fillRect(50,50, 1,1);

        contexte.stroke();
        contexte.closePath();
      }
    </script>
  </body>
</html>
```



```
        } /// dessinerPoint  
  
        // -----  
        window.onload = init;  
    </script>  
  
</body>  
</html>
```

1.18.8 - Triangles

Objectif et démarche

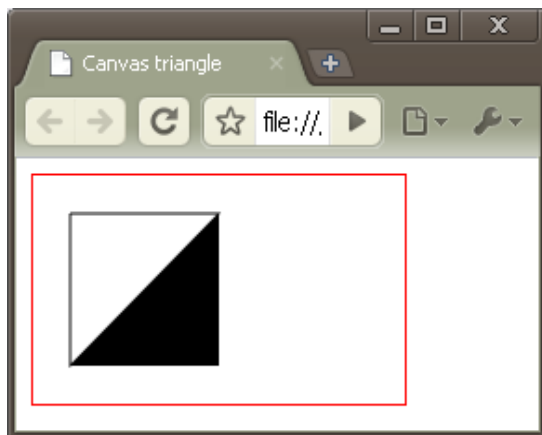
La figure triangle n'existe pas. Le triangle est créé à partir de lignes.

Il faut au départ créer un chemin.

Ensuite il faut tracer une ligne d'un point A à un point B, puis une autre du point B à un point C et enfin de C à A.

Puis dessiner.

Et enfin fermer le chemin.



Syntaxes

Comme pour les lignes.

Dessiner

```
| contexte.stroke() // --- Tracé de lignes
```

```
| contexte.fill() // --- Tracé plein
```

Code

```

<!DOCTYPE HTML>

<html>

<!-- html5_canvas_triangle.html -->
<head>
    <title>Canvas triangle</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

    <script>
        // -----
        function dessiner() {
            var canvas = document.getElementById('canevas');

            if(canvas.getContext) {
                var contexte = canvas.getContext('2d');

                // --- Cree un chemin
                contexte.beginPath();
                // --- Le premier triangle
                // --- moveTo(x,y)
                contexte.moveTo(20,20);
                // ---.lineTo(x,y) à partir du point actuel dans le
chemin
                // --- ie soit le point atteint avec moveTo()
                // --- soit le point de la fin de ligne précédemment
tracée

                contexte.lineTo(20,100); // AB
                contexte.lineTo(100,20); // BC
                contexte.lineTo(20,20); // CA
                contexte.stroke(); // --- Dessine un triangle
                contexte.closePath();

                // --- Le deuxième triangle
                contexte.beginPath();
                contexte.moveTo(100,20);
                contexte.lineTo(100,100);
                contexte.lineTo(20,100);
                contexte.fill(); // --- Dessine un triangle plein
                contexte.closePath();
            }
        }
        windows.onload = dessiner;
    </script>
</head>

<body>

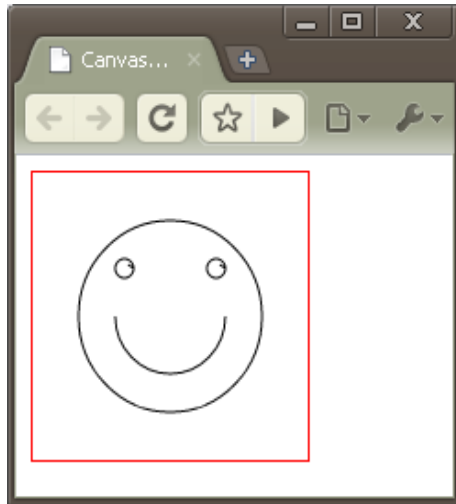
```

```
<canvas id="canevas">
  Votre navigateur ne supporte pas l'élément canvas
</canvas>
</body>
</html>
```

1.18.9 - Arcs et cercles

Objectif et démarche

Pour dessiner un arc ou un cercle on utilise une seule fonction : la fonction `arc()`.
Un cercle est un arc fermé.
Là aussi, il faut au préalable créer un chemin ...



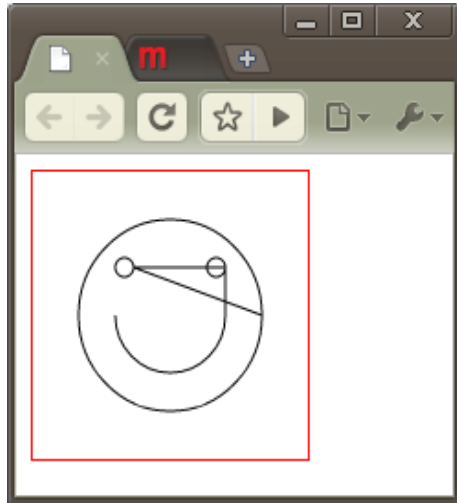
Syntaxe

```
arc(int centreX, int centreY, int rayon, float angleDebut, float  
angleFin, booléen sensInverseDesAiguilles)
```

Le point de départ d'un arc est `centreX + rayon` pour `x` et `y` pour `y`.
Pour un cercle le paramètre `sensInverseDesAiguilles` est indifférent.
Pour un arc cela a en revanche de l'importance.
Rappel : le périmètre d'un cercle est $R * \pi * 2$.
Et le paramètre `angleFin` est à calculer en fonction du périmètre à parcourir. Par exemple $\pi * 2$ pour dessiner un cercle et π pour dessiner un demi-cercle ...

Remarques

Si vous enlevez les `moveTo(x,y)` vous obtiendrez ceci :



parce qu'automatiquement une ligne sera tracée entre le point de départ de l'arc n et le point de départ de l'arc $n+1$. C'est pour ce motif qu'il faut se déplacer au départ (à 3h15) ou sur n'importe quel point de l'arc nouveau.

Code

```

<!DOCTYPE HTML>

<html>
<!-- html5_canvas_cercles.html -->

<head>
  <title>Canvas cercles</title>

  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />

  <script>
    // -----
    function dessiner() {
      var canvas = document.getElementById('canevas');

      if(canvas.getContext) {
        var contexte = canvas.getContext('2d');

        contexte.beginPath();
        // --- arc(centre x, centre y, rayon, angle debut,
angle fin, sens inverse des aiguilles)
        contexte.arc(75,75,50,0,Math.PI*2,true); // ---
Visage

