

HTML5 et CSS3

Lylia ABROUK

I.E.M. Université de Bourgogne
L3 Info

30 Janvier 2018

Plan

- 1 Rappels javascript
- 2 API de sélection dans le DOM
- 3 CSS 3

Principe

- Insertion des scripts (programmes) directement dans le code des pages HTML
- Exécution de code du coté client (temps de réponse plus court)
- Formulaire : test d'un formulaire avant envoi, affichage dynamique, ..
- Conçu pour traitement local des événements clients (click,déplacement souris, ...)
- JavaScript est un langage
 - interprété (le code est analysé et exécuté au fur et à mesure par le navigateur)
 - à base d'objets

Principe

- interagir avec un document HTML pour le rendre dynamique
- Créer de nombreuses fonctionnalités :
 - réaliser des galeries d'images,
 - transformations pour l'affichage de la page,
 - des réactions à des clics des utilisateurs,
 - créer des jeux vidéo,
 - des graphiques animés en 2D ou en 3D
- API (Application Programming Interfaces) intégrées aux navigateurs web
- Les API " tierces " développées en dehors des navigateurs pour utiliser les fonctionnalités d'autres sites.

Types de données

- String : Une chaîne de texte. `var prenom = 'Lydia' ;`
- Nombre : Un nombre. `var nb = 10 ;`
- Booléen : Une valeur qui signifie vrai ou faux. `var maVariable = true ;`
- Tableau : Une structure qui permet de stocker plusieurs valeurs dans une seule variable. `var couleur = [1,"jaune","rouge",2].`
Utilisation : `couleur[0], couleur[1],...`
- Objet : Un objet peut être n'importe quoi. Un objet est un ensemble de propriétés qui peut être enregistré dans une variable. C'est la structure la plus utilisée et la plus flexible.
`var maVoiture = new Object() ;`
`maVoiture.fabricant = 'Ford' ;`

Structures

- Structures de contrôle : conditionnelle if ...else
- Boucles : while (), do...while, for(init ; test ; incr)
- break : sort de la boucle immédiatement, continue : reprend à l'itération suivante.

Fonction

- `function $f(x_1, \dots, x_n)$ { // instructions };`
- Une méthode : une fonction associée à un objet, une action que l'on peut faire exécuter à un objet.

Interaction avec le navigateur

- `document` : L'objet global représente le document HTML. Il implémente l'interface DOM.
- `document.getElementById("toto")` méthode qui permet de récupérer un objet représentant l'élément HTML de la page ayant l'attribut `id` valant "toto"
- `toto.addEventListener("event", f)` : Exécute la fonction `f` quand l'évènement "event" se produit sur l'élément `toto` (ou un de ces descendants).
- `toto.innerHTML = "bonjour"` : Remplace tout le contenu de l'élément `toto` par le fragment de document contenu dans la chaîne de caractère.
- `toto.value` : Permet de modifier ou récupérer la valeur de certains éléments (zones de saisies de texte)

Les évènements

- onclick : un clic du bouton gauche de la souris sur une cible
- onMouseOver : passage du pointeur de la souris sur une cible
- onBlur : une perte de focus d'une cible
- onFocus : une activation d'une cible
- onSelect : une sélection d'une cible
- onChange : une modification du contenu d'une cible
- onSubmit : une soumission d'un formulaire
- onload : un chargement d'une page
- onunload : la fermeture d'une fenêtre ou le chargement d'une page autre que la courante

javascript et HTML 5

- Les nouveaux sélecteurs : `getElementsByName`, `querySelectorAll` ...
- Le stockage web ou local : `localStorage`
- Les BDD Web SQL
- Les API de cache pour les apps
- Les sockets web
- Les notifications
- Drag and drop depuis le bureau
- Geolocalisation

L'API JavaScript - Selectors

- Cette API sert à faire correspondre des éléments du DOM par leur Id, Tag, classe CSS.
 - Existant : par Id, par Tag
- HTML5 : par classe CSS
 - `getElementsByClassName`

L'API JavaScript - Selectors

- `document.querySelector`
 - Retourne le premier élément dans le document qui correspond au groupe de sélecteurs passés en paramètre.
 - `element = document.querySelector(sélecteurs);`
 - Exemple : `var el = document.querySelector(".myclass");`
- `document.querySelectorAll`
 - Retourne la liste des éléments dans le document qui correspondent au groupe de sélecteurs passés en paramètre.
 - `elements = document.querySelectorAll(selecteurs);`
 - Exemple : `var el = document.querySelector(".myclass");`

