PLB
FORMATION INFORMATIQUE ET MANAGEMENT

Formation HTML5 développeurs

Auteur : Cyrille Tuzi

Plan de la formation

1. Nouveautés sémantiques
2. Nouveautés pour les formulaires
3. Vidéo et audio
4. Canvas
5. Géolocalisation
6. Nouveautés AJAX
7. Historique et navigation
8. Stockage local
9. Communication serveur
10. WebRTC

Nouveautés sémantiques

Doctype et encodage

- Le seul doctype à utiliser :

```
<!DOCTYPE html>
```

- Le seul encodage à utiliser :

```
<meta charset="utf-8">
```

Nouvelles balises de structure

<header>

<footer>

<nav>

<aside>

<article>

<main>

Nouvelles balises de contenu

<figure>
<figcaption>
<time>

Sémantique de fond

- Micro-données :
  - itemscope
  - itemtype
  - itemprop
- RDFa Lite :
  - vocab
  - typeof
  - property

- Les types de données :
  http://schema.org/

Nouveautés pour les formulaires

Nouveaux champs

- Nouvelles valeurs pour le type des <input> :
  - email
  - url
  - tel
  - date
  - search
  - range

Nouveaux contrôles

- Contrôles du format via les types
- Nouveaux attributs :
  - required
  - maxlength
  - min / max / step
  - pattern (expressions régulières personnalisées)
- Désactivables avec l'attribut novalidate

RegExp : intervalles

[ach]	suite de caractères possibles
[a-z]	intervalle de caractères possibles
[a-zA-Z0-9_\-\.]	combinaisons
[^a]	tout sauf
(fr|com)	ou
.	n'importe quel caractère
\w	mot (word)
\d	chiffre (digit)
\s	espace (space)

RegExp : quantifieurs

{x,y}	minimum x fois et maximum y fois
{x}	x fois exactement
{x,}	minimum x fois
{,y}	maximum y fois
+	équivalent à {1,}
?	équivalent à {0,1}
*	équivalent à {0,}

Résultat

- Champ de résultat :

```
<output for="id"></output>
```

Nouveautés ergonomiques

- Adaptation des claviers tactiles en fonction du type de champ
- Texte préalable : attribut placeholder
- Attribut autofocus
- autocomplete="off"

Suggestions automatiques

```
<input type="search" list="search-list">

<datalist id="search-list">
    <option value="Suggestion 1">
    <option value="Suggestion 2">
</datalist>
```

Vidéo et audio

Balises et contrôles

- Nouvelles balises :

  <video>

  <audio>

- Barre de contrôle (lecture, pause, etc.) :

  <video controls>

Autres options

- Préchargement partiel :

  <video preload="metadata">

- Autres attributs :
  - autoplay
  - loop
  - muted
  - poster

Sources et formats

- Formats : cf support annexe

- Différentes sources :

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

- Config Apache (.htaccess) :

```
AddType video/webm .webm
```

Sous-titres

- Piste :

  <track kind="subtitles" src="en.vtt" srclang="en" label="English" default>

- Sous-titres:

  WEBVTT

  00:00:01.270 --> 00:00:05.300
  Hello world !

- Config Apache (.htaccess) :

  AddType text/vtt .vtt

API JavaScript

- Méthodes et propriétés :
  - play()
  - pause()
  - stop()
  - canPlayType()
  - currentTime
  - volume

- Nouveaux événements :
  - canplay / canplaythrough
  - play / playing / pause / ended
  - loadedmetadata / loadstart / progress / loadeddata
  - stalled / suspend / waiting
  - volumechange / timeupdate

Canvas

Canvas

- Nouvelle balise :
  - `<canvas width="500" height="200">`
- Création scriptée :
  - `getContext('2d')`

Dessiner

- Déplacer le pinceau :
  - beginPath()
  - moveTo()
  - closePath()
- Dessiner les formes :
  - stroke()
  - fill()
- Gommer :
  - clip()