        // --- On se deplace sur la courbe, au point de
depart
        // --- CentreX + rayon, centreY
        contexte.moveTo(55,50);
        contexte.arc(50,50,5,0,Math.PI*2,true); // --- Oeil
gauche

        contexte.moveTo(105,50); // -- On se deplace sur la
courbe
        contexte.arc(100,50,5,0,Math.PI*2,true); // --- Oeil
droit

        contexte.moveTo(105,75); // -- On se deplace sur la
courbe
        contexte.arc(75,75,30,0,Math.PI,false); // ---
Bouche

        contexte.stroke(); // --- Dessine une ou plusieurs
figures
        contexte.closePath();
      }
    }
    windows.onload = dessiner;
  </script>

  <style type="text/css">

```

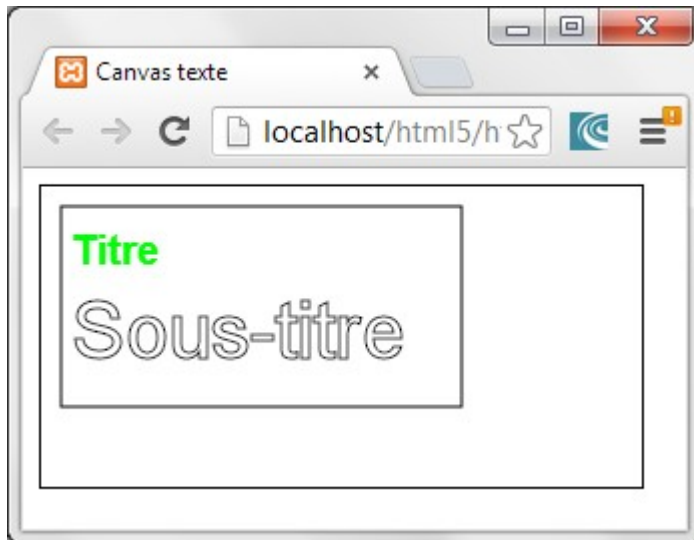
```
        canvas{ border: 1px solid red; }
    </style>
</head>

<body>
    <canvas id="canevas" width="150" height="150">
        Votre navigateur ne supporte pas l'élément canvas
    </canvas>
</body>
</html>
```


1.18.10 - Ecrire du texte

Objectif

Ecrire du texte.



Syntaxes

Ecriture du texte

```
contexte.fillText('Texte', x, y) ; // Texte plein
```

```
contexte.strokeText('Texte', x, y) ; // Texte creux
```

Choix de la police

```
contexte.font("graisse taille police")
```

Graisse : normal, bold, italic.

Taille : en pixels.

Police : nom d'une police.

Choix de la couleur

```
contexte.fillStyle = 'couleur'
```

```
contexte.strokeStyle = 'couleur'
```

Alignement horizontal

```
contexte.align = 'alignement'
```

Alignement : **start**, end, left, right, center.

Alignement vertical

```
contexte.textBaseline = 'alignement'
```

Alignement : top, hanging, middle, **alphabetic**, ideographic, bottom.

Code

```

<!DOCTYPE HTML>

<html>
<!-- html5_canvas_texte.html -->

<head>
  <title>Canvas texte</title>

  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />

  <style type="text/css">
    canvas{ border: 1px solid black; }
  </style>

  <script>
    // -----
    function dessiner() {
      var canvas = document.getElementById('canevas');

      if(canvas.getContext) {
        var contexte = canvas.getContext('2d');

        contexte.strokeRect(10, 10, 200, 100);

        contexte.fillStyle    = '#0F0';
        contexte.font          = 'bold 20px sans-serif';
        contexte.textBaseline  = 'top';
        contexte.fillText('Titre', 15, 20);

        contexte.strokeStyle  = '#000';
        contexte.font          = 'normal 40px sans-serif';
        contexte.strokeText('Sous-titre', 15, 50);
      }
    }
    windows.onload = dessiner;
  </script>
</head>

<body>
  <canvas id="canevas" width="300" height="150">
    Votre navigateur ne supporte pas l'élément canvas
  </canvas>
</body>

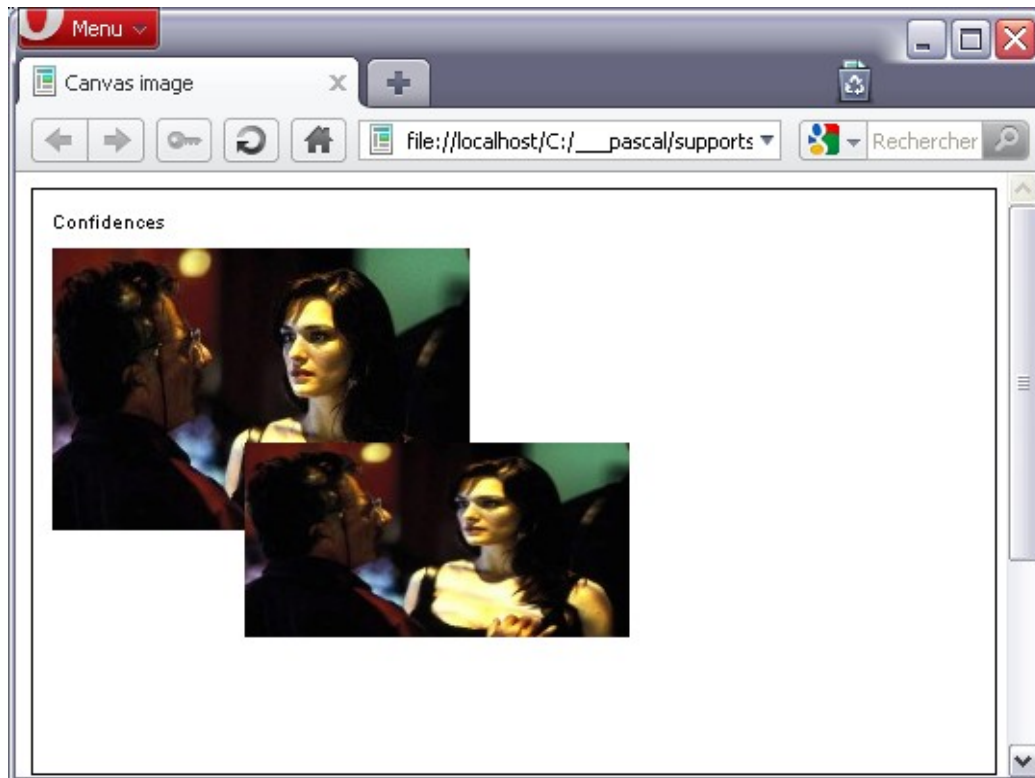
</html>

```

1.18.11 - Insérer une image

Objectif

Insérer une image dans un <canvas>.



Syntaxe

