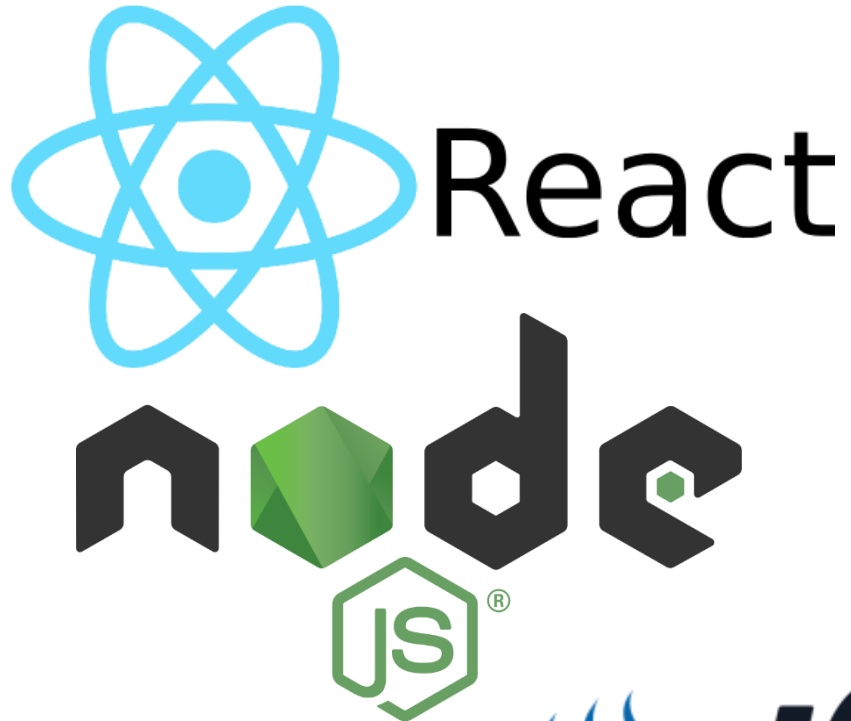




Le développement front avec JavaScript

L'usage du JavaScript aujourd'hui

Introduction au JavaScript



L'usage du JavaScript au tout début

Introduction au JavaScript

JavaScript



Historique et acteurs

Introduction au JavaScript



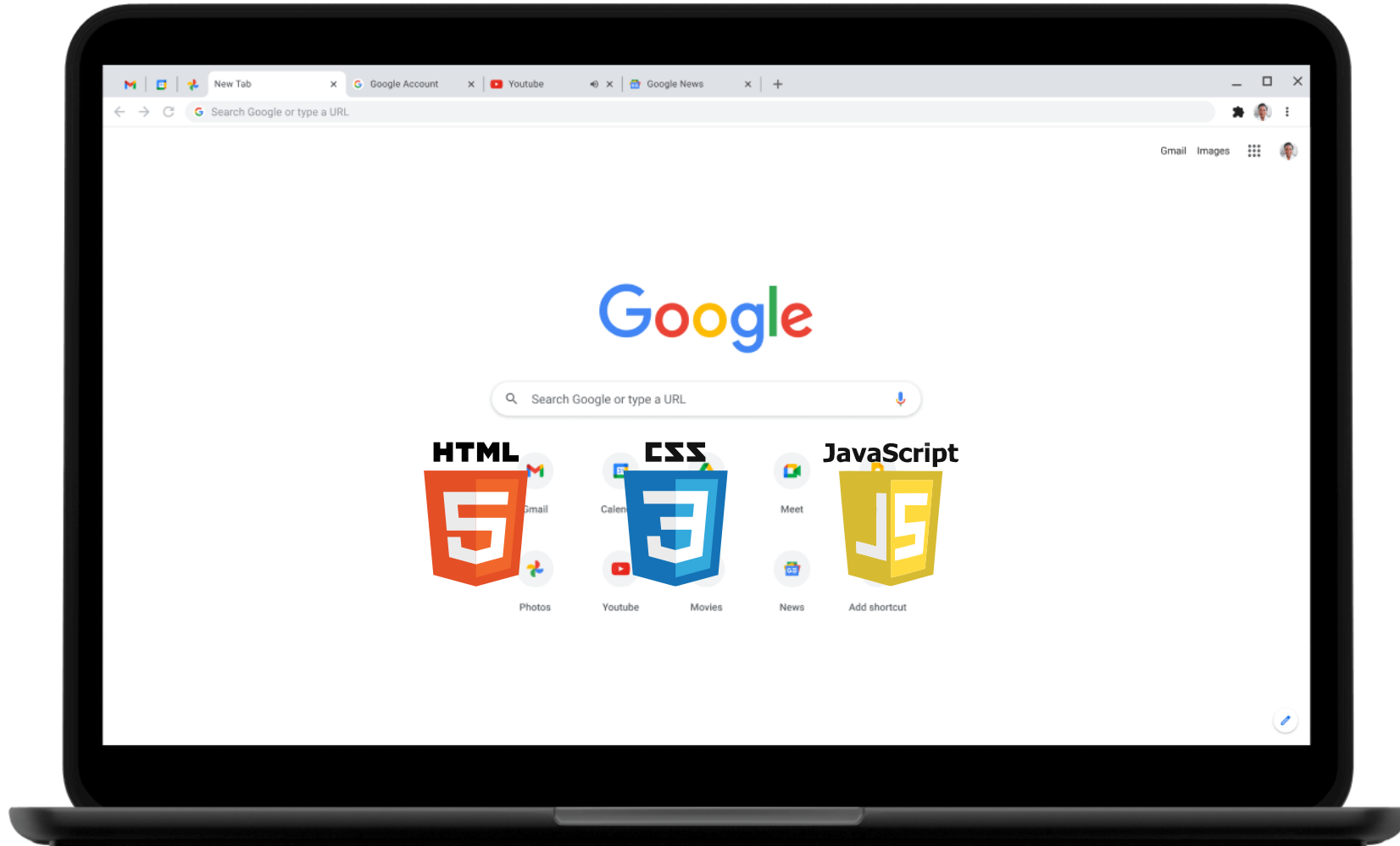
<https://www.ecma-international.org/memento/tc39-m.htm>

