Cheatography

JavaScript POO Cheat Sheet

by batdams via cheatography.com/202369/cs/42955/

instance d'un objet

let myObject = new Object();	appel du constructor
let myObject = {/*definiton des propriétés*/}	syntaxe littérale

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

Méthodes statiques

console.log(Object.keys(monObjet));	renvoie les clés des propriétés définies
console.log(Object.values(monObjet));	renvoie les valeurs des propriétés définies

Méthodes d'instances

console.log(monObjet.valueOf());	renvoie la valeur primitive de l'objet (l'objet même)
console.log(monObjet.hasOwnProperty("nom"));	renvoie un booléen si l'objet a la propriété passée comme argument.

Création de propriétés via prototype

Object.prototype.message = "Hello World!";	définition d'une propriété définie sur une valeur
Object.prototype.aff = function() { console.log("C'est un objet !"); }	définition d'une méthode

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;
  }
}
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com