```
contexte.drawImage(idImage, x, y[, largeur, hauteur, ])
```

Code

```

<!DOCTYPE HTML>

<html>
<!-- html5_canvas_image.html -->

<head>
  <title>Canvas image</title>

  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />

  <style type="text/css">
    canvas{ border: 1px solid black; }
  </style>

  <script>
    // -----
    function dessiner() {
      var canvas = document.getElementById('canevas');

      if(canvas.getContext) {
        var contexte = canvas.getContext('2d');

        contexte.fillText('Confidences', 10, 20);
        contexte.drawImage(confidences, 10, 30); // ---
Affichage
        contexte.drawImage(confidences, 110, 130, 200, 100);
// --- Affichage et redimensionnement
      }
    }
    windows.onload = dessiner;
  </script>
</head>

<body>
  <canvas id="canevas" width="500" height="300">
    Votre navigateur ne supporte pas l'élément canvas
  </canvas>
  <br/>
  
</body>

</html>

```

1.18.12 - Animation : petite introduction

1.18.12.1 - Animation autonome : le temps

Objectif

Déplacer un cercle dans un rectangle (OK navigateur, OK smartphone).



Démarche

On affiche la balle avec un cercle de couleur.

On efface la balle avec un cercle blanc.

On utilise `setInterval()` et `clearInterval()` pour lancer et arrêter le mouvement.

Codes

```
<!DOCTYPE HTML>
<html>
  <!-- html5_canvas_animation_autonome.html -->
  <!-- Balle rebondissante -->
  <head>
    <title>Canvas animation autonome</title>

    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

    <style type="text/css">
      canvas{ border: 1px solid black; }
    </style>

    <script
src="../../js/html5_canvas_animation_autonome.js"></script>
  </head>

  <body>
    <canvas id="canevas" width="450" height="150">
      Votre navigateur ne supporte pas l'élément canvas
    </canvas>
    <br/>
    <input type="button" id="btArreter" value="Arrêter" />
    <input type="button" id="btRedemarrer" value="Redémarrer" /
  >
  </body>
</html>
```

```

/*
 * html5_canvas_animation_autonome.js
 */
var contexte;
var centreX = 10, centreY = 10, rayon = 10, angleDebut = 0,
angleFin = Math.PI * 2, sens = true;
var largeurMax, hauteurMax;
var pasX = 1, pasY = 1;
var intervalle;

/*
 *
 */
function initialisation() {
    document.getElementById('btArreter').onclick = arreter;
    document.getElementById('btRedemarrer').onclick = redemarrer;
    var canvas = document.getElementById('canevas');
    if (canvas.getContext) {
        contexte = canvas.getContext('2d');
        contexte.strokeStyle = "white";
        largeurMax = canvas.width;
        hauteurMax = canvas.height;
        intervalle = setInterval(dessiner, 10);
    }
}
/*
 *
 */
function arreter() {
    clearInterval(intervalle);
}
/*
 *
 */
function redemarrer() {
    intervalle = setInterval(dessiner, 10);
}
/*
 *
 */
function dessiner() {
    // --- Parametage de la couleur du pourtour
    contexte.strokeStyle = "white";

    // --- On efface la balle
    contexte.beginPath();
    contexte.fillStyle = 'white';
    // --- arc(centre x, centre y, rayon, angle debut, angle fin,
sens inverse des aiguilles)
    contexte.arc(centreX, centreY, rayon, angleDebut, angleFin,
sens); // --- Balle
    contexte.fill(); // --- Dessine un cercle (Remplissage)
    contexte.stroke(); // --- Dessine un cercle (Pourtour)
    contexte.closePath(); // --- Facultatif

```



```
// --- On affiche la balle
contexte.beginPath();
contexte.fillStyle = 'yellow';
centreX += pasX;
centreY += pasY;
contexte.arc(centreX, centreY, rayon, angleDebut, angleFin,
sens); // --- Balle
contexte.fill();
contexte.stroke();
contexte.closePath();

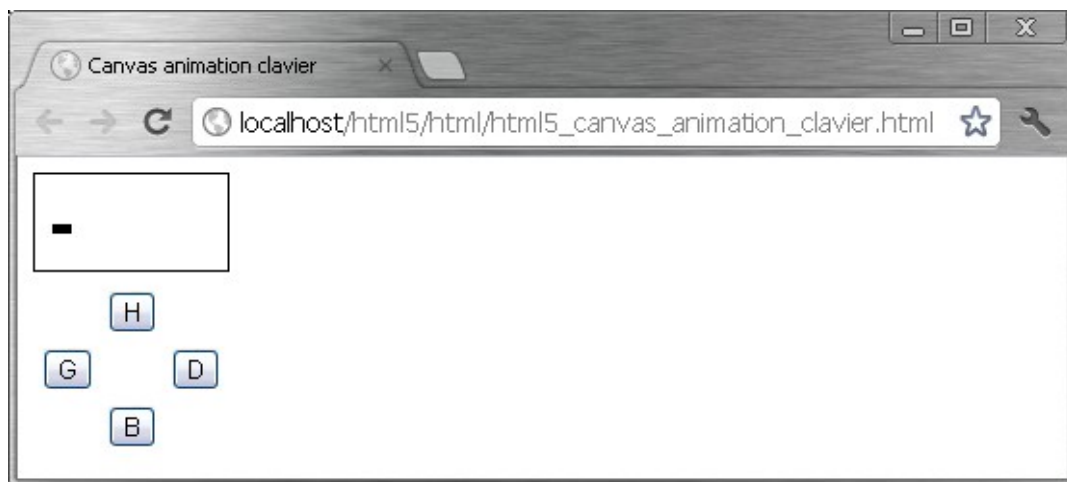
// --- Rebonds contre les parois
if (centreY === hauteurMax) {
    pasY = -pasY;
}
if (centreY === -5) {
    pasY = -pasY;
}
if (centreX === largeurMax) {
    pasX = -pasX;
}
if (centreX === -5) {
    pasX = -pasX;
}
}

