

Javascript jQuery

Guy Lapalme

Javascript

- Langage de programmation objet
- Présent dans *tous* les browsers
- Pas toujours activé, mais presque...
- Permet de manipuler
 - le contenu d'une page web une fois chargée
 - tous les éléments du browser
 - valider les éléments d'un formulaire
- Programmation serveur aussi possible
 - node.js
- Pas d'énoncé d'entrée-sortie dans la langage
 - mais, `console.log(...)` ou `document.write(...)`

Javascript : brefs rappels - I

plus de détails

- Types de base
 - nombres, chaines de caractères, booléens, `null`, `undefined`
 - objets
 - tableau
 - fonction
- Variables (déclarées avec `var`)
- Opérateurs (`+`, `-`, `*`, `/`, `<`, `<=`, `==`, `>`, `>=`)
- Énoncés
 - `if (cond) {...} else {...}`
 - `for (var i=0; i<n; i++) {...}`

Javascript : brefs rappels - 2

- Tableau

- `var t = ["veau", "vache", "cochon"]`
- `t.length, t[2], t.push("couvée"), t.pop()`
- `t.join(" et "), t.sort()`

- Fonction

```
var fact = function(n) {  
    if (n == 1) return 1;  
    return n*fact(n-1);  
};
```

Javascript : brefs rappels - 3

```
var personne={prenom:"Alain",nom:"Ternet",
  setNom:function(newNom) {
    this.nom=newNom;
    return this;
  }
};
```

```
personne.prenom+" "+personne["nom"] => ??
personne.setNom("Formatique");
```

```
personne.setPrenom=function(newPrenom) {
  this.prenom=newPrenom;
  return this;
};
```

```
personne.prenom+" "+personne["nom"] => ??
```

```
personne.setNom("Dinateur").setPrenom("Laure")
personne.prenom+" "+personne["nom"] => ??
```

Objets
simples

Autres exemples de création d'objets

Accès aux éléments du browser

- Objets et propriétés prédéfinies
 - Document Object Model
 - `node`, `children`, ...
 - Browser Object Model
 - `window`, `frames`, `history`, `location`, ...
- Gestion des événements par appel de fonction (*callback*, *event listener*)
 - `click`, `load`, `mouseover`, `submit` ...

Lien avec une page web

- Balise script
 - soit dans le corps de la page (à éviter)
 - soit dans l'entête
- Exemple : afficher la date du browser
 - Exécuter source HTML source Javascript
- Exemple: Compteur avec un *formulaire*
 - Exécuter source HTML source Javascript css

Frameworks Javascript

- Ensemble de fonctions JS pour la programmation client
 - manipulation DOM
 - extensions diverses aux tableaux, chaînes
 - gestion uniforme des événements
 - widgets pour interface usager
- Permet de s'affranchir des particularités des browsers (y compris diverses versions)
- Utilise à fond les aspects fonctionnels de JS
 - jQuery - *Manipulation du DOM et interaction*
 - D3 - *Graphiques*

jQuery

<http://jquery.com>

fortement inspiré de : <http://web.stanford.edu/class/cs98si/slides/jquery.html>

- Orienté requête pour chercher et manipuler le DOM
- Sélecteurs CSS pour identifier des éléments du DOM
- Exécution de fonctions sur tous les éléments identifiés

- `$(document).ready(code de lancement)`

- Doit charger le code dans l'entête

```
<script src="//code.jquery.com/jquery-latest.min.js"></script>
```

\$

- désigne **jQuery** : seule variable globale exportée
- aussi une fonction surchargée selon son paramètre
 - sélecteur CSS : collection d'éléments DOM
 - chaîne HTML : crée un nouvel élément DOM
 - élément DOM : retourne cet élément comme une collection jQuery
 - vide : collection jQuery vide
 - collection jQuery : retourne cette collection

Sélection

$\$(\textit{sélecteur CSS}[, \textit{contexte}])$

*	tous les éléments
.class	tous les éléments de cette classe
élément	tous les éléments de ce nom
#id	l'élément avec cet id
sel₁, sel₂, ...	combine tous les résultats des sélecteurs
parent enfant	tous les enfants d'un parent

Si *contexte* est présent, la sélection est limitée aux enfants de cet élément

Collection jQuery

- résultat d'un appel à `$ (...)`
- semblable un `array`
 - attribut `length`
 - accès par `[n]`
 - **retourne un élément DOM**
 - **pas un objet jQuery**
- donne accès à une foule de fonctions
 - `.css()`, `.text()`, `.html()`, `.each()`, ...
 - `.add()`, `.filter()`, `.not()` // comme des ensembles
- retourne la collection modifiée
- *convention*: débiter nom de variable par `$`

objet jQuery vs élément DOM

```
var $toto = $("#toto")
```

```
var toto = document.getElementById("toto")
```

```
var $bleus = $(".bleu")
```

```
var bleu = document.getElementsByClassName("bleu")
```

```
var $toto = $(toto)
```

```
var toto = $toto.get(0) // ou $toto[0]
```

Manipulation éléments DOM

- `.attr(nom attribut [, valeur])`
 - obtenir la valeur ou changer la valeur d'un attribut
- `.css(nom propriété, valeur)`
 - obtenir la valeur ou changer la valeur d'une propriété de style CSS
- `.addClass/ .removeClass/ .toggleClass(nomsDeClasses)`
 - ajouter/enlever ou basculer la présence des noms de classes séparés par des blancs dans une chaîne
- `.hide() / .show()`
 - cacher ou afficher des éléments
- `.html(...)` / `.text(...)`
 - changer le contenu HTML ou texte d'un élément

Manipulation d'éléments DOM

- `.append(élément)`, `.prepend(élément)`
 - ajouter l'élément comme dernier/premier enfant
- `.after(élément)`, `.before(élément)`
 - ajouter l'élément après / avant comme frère
- `.remove()`
 - enlever un élément pour le remettre ailleurs
- `.empty()`
 - enlever tous les éléments enfants

Comme pour le DOM, un élément existant ajouté est enlevé de son endroit précédent

Traitement des événements

- `.on("...", f(ev))` ajoute un listener
- `.off("...", f(ev))` enlève un listener
- `.click(f(ev))` ajoute un click listener
- `.keypress(f(ev))` ajoute un key listener
- `.fadeIn(...)` / `.fadeOut(...)`
apparition graduelle
- `.slideDown(...)` / `.slideUp(...)`
apparition par déplacement

Exemples jQuery

- Compteur avec un formulaire
 - [Exécuter source HTML](#) [source Javascript](#) [css](#)
- Calculette
 - [Exécuter source HTML](#) [source Javascript](#) [css](#)
- Tri d'un tableau (V0)
 - [Exécuter source HTML](#) [source Javascript](#) [css](#)
- Tri d'un tableau
 - [Exécuter source HTML](#) [source Javascript](#) [css](#)
- Jeu d'arcade SVG
 - [Exécuter source HTML](#) [source Javascript](#)
- Galerie de photos
 - [Exécuter source HTML](#) [source Javascript](#) [css](#)