The ECMAScript™ specification has been developed by Ecma TC39 whose membership includes all major browser vendors.



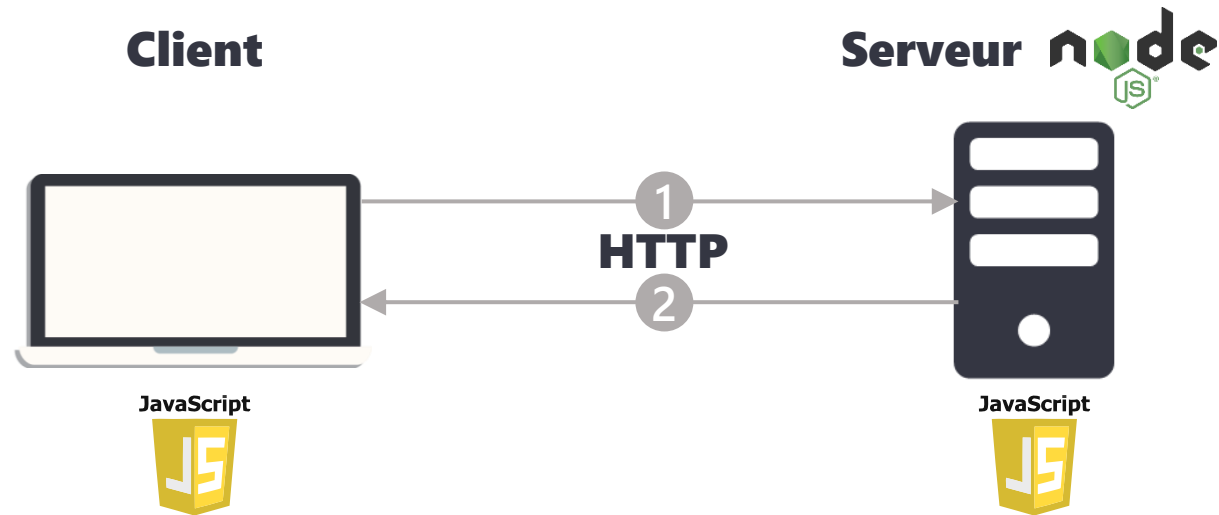
Un langage interprété par les navigateurs

Introduction au JavaScript



Un langage Full Stack

Introduction au JavaScript



Le langage JavaScript

Introduction au JavaScript

- Le JavaScript respecte les paradigmes :
 - Objet
 - Impératif
 - Fonctionnel
- Le JavaScript est un langage interprété



Les variables

Introduction au JavaScript

- Type dynamique

```
var maVariable;
```

- Les principaux types primitifs :

boolean

null

undefined

number

string

- Les types objet : Date, Math, Array, Object, Function, NaN...