/*
 *
 */
window.onload = initialisation;
```

1.18.12.2 - Animation guidée par l'utilisateur

Objectif

L'utilisateur déplace un petit rectangle au moyen des touches du clavier (Touches H,D,B,G et touches fléchées) ou des boutons (OK navigateur, KO **smartphone**).

**Démarche**

Le déplacement avec les boutons est réalisé au moyen d'une gestion événementielle souris classique (onclick).

Le déplacement avec les touches du clavier est réalisé au moyen d'une gestion événementielle clavier classique (onkeyup).

Les fonctions anonymes de gestion des clics sur les boutons appellent la fonction déplacer().

La fonction bouger(e) gère les frappes sur les touches du clavier. Elle appelle la fonction déplacer().

La fonction déplacer() modifie les valeurs de X et de Y et appelle les fonction effacer() et dessiner().

la fonction effacer() efface le petit rectangle.

La fonction dessiner() redessine le petit rectangle.

Code

```

<!DOCTYPE HTML>
<html>
  <!-- html5_canvas_animation_clavier.html -->
  <!-- Deplacement d'un rectangle avec des boutons ou les touches
du clavier -->
  <!-- OK Chrome !!! en 2014, KO Opera Mobile -->
  <head>
    <title>Canvas animation clavier</title>

    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="UTF-8">

    <style type="text/css">
      canvas{ border: 1px solid black; }
    </style>

    <script
src="../../js/html5_canvas_animation_clavier.js"></script>
  </head>

  <body>
    <canvas id="canevas" width="100" height="50">
      Votre navigateur ne supporte pas l'élément canvas
    </canvas>

    <table>
      <tr>
        <td>&nbsp;</td>
        <td><input type="button" id="btHaut" value="H" /></td>
      </tr>
      <tr>
        <td><input type="button" id="btGauche" value="G" /></td>
        <td>&nbsp;</td>
      </tr>
      <tr>
        <td><input type="button" id="btDroite" value="D" /></td>
        <td>&nbsp;</td>
      </tr>
      <tr>
        <td><input type="button" id="btBas" value="B" /></td>
        <td>&nbsp;</td>
      </tr>
    </table>
  </body>
</html>

```

```

/*
 * html5_canvas_animation_clavier.js
 */
var canvas;
var contexte;
var x = 0, y = 10;
var largeur = 10, hauteur = 5;
var largeurMax, hauteurMax;

/*
 *
 * @returns {initialisation}
 */
function initialisation() {
    // OK FX, IE, Safari et Opera
    // KO Chrome
    // KO Mobiles

    /*
     * LE CLAVIER
     */
    document.onkeyup = bouger;

    /*
     * LES CLICS
     */
    document.getElementById("btHaut").onclick = function() {
        deplacer('H');
    };
    document.getElementById("btDroite").onclick = function() {
        deplacer('D');
    };
    document.getElementById("btBas").onclick = function() {
        deplacer('B');
    };
    document.getElementById("btGauche").onclick = function() {
        deplacer('G');
    };

    canvas = document.getElementById('canevas');
    if (canvas.getContext) {
        contexte = canvas.getContext('2d');
        largeurMax = canvas.width;
        hauteurMax = canvas.height;
        dessiner();
    }
}
/*
 *
 */
function deplacer(asDirection) {
    effacer();

    if (asDirection === "H" && y > 0) {
        y--; // haut
    }
}

```

```

    }
    if (asDirection === "D" && x < largeurMax - 10) {
        x++; // droite
    }
    if (asDirection === "B" && y < hauteurMax) {
        y++; // bas
    }
    if (asDirection === "G" && x > 0) {
        x--; // gauche
    }

    dessiner();
}
/*
 *
 */
function dessiner() {
    contexte.fillRect(x, y, largeur, hauteur);
}
/*
 *
 */
function effacer() {
    contexte.clearRect(x, y, largeur, hauteur);
}
/*
 *
 */
function bouger(e) {
    var oEvent = window.event || e;

    switch (oEvent.which) {
        case 38:
            deplacer('H'); // --- Fleche haut
            break;
        case 39:
            deplacer('D'); // --- Fleche droite
            break;
        case 40:
            deplacer('B'); // --- Fleche bas
            break;
        case 37:
            deplacer('G'); // --- Fleche gauche
            break;

        case 72:
            deplacer('H'); // --- Touche H
            break;
        case 68:
            deplacer('D'); // --- Touche D
            break;
        case 66:
            deplacer('B'); // --- Touche B
            break;
        case 71:
            deplacer('G'); // --- Touche G
    }
}

```

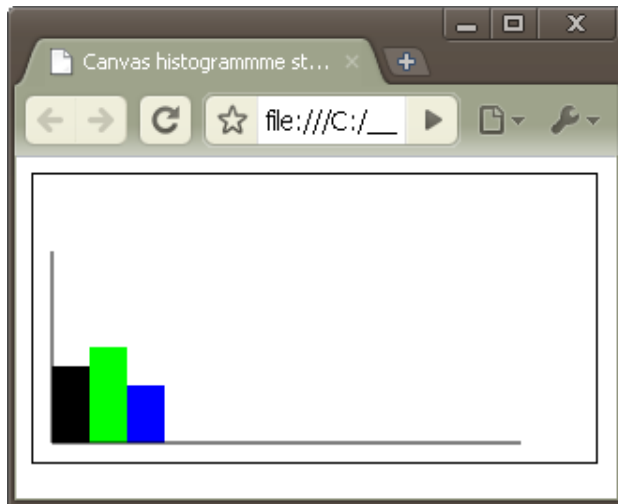
```
        break;
    default:
        //default code block
    }
}

/*
 *
 */
window.onload = initialisation;
```

1.18.13 - Un histogramme statique

Objectif

Dessiner un histogramme basique avec des axes x et y et trois bâtons.
Les données sont statiques. Les couleurs aussi.



Démarche

Définir les coordonnées du point (0,0), les longueurs des axes, la largeur des bâtons.

Dessiner les axes,

Dessiner les bâtons.

Code

```

<!DOCTYPE HTML>

<html>
<!-- html5_canvas_histogramme_statique.html -->
<!-- Une couleur par baton -->
<head>
    <title>Canvas histogramme statique</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

