

Fonctionnalités évoluées

Nouveaux éléments de périphériques

<audio>, <video> et <canvas>

Dans le monde du Web, malgré des capacités de stockage et d'échanges chaque jour améliorées, la situation semble bloquée et complexe : point de salut hors technologies propriétaires ou plugins tels que Adobe Flash ou Apple QuickTime.

HTML 5 change la donne en définissant trois nouveaux éléments destinés à gérer nativement les aspects visuels, sonores et graphiques de l'ère numérique moderne : <audio>, <video> et <canvas>.

<audio>

L'élément <audio> permet la lecture d'un fichier audio, aux formats classiques. Sa syntaxe est aussi courte qu'efficace :

```
<audio src="la-danse-des-canards.mp3"></audio>
```

Quelques attributs peuvent se greffer à la propriété <audio>. Il s'agit, entre autres, de autoplay (lecture automatique) et controls (affichage des boutons de contrôle de lecture).

Voici comment proposer sur votre page web un morceau musical lancé dès le chargement et muni de sa barre de contrôle :

```
<audio id="player" src="la-moldau.mp3" autoplay controls></ audio>
```

<audio>

À l'instar de la propriété font-family, l'élément <audio> offre la possibilité de proposer plusieurs types de formats au navigateur ; celui-ci va parcourir chacun des fichiers proposés et retenir le premier reconnu, sans charger les autres, comme le montre cet exemple :

```
<audio>  
  <source src="la-moldau.mp3">  
  <source src="la-moldau.ogg">  
  <!-- proposez ici une alternative en Flash ou QuickTime -->  
</audio>
```

<video>

HTML 5 introduit l'élément <video> pour insérer des séquences vidéo dans une page web. La syntaxe de base de l'élément vidéo est très simple :

```
<video src="une_video.mp4"></video>
```

Des attributs permettent d'apporter des informations complémentaires :

- controls affiche les possibilités de contrôle de la vidéo (avance, pause, stop...).
- autoplay lance la lecture automatiquement dès le chargement de la page.
- preload spécifie au navigateur de débiter le téléchargement de la vidéo tout de suite, en anticipant sur le fait que l'utilisateur lira la vidéo.

Les formulaires

<form></form>

Pour créer un formulaire, il vous faudra toujours commencer par les balises
<form></form>

```
<h1>Mon formulaire</h1>

<form>
  <p>Texte à l'intérieur du formulaire</p>
</form>
```


method

method : cet attribut indique par quel moyen les données vont être envoyées (réponse au problème n°1). Il existe deux solutions pour envoyer des données sur le Web :

- method="get" : méthode peu adaptée car limitée à 255 caractères. Les informations seront envoyées dans l'adresse URL de la page (http://...)
- method="post" : c'est la méthode la plus utilisée pour les formulaires car elle permet d'envoyer un grand nombre d'informations. Les données saisies dans le formulaire ne transitent pas par la barre d'adresse.

action

Adresse de la page qui va traiter les informations du formulaire

La page devra gérer ces informations grâce à un langage serveur (PHP, JAVA, Python, ect...)

```
<h1>Mon formulaire</h1>

<form method=« post » action=« reponse.php »>
  <p>Texte à l'intérieur du formulaire</p>
</form>
```

Zones de saisie : La zone de texte monoligne

Elle sert à saisir des textes courts, par exemple un prénom ou un pseudo : `<input type=« text » />`

- L'attribut `name` : sert à donner un nom à l'élément du formulaire
- Le libellé `<label>` : sert à « labelliser » l'élément (pour l'utilisateur)
- L'attribut `for` du `label` et `id` de l'`input` servent à se lier entre eux

```
<label for=« prenom">Votre prénom</label> : <input  
type="text" name= "prenom" id="prenom" />
```

Les attributs supplémentaires

On peut ajouter un certain nombre d'autres attributs à la balise `<input />` pour personnaliser son fonctionnement :

