

TELECOM
ParisTech



Institut
Mines-Télécom

JAVASCRIPT et les applications web

partie 2

Télécom ParisTech

Jean-Claude Moissinac – Octobre 2014

Avec l'aide de Jean-Claude Dufourd

Et Thomas Bertrand

Mastère CPD



Array

```
a = new Array();  
Est équivalent à  
a = [];
```

```
a = [1, 2, 8];  
a.length → 3  
a.push(4);  
b = a.pop(); → 4  
delete a[1];  
1 in a → false  
a.length → 3 non modifié par delete  
a.join() → "1,2,8"  
a.join(" ") → "1 2 8"  
a.reverse()  
a.sort()  
a.concat(...)  
a.slice(...)  
-1 est le dernier element, -3 l'antepenultième...
```

Array 2

```
a.shift() // = pop à gauche  
a.unshift() // = push à gauche  
a.toString() // comme join()
```

Versions récentes de javascript:

```
a.forEach(f) // appliquer une fonction  
a.map(f) // + retour tableau résultats  
a.filter(f) // selection selon prédicat  
a.every(f) // && sur prédicat  
a.some(f) // || sur prédicat  
a.reduce(f,i) // applique f (2 args) à tous les  
éléments de gauche à droite et retourne le résultat  
accumulé  
a.reduceRight(f,i) // de droite à gauche  
a.indexOf()  
a.lastIndexOf()  
Array.isArray(a)
```

String

```
var s = "hello, world"  
s.charAt(0)  
s.charAt(s.length-1)  
s.substring(1,4)  
s.slice(1,4)  
s.slice(-3)  
s.indexOf("l")  
s.lastIndexOf("l")  
s.indexOf("l", 3)  
s.split(",")  
s.replace("h", "H")  
s.toUpperCase()
```

Math

```
Math.pow(2, 53)
Math.round(.6)
Math.ceil(.6)
Math.floor(.6)
Math.abs(-5)
Math.max(x, y, z)
Math.min(x, y, z)
Math.random()
Math.PI
Math.E
Math.sqrt(3)
Math.pow(3, 1/3)
Math.sin(0)
Math.log(10)
Math.log(100) / Math.LN10
Math.log(512) / Math.LN2
Math.exp(3)
```

JSON

- Dans versions récentes de javascript

```
obj = JSON.parse(line);
```

```
line = JSON.stringify(obj);
```

Exemple:

```
var book = {  
    topic: "Javascript", fat: true,  
    author: "Jean",  
};
```

```
JSON.stringify(book) →
```

```
'{"topic":"JavaScript","fat":true,\n"author":"Jean"}'
```

JSON

■ Une façon commode de

- sauver des objets complexes dans un fichier puis de les relire
- Envoyer des objets complexes dans des messages sur le réseau puis de les reconstruire à distance

delete

Permet de détruire un attribut d'un objet ou un élément d'une table

```
var o = { x: 1, y: 2};
```

```
delete o.x;
```

```
"x" in o → false
```

```
var a = [1,2,3];
```

```
delete a[2];
```

```
a.length → 2
```


Dans le navigateur

```
window
window.URL
window.title
window.cookie
window.referrer
x = 4;
foo = "x";
window[foo] → 4
document
document.root
document.body
o = document.getElementById("id")
a = document.getElementsByTagName("h2")
o.appendChild(document.createElement("h3"));
obj.insertBefore(ch, obj.childNodes[4]);
o.firstChild
o.lastChild
```

Dans le navigateur 2

```
o.nextSibling
o.textContent
o.style.cssText
list = obj.addEventListener('event', func);
removeEventListener('event', list);
dans HTML: onclick="func"
obj.onclick=function(evt) {...}
history.go(-2);
window.screen.width/height
w = window.open(url, target, style)
    target peut être _top, _blank, _parent
w.close();
a = document.getElementsByClassName("cssClass");
```

Portée dans l'exemple du TP

```
var http = require("http");
var iglobal = 0;
function generePage(request, response) {
  var ilocal = 0;
  response.write('iglobal vaut '+ iglobal+ '\n');
  response.write('ilocal vaut '+ ilocal+ '\n');
  response.end() ;
  iglobal++;
  ilocal++;
}
var server = http.createServer(generePage);
server.listen(8080);
```

Bibliothèques

- **Beaucoup de bibliothèques ont**
 - du CSS pour configurer les aspects
 - des plugins pour les étendre
- **Les "styles" sont très variables**
 - le \$ de jQuery rend le code difficile à lire
 - beaucoup de "chaining":

```
$("#id3").get().sort().slice(2,5);
```
- **C'est un nouvel environnement à chaque fois, avec ses trucs et astuces pas évidentes**
- **Très puissants mais pas toujours très bien documenté**