    <script>
        // -----
        function dessiner() {
            var canvas = document.getElementById('canevas');

            if(canvas.getContext) {
                var contexte = canvas.getContext('2d');

                // --- Points 0,0 de x,y
                var x = 10, y = 140;
                // --- Largeur du baton, hauteur du baton
                var largeur = 20, hauteur;
                // --- Longueur de l'axe des abscisses, longueur de
l'axe des ordonnees
                var maxX = 250, maxY = 100;

                contexte.beginPath();
                // --- Les axes : le point 0,0 est en 10,140
                contexte.moveTo(x, y);
                contexte.lineTo(x, y-maxY); // --- L'axe des
ordonnees
                contexte.moveTo(x, y);
                contexte.lineTo(x+maxX, y); // --- L'axe des
abscisses

                // --- Le point de depart en y sera y - hauteur
                // --- Dessine baton 1 en noir puisque fillStyle
n'est pas precise
                hauteur = 40;
                contexte.fillRect(x, y-hauteur, largeur, hauteur);

                // --- Dessine baton 2 en vert
                x += largeur; // --- "Deplacement" vers la droite
                hauteur = 50;
                contexte.fillStyle = "rgb(0,255,0)";
                contexte.fillRect(x, y-hauteur, largeur, hauteur);

                // --- Dessine baton 3 en bleu
                x += largeur;
                hauteur = 30;
            }
        }
    </script>

```



```
        contexte.fillStyle = "rgb(0,0,255)";
        contexte.fillRect(x, y-hauteur, largeur, hauteur);

        contexte.stroke();
        contexte.closePath();
    }
}
windows.onload = dessiner;
</script>
</head>

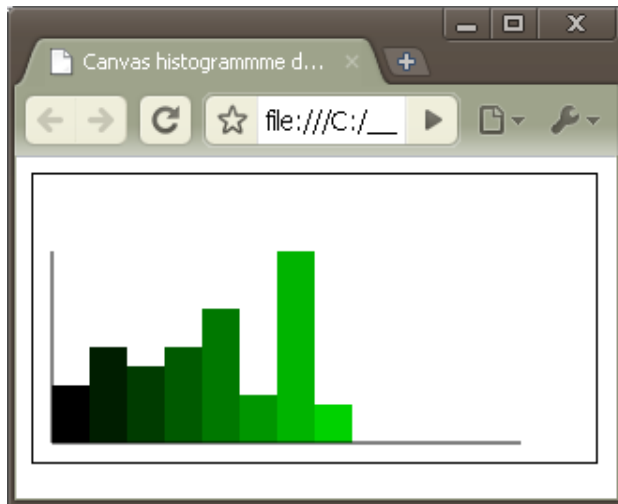
<body>
    <canvas id="canevas" width="300" height="150">
        Votre navigateur ne supporte pas l'élément canvas
    </canvas>
</body>

</html>
```

1.18.14 - Un histogramme dynamique

Objectif

Dessiner un histogramme basique avec des axes x et y et des bâtons.
Les données sont dynamiques. Les couleurs aussi.



Démarche

Les données sont stockées dans un tableau ... statique!

Une boucle permettra de générer les bâtons.

Les couleurs seront générées à partir de la fonction `rgb(rouge, vert, bleu)` qui est une chaîne de caractères. Il s'agit de faire une concaténation.

Code

```

<!DOCTYPE HTML>

<html>
<!-- html5_canvas_histogramme_dynamique.html -->
<head>
  <title>Canvas histogramme dynamique</title>
  <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
  <meta charset="utf-8" />
  <script>
    // -----
    function dessiner() {
      var canvas = document.getElementById('canevas');
      var tDonnees = new Array(30,50,40,50,70,25,100,20);
      var vert = 0;

      if(canvas.getContext) {
        var contexte = canvas.getContext('2d');

        // --- Points 0,0 de x,y
        var x = 10, y = 140;
        var largeur = 20, hauteur;
        var maxX = 250, maxY = 100;

        contexte.beginPath();
        // --- Les axes : le point 0,0 est en 10,140
        contexte.moveTo(x, y);
        contexte.lineTo(x, y-maxY); // --- L'axe des
ordonnees
        contexte.moveTo(x, y);
        contexte.lineTo(x+maxX, y); // --- L'axe des
abscisses

        // --- Le point de depart en y sera y - hauteur
        // --- Dessine les batons en fonction du tableau,
les couleurs varient
        for(var i=0; i<tDonnees.length; i++) {
          hauteur = tDonnees[i];
          contexte.fillStyle = "rgb(0," + vert + ",0)";
          contexte.fillRect(x, y-hauteur, largeur,
hauteur);

          x += largeur;
          vert += 30;
        }
        contexte.stroke();
        contexte.closePath();
      } /// if canvas
    } /// dessiner()
    windows.onload = dessiner;
  </script>

  <style type="text/css">

```

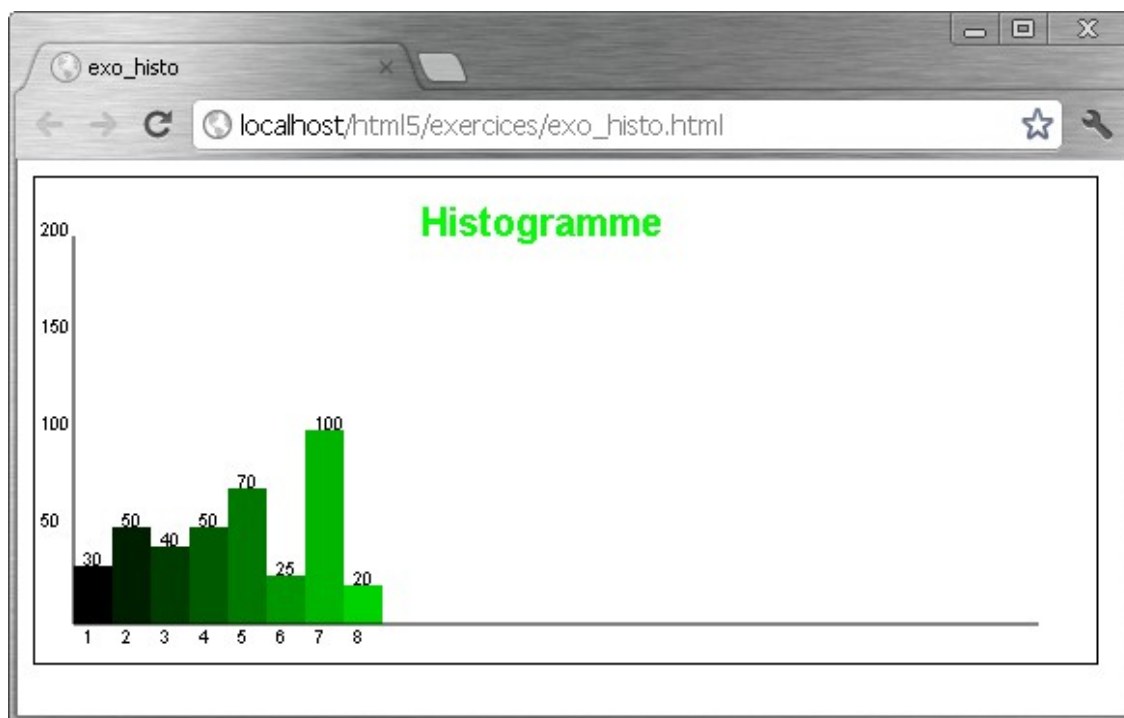
```
        canvas{ border: 1px solid black; }
    </style>
</head>