- On peut agrandir le champ avec `size`.
- On peut limiter le nombre de caractères que l'on peut saisir avec `maxlength`.
- On peut pré-remplir le champ avec une valeur par défaut à l'aide de `value`.
- On peut donner une indication sur le contenu du champ avec `placeholder`. Cette indication disparaîtra dès que le visiteur aura cliqué à l'intérieur du champ.
- On peut définir un mot de passe 'texte caché lors de la saisie' avec l'attribut : `type=« password »`

Zones de saisie : La zone de texte multilignes

Cette zone de texte permet d'écrire une quantité importante de texte sur plusieurs lignes.

```
<textarea rows="4" cols="50">  
  Ecrivez votre description ici  
</textarea>
```

Tout comme pour l'élément `<input type="text" />`, il vous faudra définir le nom et le label de l'élément `<textarea>`

Et encore pleins d'autres balises à découvrir

Submit

Image

Text

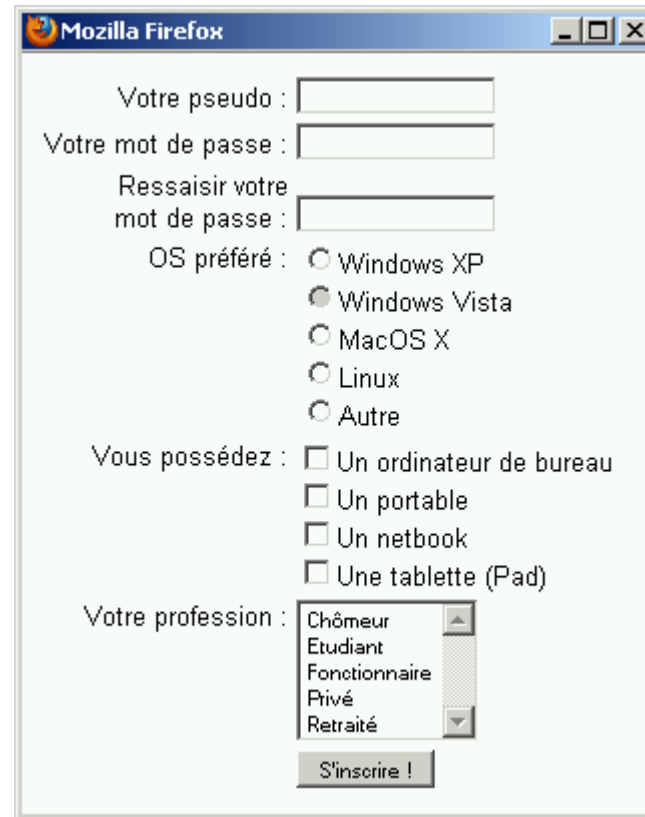
Radio

Button

Checkbox

Hidden

File



A screenshot of a Mozilla Firefox browser window displaying a registration form. The form includes the following fields and controls:

- Votre pseudo :
- Votre mot de passe :
- Ressaisir votre mot de passe :
- OS préféré : ☐ Windows XP, ☒ Windows Vista, ☐ MacOS X, ☐ Linux, ☐ Autre
- Vous possédez : ☐ Un ordinateur de bureau, ☐ Un portable, ☐ Un netbook, ☐ Une tablette (Pad)
- Votre profession : (dropdown menu with options: Chômeur, Etudiant, Fonctionnaire, Privé, Retraité)
- S'inscrire ! (Submit button)



A screenshot of a 6ma website registration form. The form is divided into two sections: "Coordonnées" and "Message".

