



La classe en JS

Création de classe

```
function Person(name) {  
  |   .....  
  |   this.name = name;  
  |  
  |  
}
```

```
const person1 = new Person;  
const person2 = new Person('Coco');
```

```
// Person { name: undefined }, Person { name: 'Coco' }  
console.log(person1, person2);
```

Comparaison d'instance

```
function Person(name) {
  .....
  this.name = name;
}

const person1 = new Person('Coco');
const person2 = new Person('Coco');

// Person { name: 'Coco' }, Person { name: 'Coco' }
console.log(person1, person2);

console.log(person1 == person2); // false
console.log(person1 === person2); // false
console.log(person1 instanceof Person); // true
```

Méthode d'instance

```
function Person(name) {  
  this.name = name;  
  
  this.sayHello = function() {  
    console.log(`Salut ${this.name} !`);  
  }  
}  
  
const person1 = new Person('Pedro');  
const person2 = new Person('Coco');  
  
person1.sayHello(); // Salut Pedro !  
person2.sayHello(); // Salut Coco !  
  
console.log(person1.sayHello === person2.sayHello); // false
```

Méthode partagée

```
function Person(name) {  
  |   this.name = name;  
}  
  
Person.prototype.sayHello = function() {  
  |   console.log(`Salut ${this.name} !`);  
}  
  
const person1 = new Person('Pedro');  
const person2 = new Person('Coco');  
  
person1.sayHello(); // Salut Pedro !  
person2.sayHello(); // Salut Coco !  
  
console.log(person1.sayHello === person2.sayHello); // true
```

Méthode de classe

```
function Person(name) {  
  .....  
  this.name = name;  
}
```

```
Person.getDefaultRole = function() {  
  console.log("Par défaut, l'utilisateur est enregistré comme anonyme");  
}
```

```
Person.getDefaultRole();
```

FIN