

instance d'un objet

```
let myObject = new Object();           appel du constructor

let myObject = {[définition des propriétés]}  syntaxe littérale

let myObject = new Object();           Exemple création
myObject = {                             d'un objet Object
  brand: new String("Brand"), type: 'modèle',
  ram: new Number(4), price: 500,
  upPrice: function(value) {
    return this.price += value; } }
```

Méthodes statiques

```
console.log(Object.keys(mon-   renvoie les clés des propriétés
Objet));                      définies

console.log(Object.values(m-   renvoie les valeurs des propriétés
onObjet));                    définies
```

Méthodes d'instances

```
console.log(monObjet.val-   renvoie la valeur primitive de l'objet
ueOf());                  (l'objet même)

console.log(monObjet.has-   renvoie un booléen si l'objet a la
OwnProperty("nom"));       propriété passée comme argument.
```

Création de propriétés via prototype

```
Object.prototype.message = "Hello   définition d'une propriété
World!";                       définie sur une valeur

Object.prototype.aff = function() {  définition d'une méthode
  console.log("C'est un objet !"); }
```

Utilisation de "this"

```
let test = new Object("Hello");      // Cela retournera "vrai"
Object.prototype.checkString = function() {
  if (typeof this.valueOf() === 'string') {
    return "vrai"; } else { return "faux"; } }
console.log(test.checkString());
```

Utilisation de "this" (cont)

```
let myObject2 = {                  // affiche "type : outil.
  type: "outil", matiere: "acier",   matière : acier"
  affObject: function() {
    console.log("type : " + this.type + ".
    matière : " + this.matiere); } }
myObject2.affObject();
```

Définition d'une classe

```
class Ordinateur {
  //déclaration des propriétés d'instance
  marque;
  ram;
  stockage;
  //définition de propriétés statiques
  static OBJETS_PRIS_EN_CHARGE = "ordinateurs";
  //définition de propriétés de classe
  type = "laptop";
  clavier = "azerty";
  //définition du constructeur
  constructor(marque, ram, stockage) {
    this._marque = marque;
    this._ram = ram;
  }
  // def de getters{
  get marque() {
    return this._marque;
  }
  // def de setters{
  set marque(value) {
    this._marque = value;
  }
  //définition de méthodes statiques
  static definition() {
    console.log("Les ordinateurs sont des machines automatisées
    permettant le traitement d'informations.");
  }
  //définition de méthodes d'instance
  addRam(value) {
    return this.ram += value;
  }
}
```