Coordonnées

- Civilité : ☐ M., ☐ Mlle, ☐ Mme
- Nom :
- Adresse :
- Code postal :
- Ville :
- Pays : (dropdown menu)
- Centres d'intérêts : ☐ Sport, ☐ Cinéma, ☐ Internet, ☐ Voyages

Message

Envoyer Recommencer

Une nouvelle génération de formulaires

Les nouveaux types de champs

email : le champ requiert un contenu au format d'adresse électronique.

url : le champ accueille des URL absolues.

tel : le champ est destiné aux numéros de téléphone.

number : le champ accepte uniquement les caractères numériques.


color : le champ est prévu pour les chaînes représentant une valeur de couleur.

```
<input type="email" name="email">
```


Éléments valides ou invalides

Une idée ? Un projet ?

Votre nom



Votre e-mail

Votre message

ENVOYER

Chacun des types de champs obéit à une norme qui définit les valeurs acceptées ou rejetées.

Ainsi, il est possible de vérifier quels éléments sont invalides à l'aide des sélecteurs CSS 3 `:valid` et `:invalid`

```
[type="email"]:valid {background: green;}
```

Type=« number »

Au cas par cas, certains attributs complémentaires peuvent élargir les fonctionnalités des nouveaux éléments de formulaire. Le type number accepte par exemple les attributs min, max, step et value

```
<input type="number" min="0" max="10" step="2" value="6">
```

type="number" indique qu'il s'agit d'un champ de type numérique.

min="0" spécifie la valeur minimale acceptable pour ce champ.

max="10" est le maximum acceptable.

step="2", combiné à la valeur minimale indique que seules les valeurs 0, 2, 4, 6, 8 et 10 sont acceptées, par « pas » de 2.

value="6" est la valeur par défaut au chargement de la page.

range

Les champs de type range se présentent sous la forme d'un curseur défilant, ou slider

À l'instar du type number, un champ de type range accepte les propriétés complémentaires min, max, step et value

```
<input type="range" min="0" max="10" step="2" value="6">
```



date, datetime, month, week, time

Un certain nombre de nouveaux types de champs concernent les dates et l'heure :

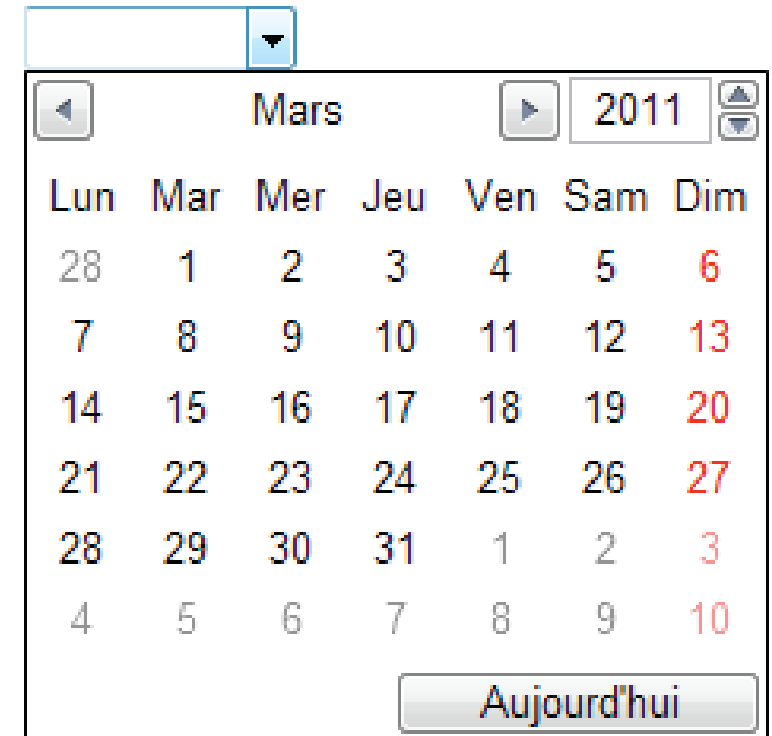
`<input type="date">` affiche un contrôle de date (année, mois, jour).

`<input type="datetime">` affiche un contrôle de date associé à une heure.

`<input type="month">` affiche un contrôle de mois.

