

Les classes

Les classe offres l'avantage de pouvoir configurer nôtre constructeur et toute les méthodes de classe sont mis dans le prototype de la classe automatiquement.

Donc les méthodes ne sont instancié qu'une seul fois et elle sont accessible depuis toute les instances de la classe.

```
class Person{

  // LE CONSTRUCTEUR DE LA CLASSE.
  constructor(name, age){
    this.name = name;
    this.age = age;
  }

  // METHODE DE LA CLASSE.
  present(){
    console.log("Hello my name is "+ this.name);
  }

}

let matt = new Person("Matt", 32);
let franck = new Person("Franck", 21);

console.log(matt);
console.log(franck);
```



```
▼ Person {name: "Matt", age: 32} ⓘ
  age: 32
  name: "Matt"
  __proto__:
    ► constructor: class Person
    ► present: f present()
    ► __proto__: Object

▼ Person {name: "Franck", age: 21} ⓘ
  age: 21
  name: "Franck"
  __proto__:
    ► constructor: class Person
    ► present: f present()
    ► __proto__: Object

> |
```

Héritage de classe.

JS script.js x

```
1  class Person{
2
3      constructor(name, age){
4          this.name = name;
5          this.age = age;
6      }
7
8      present(){
9          console.log(`Hello my name is ${this.name}`);
10     }
11 }
12
13 // THE CLASS HERO EXTENDS PERSON-----
14 class Hero extends Person{
15
16     constructor(name,age,power){
17         super(name,age);
18         this.power = power;
19     }
20
21     usePower(){
22         console.log(`I'am using my power : ${this.power}`);
23     }
24
25     presentHero(){
26         super.present();
27         console.log(`My power is : ${this.power}`);
28     }
29 }
30
31 let matt = new Person("Matt","32");
32 let superman = new Hero("Aquaman", 35, "Bubbling");
33
34
```