<body>
    <canvas id="canevas" width="300" height="150">
        Votre navigateur ne supporte pas l'élément canvas
    </canvas>
</body>

</html>
```

1.18.15 - Exercices au choix

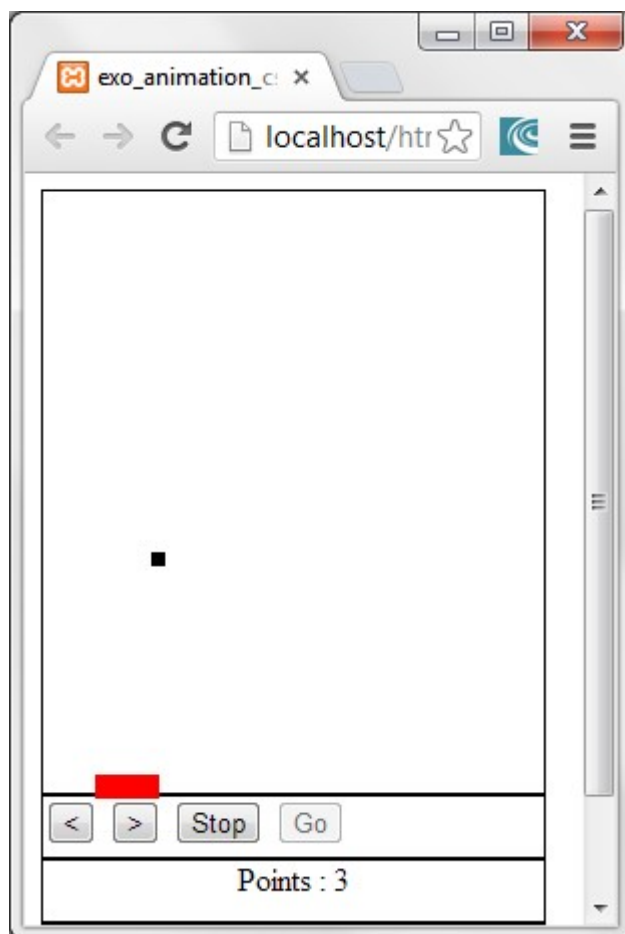
Reprenez l'histogramme précédent et affichez des légendes (du graphique et des axes) ainsi que les valeurs des Y.



Autres

Histo+ : le même mais avec des données XML (ou JSON).

Le petit jeu :



CHAPITRE 2 - ANNEXES

2.1 - LE SITE TP

Préférences : LocalStorage
Authentification : SessionStorage

Catalogue : OffLine
Panier : IndexedDB

Eventuellement IndexedDB pour la localisation des boutiques avec Google Maps derrière.

2.2 - LES BALISES HTML

En rouge les balises du HTML 4 non reconnues en HTML5.

En vert les nouvelles balises du HTML5.

En bleu les balises modifiées.

En violet les balises "limites".

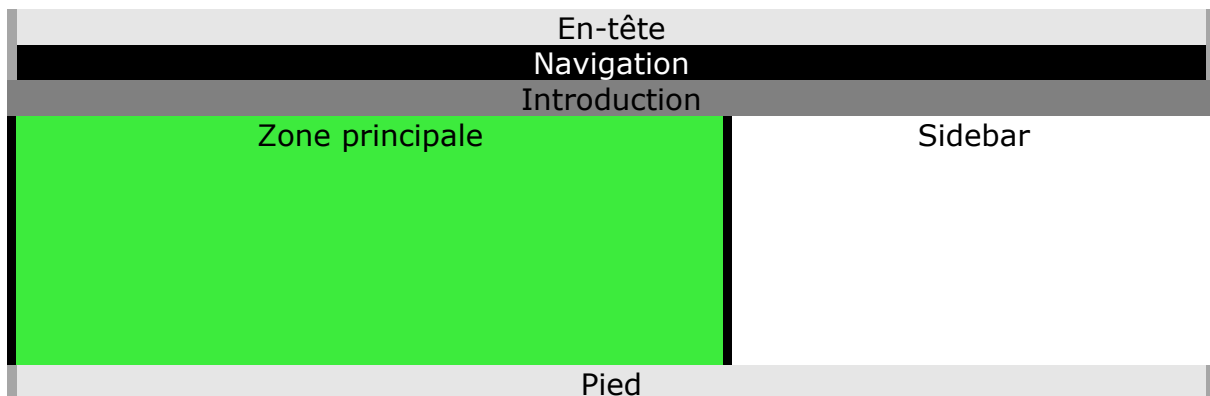
Pour les balises du HTML 4 cf le support html.doc.

Balise	Description
<!-->	Commentaire
<!DOCTYPE>	Le type de document. En HTML5 le DOCTYPE est une balise non fermée, avec une seule valeur HTML, et est insensible à la casse.
<a>	Ancre mais ...
<abbr>	Abréviation
<acronym>	Acronyme – Utilisez <abbr>
<address>	Adresse
<applet>	Applet – Utilisez <object>
<area>	Zone d'image
<article>	Article
<aside>	Élément tangent
<audio>	Contrôle audio
	Bold – Utilisez CSS
<base>	Définit, dans le <head> un lien par défaut pour tous les liens de la page
<basefont>	Police – Utilisez CSS
<bdo>	Bidirectional override – Utilisez pour prévenir d'une lecture inverse du texte (De la droite vers la gauche)
<big>	Big - Utilisez CSS
<blockquote>	Citation décalée vers la droite
<body>	Corps de requête http
 	Saut de ligne
<button>	Bouton
<canvas>	
<caption>	
<center>	Centré - Utilisez CSS
<cite>	Citation
<code>	Code
<col>	
<colgroup>	
<command>	
<datalist>	Liste de suggestions
<dd>	
	
<details>	
<dfn>	Définition. Rendu
<dir>	Liste – Utilisez
<div>	Élément block générique
<dl>	
<dt>	
	Utilisez plutôt du CSS.
<embed>	Élément incorporé tel un plugin (Flash, ...)
<fieldset>	
<figcaption>	
<figure>	Permet d'encadrer une image et de lui adjoindre une légende
	Police – Utilisez CSS

<footer>	Pied de page ou de section
<form>	Formulaire
<frame>	Cadre
<frameset>	Ensemble de cadres
<h1> - <h6>	Titre. Du plus large au plus petit.
<head>	En-tête HTTP
<header>	En-tête de page ou de section
<hgroup>	
<hr/>	Horizontal rule – Ligne horizontale.
<html>	No comment ☺
<i>	Italic - Utilisez CSS (l'a souvent remplacée)
<iframe>	Cadre interne
	Image
<input>	Zone de saisie
<ins>	
<keygen>	Générateur de clés
<kbd>	
<label>	Etiquette
<legend>	
	Élément de liste non numérotée
<link>	Lien
<map>	Pour le découpage d'images
<mark>	Mise en exergue
<menu>	
<meta>	Metadata du <head>
<meter>	
<nav>	Section de navigation
<noframes>	Utilisé dans le cadre des frames
<noscript>	Pour avertir les navigateurs ne supportant pas JavaScript
<object>	Pour l'incorporation d'objets (Applets, ...)
	Liste numérotée
<optgroup>	Regroupe des <option> au sein d'un <select>
<option>	Élément de liste déroulante – Cf la balise select
<output>	Champ de sortie pour un message
<p>	Paragraphe
<param>	Paramètre de la balise <object>
<pre>	Préserve les espaces et l'indentation et affiche le texte dans une police non proportionnelle
<progress>	
<q>	Citation : met le texte entre ""
<rp>	
<rt>	
<ruby>	
<s>	Strike – Barré – Utilisez CSS
<samp>	Exemple.
<script>	Script – cf JavaScript
<section>	Section, chapitre, ...
<select>	Liste déroulante ou liste. Élément de formulaire.
<small>	No comment ☺
<source>	
	Texte (Inline)
<strike>	Barré – Utilisez CSS
	Utilisez plutôt du CSS
<style>	Style – cf CSS
<sub>	"Imposant" ... le contraire de exposant
<summary>	Résumé de <table>
<sup>	Exposant
<table>	Table HTML
<tbody>	Corps de table
<td>	Table data – donnée d'une cellule de table
<textarea>	Zone de saisie multiligne

<tfoot>	Pied de table
<th>	Donnée de cellule de <thead>
<thead>	En-tête de table
<time>	Datation ... principalement d'un article
<title>	Titre du document. Dans le <head> c'est le seul élément visible.
<tr>	Table row – Ligne de table
<tt>	Télétype – Utilisez CSS
<u>	Underline – Utilisez CSS
	Liste non numérotée
<var>	Une variable
<video>	Contrôle video
<xmp>	Texte non proportionnel – Utilisez <pre>

2.3 - UNE PAGE PAS À PAS



2.3.1 - Phase 1 : La macro structure et les textes