Les fonctions

Introduction au JavaScript

```
function nom([parametre[, parametre[, ... parametre]]])  
{  
    instructions  
    [return ...]  
}
```



Les opérateurs de base

Introduction au JavaScript

- Presque comme en Java !
- La principale nouveauté : l'égalité stricte

=== !==

```
console.log("1"==1); //true
```

```
console.log("1"===1); //false
```



Les structures de code

Introduction au JavaScript

- Les conditionnelles
 - if ... else
 - switch
- Les boucles
 - while
 - do ... while
 - for, for ... in, for ... of
- La gestion d'erreurs
 - try ... catch



Les principaux objets disponibles

Introduction au JavaScript

- String
 - `charAt()`, `concat()`, `indexOf()`, `substring()`...
- Number
 - `MAX_VALUE`, `MIN_VALUE`, `isInteger()`, `isNaN()`...
- Math
 - `random()`, `floor()`, `ceil()`...
- Date
 - `getDate()`, `getMonth()+1`, `getFullYear()`, `getHours()`... `setMinutes()`, `setSeconds()`... `toLocaleDateString()`...
- Error
 - `name`, `message`
- Propriétés et variables globales
 - `Infinity`, `NaN`, `undefined`, `isNaN()`, `Number()`, `parseFloat()`, `parseInt()`...



Les tableaux

Introduction au JavaScript

- Déclaration

```
var etablisements = [ 'Maternelle', 'Primaire' ];  
var eleves = [ ];
```

- Utilisation

- Comme en Java
- Taille non fixée

- Propriétés et méthodes disponibles

- length, push(), pop(), slice(), sort(), toString()...





DOM par le JavaScript

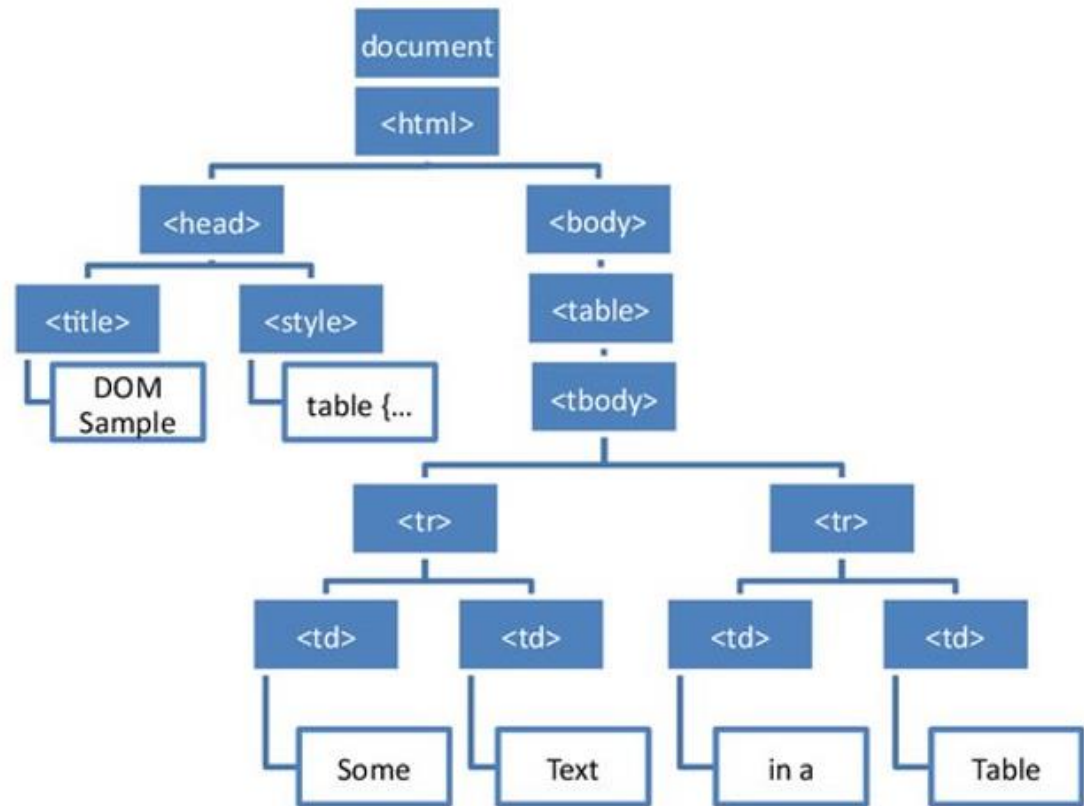
Qu'est-ce que le DOM (Document Object Model)

