Javascript et les applications Web

Télécom ParisTech
Jean-Claude Moissinac – Octobre 2014
Avec l'aide de Jean-Claude Dufourd
Et Thomas Bertrand

Mastère CPD

JavaScript

- **Bases**
  - Syntaxe de base, variables, fonctions, expressions, boucles, conditions
- **Programmation « avancée »**
  - Object, Array et autres objets globaux
  - Function, arguments, call, apply, map
- **Programmation Object: constructeur, héritage, surcharge,…**
- **Environnements (navigateur, node.js)**

Javascript

- **Exécution dans les navigateurs en complément des pages Web**
  - Usage le plus courant
- **Exécution possible dans d'autres environnements**
  - En 'extension' de certains programmes
  - Dans nodejs

Nodejs

- **Environnement qui permet d'exécuter du code javascript dans un ordinateur sans passer par un navigateur**
- **Permet de développer des applications**
- **Permet notamment de développer des applications 'serveur'**
  - Comme PHP le permet

Variables et Javascript

- **LES BASES PAR L'EXEMPLE**
  - var x;
  - var y = 0;
  - x = 0;
  - x = 1;
  - x = 0.01;
  - x = "hello";
  - x = 'Hello world!';
  - x = true;
  - x = false;
  - x = null;
- x = undefined;

Types de base

- Boolean
- Number
- String
- Object
- Function
- NaN
- null

Type table

var primes = [2,3,5,7];

primes[0] → 2

primes.length → 4

primes[primes.length – 1] → 7

primes[4] = 9;

primes[4] = 11;

var uneNouvelleTable = []; // création table vide

uneNouvelleTable.length → 0

Type ‘objet’

```
var book = {
        topic: "JavaScript",
        hard: true
};
```

book.topic → JavaScript
Book.hard → true
book["hard"] → true
book.author = "Jean Dupont"; //ajout de l’attribut author
book.contents = {}; // ajout d’un objet vide

Combiner table et objet

```
var points = [
        {x: 0, y: 1},
        {x: 1, y: 1}
];
var data = {
        p1: {x: 0, y: 1},
        p2: {x: 1, y: 1}
};
var trials = {
        trial1: [[1, 2], [3, 4]],
        trial2: [[1, 2], [4, 6]]
 };
```

points[1] -> ?

data.p2 -> ?

data[« p1 »] -> ?

Trials.trial2[0] -> ?

Opérations de base

- 3 + 2 → 5
- 3 * 2 → 6
- 3 – 2 → 1
- 3 / 2 → 1.5
- 3 % 2 → 1 (modulo)
- "3"+"2" → 32 (concaténation)
- data.p1.x + 3 → 3

- primes["2"] → 5
- primes.2 → SyntaxError: unexpected number
- var count = 0;
- count++;
- count--;
- count += 2;
- count -= 4;
- count *= 5;
- +str convertit str en nombre

Des comparaisons (1)

Valeurs de x et valeur de y

- **var x=3;**
- **var y=2;**

Réponse à la question est-ce que x est égal à y?

- **x == y**

→ false

Réponse à la question est-ce que x est égal à 3?

- **x==3**

→ true

Réponse à la question est-ce que x est différent de y?

- **x != y**

Réponse à la question est-ce que x est inférieur à y?

- **x < y**

→ false

Réponse à la question est-ce que x est supérieur ou égal à y?

- **x >= y**

Réponse à la question est-ce que x est inférieur ou égal à y?

- **x <= y**

Réponse à la question est-ce que x est supérieur à y?

- **x > y**

Réponse à la question est-ce que false est égal au résultat de (x<y)?

- **false == (x < y)**

→ true

Des comparaisons (2)

Est-ce que x est égal à 2 ET y est égal à 3?

- **(x == 2) && (y == 3)**

Est-ce que x est égal à 2 OU y est égal à 3?

- **(x == 2) || (y == 3)**

L'inverse de la réponse à la question est-ce que x est égla à y

- **!(x == y)**

Tests pour contrôler les actions

```
if (b == 0)
{
        x = 4;
        y = 2;
}
```

```
if (b == 0)
{
        x = 4;
        y = 2;
}
else
{
        x = -4;
}
```

D’autres contrôles

```
while (b!=23)
{
        afficherUnPoint();
        b = nombreAleatoire(0,30);
}
```

```
for (i=0; i<10; i++)
{
   afficherUnPoint();
}
```

Plus tard
for (a in obj) {…}

Attention, égalité, etc

```
2 == 2 → true
"2" == 2 → true !!!
égalité après chgts de type
"2" === 2  → false
égalité stricte

"22" > "3" → false
+"22" > "3" → true
```

Fonctions

- **Bloc d'instructions auquel on donne un nom et des paramètres**

Je définis la fonction:

```
function maBelleAddition (param1, param2)
{
        var resultat = param1 +param2;
        return resultat;
}
```

Ailleurs je l'utilise

```
Tot1 = maBelleAddition (1, 5);
Tot2 = maBelleAddition (6, 12);
```

Exemples de fonctions

```
function inc(x)
{
      return x+1;
}
inc(4) → 5
function square(x)
{
      return x*x;
}
square(inc(4)) → 25
```

```
var a = [];
a.push(4);
a.reverse();

points.dist =
function() {…}
points.dist();
```

A noter

```
function inc(x) { return x+1; }
inc est une fonction
inc(4) est une invocation de la fonction

var a = [];      (ou new Array() )
a.push(4);
push est une méthode de Array
a.push(4) est une invocation de méthode
```

Exercices