Les formes

- Indirect :
  - lineTo()
  - arc()
  - rect()
- Direct :
  - strokeRect()
  - fillRect()
  - clearRect()

Textes et styles

- Texte :
  - fillText()
  - strokeText()

- Styles :
  - fillStyle
  - strokeStyle
  - lineWidth
  - lineJoin
  - font

Images

```
var image = new Image();

image.onload = function() {
    contexte.drawImage(image, x, y);
};

image.src = 'chemin.png';
```

Géolocalisation

API JavaScript

- Nouvel objet :
  - navigator.geolocation

- Nouvelles méthodes :
  - getCurrentPosition()
  - watchPosition()
  - clearWatch()

Options

- Ancienneté de la position :

  maximumAge: 0 / nombre de millisecondes / Infinity

- Temps d'attente maximum :

  timeout: nombre de millisecondes

- Meilleure précision :

  enableHighAccuracy: true

Erreurs et précision

- Erreurs possibles :
  - 1 : refus de l'utilisateur
  - 2 : indisponible
  - 3 : temps d'attente maximum dépassé

- Précision :
  - GPS ~ 5m
  - Wifi ~ 50m
  - Réseau mobile ~ 500m
  - Réseau filaire Plusieurs km

Position

- Asynchrone :

  function(position) {}

- Coordonnées GPS :
  - position.coords.latitude
  - position.coords.longitude

- Précision :
  - position.coords.accuracy (en mètres)

Nouveautés AJAX

XMLHttpRequest 2

- Nouvelle option pour XMLHttpRequest :
  - timeout (en millisecondes)
- Nouveaux événements
  - loadstart, progress
  - loadend, load (succès)
  - timeout, error, abort

Formats de réponse

- Nouvelles propriétés :
  - responseType
  - response

- Types de réponse :
  - text
  - document
  - json
  - blob / arraybuffer

Formulaires et fichiers

- Nouvel objet pour récupérer les champs d'un formulaire, fichiers joints compris :
  - new FormData()

- Suivi de la progression :
  - événement progress dans la propriété upload
  - event.lengthComputable
  - event.loaded
  - event.total
  - Balise : <progress value="0.5">

Historique et navigation

Historique avancé

- Nouvelles méthodes dans l'objet history :
  - pushState()
  - replaceState()

- Paramètres :
  - objet, stocké dans history.state
  - Titre
  - URL classique

- Nouvel événement :
  - popstate

Navigation interne

- En CSS3 :

```
#pages>div              { display: none; }
#pages>div:target       { display: block; }
```

- En JavaScript:
  - Ancre dans location.hash
  - Nouvel événement : hashchange

Glisser-déposer

Attributs et événements

- Nouveaux attributs :
  - draggable — auto | true | false
  - dropzone — copy | move

- Evénements :
  - dragstart / drag / dragend
  - dragenter / dragover / dragleave / drop

Données à transférer

- Données :
  - event.dataTransfer.setData('text', data);
  - event.dataTransfer.getData('text');

- Penser à annuler les comportements par défaut sur dragover et drop.

- Savoir si le dépôt a eu lieu :
  - event.dataTransfer.dropEffect

Fichiers externes

- Fichiers :
  - event.dataTransfer.files[i]
    - .type
    - .name
    - .size

FileReader API

```
var myFile = new FileReader();

myFile.addEventListener('loadend', function(event) {

    var img = document.createElement('img');
    img.src = event.target.result;

});

myFile.readAsDataURL(event.dataTransfer.files[0]);
```

Stockage local

Application hors ligne

- Côté HTML

```
<html manifest="manifest/myapp.appcache">
```

- Côté Apache

```
AddType text/cache-manifest .appcache
ExpiresActive On
ExpiresByType text/cache-manifest "access"
```

Manifest

```
CACHE MANIFEST

# Version 1.0.0

index.html
styles.css
script.js
images/logo.png
```