Document Object Model

DOM Tree



```
<!DOCTYPE html>
<html>
  <head>
    <title>DOM Sample</title>
    <style type="text/css">
      table {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <table>
      <tbody>
        <tr>
          <td>Some</td>
          <td>Text</td>
        </tr>
        <tr>
          <td>in a</td>
          <td>Table</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```



Les principales propriétés (Document)

Document Object Model

`<html>` *document*.documentElement;

`<head>` *document*.head;

`<title>` *document*.title;

`<body>` *document*.body;

`<form>` *document*.forms;

`<form>`

`<form>`



Les principales méthodes (Document)

Document Object Model

<element>

document.getElementById();

<element>

document.querySelector();

<element>

document.getElementsByTagName();

<element>

document.querySelectorAll();

document.write();

<element>

document.createElement();

attribut

document.createAttribute();

Du texte

document.createTextNode();



Les principales propriétés (<element>)

Document Object Model

```
unElement.id;  
unElement.value;  
unElement.attributes;  
unElement.innerText;  
unElement.innerHTML;
```

```
unElement.firstElementChild;  
unElement.lastElementChild;
```

```
unElement.nextElementSibling;  
unElement.previousElementSibling;  
unElement.parentElement;
```



Les principales méthodes (<element>)

Document Object Model

attribut

```
unElement.getAttribute();  
unElement.hasAttribute();  
unElement.hasAttributes();
```

```
unElement.focus();
```

<element>

```
unElement.getElementsByTagName();  
unElement.querySelector();
```

```
unElement.insertAdjacentElement();  
unElement.setAttribute();  
unElement.appendChild();
```

```
unElement.replaceChild();
```

```
unElement.removeChild();  
unElement.removeAttribute();
```



Intégration du JavaScript dans une page HTML

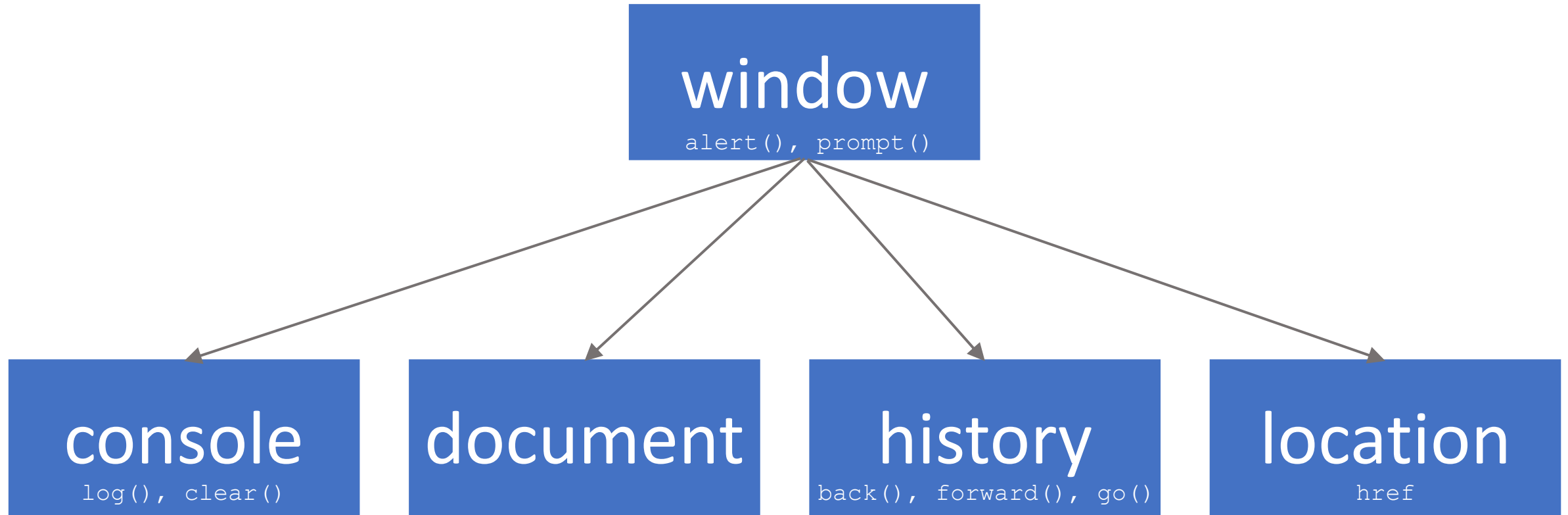
Document Object Model

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script>
    //Votre code JavaScript
  </script>
  <script src="module1.js"></script>
</head>
<body>
  ...
  <script>
    //Votre code JavaScript
  </script>
  ...
  <script src="module2.js"></script>
</body>
</html>
```