`<input type="week">` affiche un contrôle de numéro de semaine.

`<input type="time">` affiche un contrôle d'heure.



search

Le type de champ search, comme son nom l'indique, désigne une zone de recherche au sein du formulaire

```
<input type="search" name="recherche">
```

placeholder

placeholder est un attribut applicable à tout élément de formulaire. Sa fonction est d'afficher une chaîne de caractères temporaire au sein du champ. Lors du clic ou de l'activation du champ à l'aide du clavier, ce texte disparaît pour laisser la place au contenu à saisir.

Idéal, par exemple, pour indiquer qu'un champ est obligatoire, placeholder s'applique à tous les champs de formulaires (<input>), même sur les éléments de type password, ce qui n'était pas réalisable précédemment, puisque toute valeur dans un tel champ s'affichait avec des caractères masqués.

```
<input type="search" name="recherche" placeholder="Recherche">
```

autofocus

L'attribut autofocus confère la focalisation à un champ déterminé dès le chargement de la page. Google a très bien compris l'utilité de cette fonctionnalité : grâce à autofocus, le champ de recherche est immédiatement opérationnel à l'affichage du site.

```
<input type="search" name="recherche" autofocus>
```

autocomplete

L'attribut autocomplete affiche une boîte contenant les dernières entrées de formulaire que vous avez saisies et qui ont été conservées en mémoire sur votre poste. La valeur par défaut est on (activé), mais il est possible de masquer ces termes à l'aide de la valeur off.

```
<input type="search" name="recherche" autocomplete="off">
```

http://www.goetter.fr
http://
http://www.polychromies.com
http://www.alsacreations.com
blog.goetter.fr
http://www.alsacreations.fr

required

Les champs bénéficiant de l'attribut `required` indiquent au navigateur qu'ils doivent nécessairement être remplis lors de la soumission du formulaire, sans quoi la validation doit être refusée et le champ concerné mis en évidence.

Cette étape, à présent native dans HTML 5, ne dispense pas de concevoir ces tests à l'aide de vérifications via JavaScript ou un langage serveur.

```
<form>  
  <input name="name" type="text" required>  
  <input type="submit" value="Go">  
</form>
```

CSS – media queries

Les media queries permettent de changer la résolutions du site selon des conditions données, comme la taille de l'écran, si l'utilisateurs utilise un smartphone, une télé ...

Il y a deux moyen d'applique une media query :

- En chargeant une feuille de style .css différent en fonction de la règle (ex : résolution inférieur à 1024 px, charge ce css) .
- En écrivant directement dans le fichier .css (ex : Si la résolution est inférieur à 1024 px, applique ces propriétés css).

```
1 <link rel="stylesheet" media="screen and (max-width: 1280px)" href="petite_resolution.css" />
```

```
1 @media screen and (max-width: 1280px)
2 {
3     /* Rédigez vos propriétés CSS ici */
4 }
```

CSS – media queries - exemples

```
1 /* Sur les écrans, quand la largeur de la fenêtre fait au maximum 1280px */
2 @media screen and (max-width: 1280px)
3
4 /* Sur tous types d'écran, quand la largeur de la fenêtre est comprise entre 1024px et 1280px */
5 @media all and (min-width: 1024px) and (max-width: 1280px)
6
7 /* Sur les téléviseurs */
8 @media tv
9
10 /* Sur tous types d'écrans orientés verticalement */
11 @media all and (orientation: portrait)
```

CSS – Déclarer des variables

En css il est possible comme pour les langages de programmations de déclarer des variables. Cela vous permettra si vous voulez changer le style de votre site d'avoir à changer uniquement la couleur de la variable plutôt que de repasser sur chaque élément.

Il faut déclarer la variable sur l'élément parent des enfants ou nous allons l'utiliser.

```
*{  
  --couleurPrincipale : pink;  
}  
body{  
  background: var(--couleurPrincipale);  
}
```