Mises à jour

window.applicationCache;

- Fonctionnalités :
  - update();
  - swapCache();

- Evénements :
  - checking
  - error / obsolete
    - downloading / progress
    - noupdate / updateready
    - cached

Etat de la connexion

- Etat actuel :
  - navigator.onLine;

- Evénements :
  - online
  - offline

Stockage local

- Deux nouveaux objets :
  - localStorage : global et permanent
  - sessionStorage :spécifique à l'onglet et temporaire

- Méthodes identiques :
  - getItem('index')
  - setItem('index', "valeur")
  - removeItem('index')
  - clear()

Données complexes

- Syntaxe JSON :

```
var myObject = {
    property1: 'valeur 1',
    property2: 'valeur 2'
};
```

- Possibilité de stocker des objets :
  - JSON.stringify()
  - JSON.parse()

Base de données locale

Présentation

- IndexedDB :
  - modèle objet
  - asynchrone
  - transactionnel

Connexion

```
var database;
var openRequest = indexedDB.open('nomdelabase', version);

openRequest.addEventListener('upgradeneeded', function(event) {
	database = event.target.result;
	// Création ou mise à jour de la base
});

openRequest.addEventListener('success', function(event) {
	database = event.target.result;
	// Traitements en cas de connexion réussie
});

database.close();
```

Création de "tables"

- Objet de stockage (table) avec clef primaire :

```
database.createObjectStore('table1', { keyPath: 'id'});
```

- Auto-incrément:

```
{ keyPath: 'id', autoIncrement: true }
```

- Index, avec option d'unicité :

```
table.createIndex('nom', 'colonne', { unique: true });
```

- Tables existantes :

```
database.ObjectStoreNames.contains('table1');
```

Transactions et requêtes

- Transaction :

```
database.transaction('table1', 'readonly').objectStore('table1');
database.transaction('table1', 'readwrite').objectStore('table1');
```

- Requêtes :
  - Insertion `table.add({champ1: "valeur1", champ2: "valeur2"});`
  - Update `table.put({champ1: "valeur1"}, 'valeur de la clef');`
  - Select `table.get('valeur de la clef');`
  - `table.index('nom index').get('valeur index');`
  - Delete `table.delete('valeur de la clef');`
  - Truncate `table.clear();`

```
requete.addEventListener('success', function(event) {
    event.target.result;
});
```

Communication serveur

Server-Sent Events

- Côté client :

```
var source = new EventSource('server.php');
source.addEventListener('message', function(event) {
    event.data;
}, false);
```

- Côté serveur :

```
header('Content-Type: text/event-stream; charset=utf-8');
```

data: Texte	Données envoyées à event.data
retry: 3000	Délai avant la relance (en ms)

Websockets

- Nouveau protocole :
  - ws://
  - wss://

- Nouvel objet :
  - `new WebSocket('ws://example.com/service/')`

API JavaScript

- Envoyer un message au serveur :
  - send('Message')
- Recevoir les messages du serveur :
  - événement message
  - donnée dans event.data
- Stopper la connexion :
  - close()

Côté serveur

- http://nodejs.org/

+

- https://github.com/Worlize/WebSocket-Node

WebRTC

WebRTC

```
navigator.getUserMedia(
    {video: true, audio: true},
    function (localMediaStream) {
        URL.createObjectURL(localMediaStream);
    },
    function () {}
);
```

Notifications

Permission

- A la suite d'une action utilisateur :

```
Notification.requestPermission(function (permission) {
    if ('granted' === permission) {
    }
});
```

Notification

```
new Notification("Hello world !", {
        body: "Message d'information",
        icon: 'chemin/image.png'
});
```

Templates

Templates

- Nouvelle balise HTML :

```
<template id="my-template"></template>
```

- Activation en JavaScript :

```
document.getElementById('post-title').content.
cloneNode(true);
```