L'API JavaScript - Selectors

- `element.querySelector`
 - Retourne le premier élément parmi les descendants de l'élément sur lequel on l'invoque qui correspond au groupe de sélecteurs spécifiés.
 - `element = baseElement.querySelector(selectors);`
 - `var div = document.getElementById("bar");`
`var p = div.querySelector("p");`
- `element.querySelectorAll`
 - Retourne la liste des éléments descendants de l'élément sur lequel on l'invoque qui correspond au groupe de sélecteurs spécifiés.
 - `element = baseElement.querySelectorAll(selectors);`
 -

Plan

- 1 Rappels javascript
- 2 API de sélection dans le DOM
- 3 **CSS 3**

CSS

- Les nouveaux sélecteurs CSS : nth-child(even), nth-child(odd), :not, first-child.
- Bordures et fonds
- Images
- Effets texte
- Transitions
- Transformation et rotations d'éléments HTML (2D/3D)
- Animations en CSS
- Gestion de colonnes : -webkit-column-count, -webkit-column-rule, -webkit-column-gap
- Interface utilisateur

Sélecteurs

- selecteur `element1 ~ element2`
 - `element ~ element { declarations css; }`
- selecteur `[attribute $\hat{=}$ value]`
 - `[attribute $\hat{=}$ value]{ declarations css; }`
- selecteur `[attribute. =value]`
 - `[attribute. =value] { declarations css; }`
- selecteur `[attribute* =value]` : sous chaîne de caractère
 - `[attribute* =value] { declarations css; }`
- `:checked` : OPTION de la balise SELECT et INPUT type radio ou checkbox
 - `:checked { declarations css; }`
- `:disabled`
- `:empty`

Sélecteurs

- `:first-of-type` : Premier type d'élément
- `:not(p)` : Négation de l'expression
- `:nth-child(n)` : n-ième enfant
- `:nth-last-child(n)` : n-ième dernier enfant
- `:nth-last-of-type(n)` : n-ième dernier type d'élément
- `:nth-of-type(n)` : n-ième type d'élément
- `:only-of-type` : Élément de type unique
- `:last-child` : dernier enfant
- `:first-child` : Premier enfant

Sélecteurs - formulaires

- `:in-range` : La valeur est dans l'intervalle demandé (attributs `min` et `max`). (`:out-of-range`)
- `:valid` : La valeur est valide (par exemple, la syntaxe de l'adresse e-mail ou de l'URL est vérifiée). (`:invalid`)
- `:required` : La valeur est requise (attribut `required`).
- `:optional` : La valeur est optionnelle (absence d'attribut `required`).

Fonds

- background-size : Taille de l'image de fond par rapport aux dimensions de l'élément
- background-origin : Position du fond par rapport à la boîte de l'élément
- background-clip : définit la façon dont l'arrière-plan d'un élément s'étend sous la bordure.

Fonds

- `border-image` : ajouter des contours décoratifs à n'importe quel élément
- Propriétés :
 - `border-image-source` : La source de l'image de contour
 - `border-image-slice` : la distance à partir de chaque coin de l'image
 - `border-image-width` : la largeur du contour de l'élément
 - `border-image-outset` : décrit l'espace dont l'image de la bordure dépasse de la boîte de bordure.
 - `border-image-repeat` : définit la façon dont les zones de l'image de bordure sont utilisées pour s'étendre sur chacun des côtés de la bordure (stretch, repeat, round, space)

Textes

- `text-overflow` : Détermine le mode d'affichage lorsque le texte dépasse la limite de son conteneur
- `text-shadow` : Applique un effet d'ombre portée sur le texte
 - `blur` : netteté / flou de l'ombre portée
 - `color` : couleur du texte de l'ombre portée
 - `h-shadow` : décalage horizontal du texte de l'ombre portée par rapport au texte d'origine
 - `v-shadow` : décalage vertical du texte de l'ombre portée par rapport au texte d'origine
- `word-wrap` : Permet de faire un retour à la ligne au milieu d'un mot
- `word-break` : Spécifie comment effectuer des retours à la ligne en coupant des mots

Transformation 2D

- `scale ()`
 - permet d'augmenter ou réduire la taille d'un élément
 - *transform* : `scale(0.5, 0.5);`

Transformation 2D

- `translate ()`
 - permet de déplacer un élément depuis sa position d'origine à l'écran, selon des coordonnées x et y.
 - *transform : scale(1.5)translate(20px,40px) ;*