```
<!DOCTYPE html>

<html>

  <head>
    <title>HTML5-Pas à pas-1</title>
    <meta charset="UTF-8">
  </head>

  <body>

    <header>
      <!-- En-tête de page - Header -->
      En-tête
    </header>

    <nav>
      <!-- Navigation -->
      Menu
    </nav>

    <section>
      <!-- Introduction -->
      Introduction
    </section>

    <section>
      <!-- Zone principale -->
      <article>
        <!-- Article -->
        Article
      </article>
      <aside>
        <!-- Sidebar - Encadré -->
        Encadré
      </aside>
    </section>

  </body>

</html>
```

```
        </aside>
    </section>

    <footer>
        <!-- Pied - Footer -->
        Pied
    </footer>

</body>
</html>
```

2.3.2 - Phase 2 : le style

```
<!DOCTYPE html>

<html>

  <head>
    <title>HTML5-Pas à pas-2</title>
    <meta charset="UTF-8">
    <style>
      *{margin: 0; padding: 0; color: white;}
      #entete{ background-color: silver;}
      #menu{ background-color: black;}
      #intro{ background-color: gray;}
      #principale{ background-color: white;}
      #article{ background-color: greenyellow; width: 70%;
min-height: 300px; float: left;}
      #aside{ background-color: yellow; width: 30%; min-
height: 300px;; float: left;}
      #pied{ background-color: silver; clear: both;}
    </style>
  </head>

  <body>

    <header id="entete">
      <!-- En-tête de page - Header -->
      En-tête
    </header>

    <nav id="menu">
      <!-- Navigation -->
      Menu
    </nav>

    <section id="intro">
      <!-- Introduction -->
      Introduction
    </section>

    <section id="principale">
      <!-- Zone principale -->
      <article id="article">
        <!-- Article -->
        Article
      </article>
      <aside id="aside">
        <!-- Sidebar - Encadré -->
        Encadré
      </aside>
    </section>