L'objet racine d'une page web

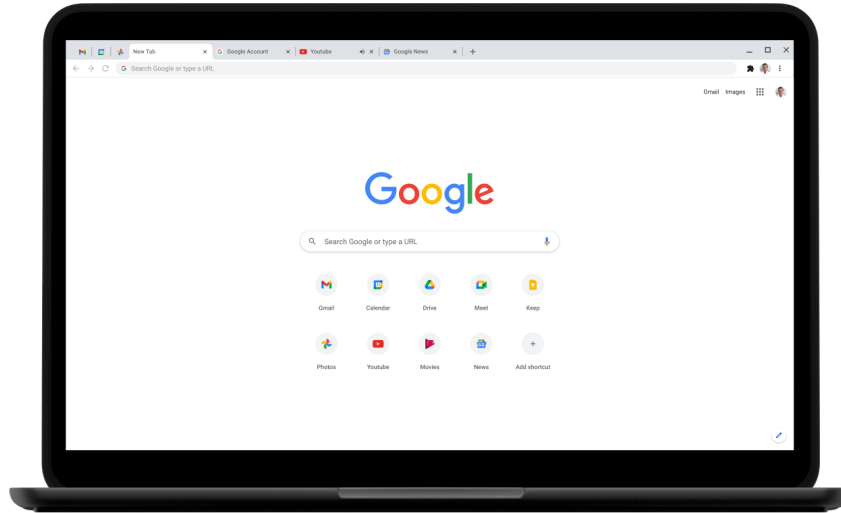
Document Object Model



Les événements en JavaScript

Document Object Model

⚡ click, dblclick



⚡ keydown, keypress, keyup

⚡ mouseenter, mousemove..., wheel

⚡ load

⚡ select, change

⚡ ...



S'abonner à un événement

Document Object Model

- Méthode déclarative :

```
<input type="button" id="idAjouter" value="Ajouter" onclick="ajouterClique();">
```

- Méthode programmatique :

```
document.getElementById("idAjouter").addEventListener("click", ajouterClique);
```





Programmation orientée objet en JavaScript

Créer un objet

Les objets en JavaScript

- Déclaration
- Manipulation `var objet = {cle: "valeur", cle2: 256} ;`

- Structure dynamique

```
objet.cle  
objet['cle2']
```

- Tableau d'objets

```
objet.cle3=10;
```

```
var objets = [{}, {}, ...];
```



Classe

Programmation orientée objet en JavaScript

- Mot clé : **class**

```
class Sport{  
  constructor(nom, description){  
    this.nom = nom || "pas de nom";  
    this.description = description || "";  
  }  
  
  afficher() {  
    console.log(`${this.nom} ${this.description}`);  
  }  
}
```



Extends

Programmation orientée objet en JavaScript

- Mot clé d'héritage : **extends**

```
class SportDeCompétition extends Sport{  
  constructor(nom, description, niveau) {  
    super(nom, description);  
    this.niveau = niveau || "";  
  }  
  
  afficher() {  
    console.log(`${this.nom} ${this.description} ${this.niveau}`);  
  }  
}
```



Les fonctions anonymes 1/2

- Déclaration

```
var uneFonction = function ([unParametre,...]) {  
    //le code de la fonction  
};
```

- Appel

```
uneFonction ([valeurParametre,...]);
```



Les fonctions anonymes 2/2

- Déclaration et appel consécutif

```
(function([unParametre,...]) {  
    //le code de la fonction  
}) ([valeurParametre,...]);
```



Le stockage local



LocalStorage

sessionStorage

```
setItem(cle, valeur)  
getItem(cle)  
removeItem(cle)  
clear()
```