Transformation 2D

- `skew()` : `skew` prend deux coordonnées `x` et `y` et déforme un élément sur deux axes
- `skewX()`
- `skewY()`

Transformation 2D

- `matrix()`
 - combine les méthode de transformation
 - `matrix(scaleX(),skewY(),skewX(),scaleY(), translateX(),translateY())`

Transformation 3D

- rotateX()
- rotateY()
- rotateZ()

Transition

- Les Transitions permettent de modifier les valeurs de propriétés CSS dans le temps, d'un état A à un état B.
- Propriétés
 - transition-property : les propriétés CSS à modifier.
 - transition-duration : le temps total de la transition (en secondes, millisecondes).
 - transition-timing-function : la méthode d'interpolation (accélération, décélération)
 - transition-delay : le temps avant que la transition ne démarre.

Transition

- transition-timing-function : précise la méthode d'accélération entre l'état initial et final d'une transition
- Valeurs
 - ease : action par défaut. Accélération naturelle. Le départ est rapide, l'arrivée lent.
 - ease-in : départ lent. L'arrivée paraît donc plus rapide.
 - ease-out : arrivée lent. Le départ paraît donc plus rapide.
 - ease-in-out : départ lent, arrivée lent. Une accélération se fait sentir au milieu.
 - cubic-bezier(p1, p2, p3, p4) : création d'une courbe d'accélération personnalisée (grâce aux courbes de bézier).
 - steps(nombre, start|end) : Progression par palier (animation image par image)

Animation

- Principe = évolution des transitions. Les mêmes propriétés sont animables
- Différence entre une animation et une transition :
 - Une animation n'est pas entre deux états, mais entre n états,
 - Définition d'un ensemble de KeyFrames (états), associés à un pourcentage (0% = début, 100% = fin, 50 % = milieu) et les valeurs des propriétés pour chaque état.
 - Nommer l'animation
 - Associer l'animation à des objets de la page (durée).

Animation

- `animation-name` : le nom de l'animation à utiliser.
- `animation-duration` : le temps total de l'animation.
- `animation-timing-function` : la méthode d'interpolation (accélération, décélération). : `linear`, `ease`, `ease-in`, `ease-out`, `ease-in-out`, `steps` (nombre, `start` | `end`), `cubic-bezier`(`p1`, `p2`, `p3`, `p4`)
- `animation-iteration-count` : le nombre de répétition de l'animation. La valeur infinie permet de jouer une animation en continu.
- `animation-direction` : permet de jouer une animation en sens inverse (en fonction du cycle).
- `animation-delay` : le temps avant que l'animation ne démarre.
- `animation-fill-mode` : conserver l'état de l'animation avant le début où après la fin de celle-ci.

Gestion des colonnes

- contenu sur plusieurs colonnes de largeurs égales (imprimerie).
 - column-count : représente le nombre maximal de colonnes
 - column-fill : contrôle la façon dont le contenu est réparti entre les colonnes
 - column-gap : permet de définir l'espace entre les colonnes
 - column-rule définit l'affichage d'une ligne droite entre chaque colonne
 - column-rule-color : définit la couleur de la ligne tracée entre les colonnes
 - column-rule-style : définit le style de la ligne tracée
 - column-rule-width : définit la largeur de la ligne tracée
 - column-span : offre à un élément la possibilité de s'étaler sur une (valeur none) ou toutes les colonnes à la fois (valeur all)
 - column-width : Largeur optimale
 - columns : column-width et column-count

Interface utilisateur

- redimensionnement, contours, et la boîte de dimensionnement..
 - resize : un élément doit être redimensionnable par l'utilisateur ou non (horizontal, vertical, both).
 - outline-offset : ajoute de l'espace entre les contours et la bordure d'un élément

propriétés

- `outline-offset` : définit l'espace qu'il y a entre la bordure dessinée par `outline` et le bord de la boîte de bordure
- `resize` : si l'élément peut être redimensionné

Media Queries

- Le type de média
- La taille de l'écran
- La taille de la fenêtre
- La résolution
- Le nombre de couleurs
- L'orientation
-

Pour s'amuser

- Exemples W3C school

Références

- HTML 5 Une référence pour le développeur web, Rodolphe Rimelé.
- Cours HTML5, Michel Buffa, UNSA 2012
- <http://www.css3create.com/>
- <http://www.w3schools.com/>