    <footer id="pied">
```

```
        <!-- Pied - Footer -->
        Pied
    </footer>

</body>

</html>
```

2.4 - LES ÉVÉNEMENTS HTML

Les événements en gras existaient en HTML 4.

Ce paragraphe nécessite une connaissance de JavaScript pour être intelligible et profitable.

2.4.1 - Les événements de niveau Page

Attribut	Valeur	Description
onafterprintNew	script	Après impression
onbeforeprintNew	script	Avant impression
onbeforeunloadNew	script	Avant le chargement
onblur	script	Sur perte de focus
onerrorNew	script	Sur erreur
onfocus	script	Sur prise de focus
onhaschangeNew	script	Sur changement de valeurs de document
onload	script	Sur chargement
onmessageNew	script	Sur message
onofflineNew	script	Sur offline
ononlineNew	script	Sur online
onpagehideNew	script	Sur masquage
onpageshowNew	script	Sur affichage
onpopstateNew	script	Sur changement de choix dans l'historique
onredoNew	script	Sur retour
onresizeNew	script	Sur redimensionnement
onstorageNew	script	Sur chargement
onundoNew	script	Sur annulation
onunloadNew	script	Sur déchargement

2.4.2 - Les événements de niveau formulaire

Attribut	Valeur	Description
onblur	script	Sur perte de focus
onchange	script	Sur changement de valeur
oncontextmenuNew	script	Sur événement de menu contextuel
onfocus	script	Sur prise de focus
onformchangeNew	script	Sur changement
onforminputNew	script	Sur saisie
oninputNew	script	Sur saisie
oninvalidNew	script	Sur saisie invalide
onselect	script	Lorsque élément est sélectionné
onsubmit	script	Sur soumission de formulaire

2.4.3 - Événements clavier

Utilisés pour des contrôles de saisie, ...

Attribut	Valeur	Description
onkeydown	script	Sur touche pressée
onkeypress	script	Sur touchée appuyée
onkeyup	script	Sur touche relâchée

2.4.4 - Événements souris

Pour les animations, l'interactivité, ...

Attribut	Valeur	Description
onclick	script	Sur un clic de souris
ondblclick	script	Sur double-clic de souris
ondragNew	script	Sur drag
ondragendNew	script	Sur fin de drag
ondragenterNew	script	Sur entrée dans un élément de dépôt
ondragleaveNew	script	Sur sortie d'élément de dépôt
ondragoverNew	script	Sur entrée dans un élément de dépôt
ondragstartNew	script	Sur drag
ondropNew	script	Sur drop
onmousedown	script	Sur bouton de souris appuyé
onmousemove	script	Sur mouvement de souris
onmouseout	script	Sur sortie de souris
onmouseover	script	Sur entrée de souris
onmouseup	script	Sur relâchement du bouton de souris
onmousewheelNew	script	Sur mouvement de molette
onscrollNew	script	Sur défilement

2.4.5 - Événements media

Tous ces événements sont nouveaux sauf le premier.
Ils sont activables sur les éléments audio, video, img, object et embed.

Attribut	Valeur	Description
onabort	script	Script to be run on an abort event
oncanplayNew	script	Script to be run when media can start play, but might has to stop for buffering
oncanplaythroughNew	script	Script to be run when media can be played to the end, without stopping for buffering
ondurationchangeNew	script	Script to be run when the length of the media is changed
onemptiedNew	script	Script to be run when a media resource element suddenly becomes empty (network errors, errors on load etc.)
onendedNew	script	Script to be run when media has reach the end
onerrorNew	script	Script to be run when an error occurs during the loading of an element
onloadeddataNew	script	Script to be run when media data is loaded
onloadedmetadataNew	script	Script to be run when the duration and other media data of a media element is loaded
onloadstartNew	script	Script to be run when the browser starts to load the media data
onpauseNew	script	Script to be run when media data is paused
onplayNew	script	Script to be run when media data is going to start playing
onplayingNew	script	Script to be run when media data has start playing
onprogressNew	script	Script to be run when the browser is fetching the media data
onratechangeNew	script	Script to be run when the media data's playing rate has changed
onreadystatechangeNew	script	Script to be run when the ready-state changes
onseekedNew	script	Script to be run when a media element's seeking attribute is no longer true, and the seeking has ended
onseekingNew	script	Script to be run when a media element's seeking attribute is true, and the seeking has begun
onstalledNew	script	Script to be run when there is an error in fetching media data (stalled)
onsuspendNew	script	Script to be run when the browser has been fetching media data, but stopped before the entire media file was fetched
ontimeupdateNew	script	Script to be run when media changes its playing position
onvolumechangeNew	script	Script to be run when media changes the volume, also when volume is set to "mute"
onwaitingNew	script	Script to be run when media has stopped playing, but is expected to resume

2.5 - DIVERS

2.5.1 - Redirection HTML

```
<meta http-equiv="refresh" content="5;  
URL=http://www.manouvelleadresse.com">
```

2.5.2 - Détecter la source de la requête : desktop ou mobile

<http://detectmobilebrowsers.com/>

2.5.3 - Le maquetage

Site d'exemples de maquettes (et fournit les codes HTML et CSS) de maquettes.

<http://freehtml5templates.com/>

2.5.4 - Logiciel de miniaturisation d'images

<http://www.fookes.com/ezthumbs/>

2.5.5 - Miniature cliquable

La page affiche l'image miniature (1_mini.png) et sur un clic l'image grand format est affichée (1.png).

```
<a href="../../../images/1.png">  
      
</a>
```


2.6 - LA GÉOLOCALISATION

Cf geolocalisation.odt








2.7 - HTML5 ET QUELQUES NAVIGATEURS LINUX (LES DERNIÈRES VERSIONS)

Tests réalisés en juillet 2015.

Input attributes : autocomplete, autofocus, list, placeholder, max, min, multiple, pattern, required, step.

Input Types : search, tel, url, email, datetime, date, month, week, time, datetime-local, number, range, color.

Éléments : output, datalist, **optgroup**, keygen, progress, meter.

	Chromium	Links	Links 2	Midori	QupZilla	Opera	Firefox	Epiphany	Konqueror
									
Required	X			X	X	X	X	X	
Placeholder	X			X	X	X	X	X	X
Min	X			X	X	X	X	X	
Max	X			X	X	X	X	X	
Multiple	X			X	X	X	X	X	
Pattern	X			X	X	X	X	X	
Step	X			X	X	X	X	X	
Autocomplete	X			X	X	X	X	X	
Autofocus	X			X	X	X	X	X	
List	X				X	X	X	X	
Output	X			X	X	X	X	X	
Keygen	X			X	X	X	X	X	
Datalist (*)	X				X	X	X	X	
Meter	X			X	X	X	X	X	
Progress	X			X	X	X	X	X	
Optgroup									

(*) avec List bien entendu.