L’objet en JS

Objet littéral

JS permet la création d'objet sans classe → **forme littérale**

Dans ce cas, l'instance est la variable qui contient la référence vers l'objet

```
const contact = { // contact = instance
  name: 'Michael',
  age: 14,
  email: 'michael@mail.fr',
  man: true
};
```

Instance d’un objet littéral

Pour accéder à l’instance, on utilise le mot clé **this**

```
const contact = {
  name: 'Michael',
  sayHello: function() {
    console.log(this.name);
  }
};
```

Création d'une instance à partir de l'instance initiale

```
const contact = { name: 'Michael' };

const c1 = Object.create(contact);
const c2 = Object.create(contact);
console.log(c1.name, c2.name); // Michael, Michael
```

Accès aux valeurs d’un objet

```
const contact = ({
  name: 'Michael',
  "first-and-last-name": 'Michael CORNILLON',
  sayHello: function() {
    console.log(this['first-and-last-name']);
  }
});
```

Objet non manipulable

```
({
  name: 'Michael',
  sayHello: function() {
    console.log(this.name);
  }
}).sayHello();
```

Boucler sur un objet

```
for (let key in contact) console.log(key); // name, sayHello
try {
  for (let value of contact) console.log(value); // TypeError: contact is not iterable
}
catch (e) {
  console.log('Oups, an error !');
}
```

```
for (let index in Object.keys(contact)) console.log(index); // 0, 1
for (let key of Object.keys(contact)) console.log(key); // name, sayHello

for (let index in Object.values(contact)) console.log(index); // 0, 1
for (let value of Object.values(contact)) console.log(value); // Michael, [Function: sayHello]

for (let index in Object.entries(contact)) console.log(index); // 0, 1
for (let value of Object.entries(contact)) {
  console.log(value); // ['name', 'Michael'], ['sayHello', [Function: sayHello]]
}
```

Descripteur de propriété

```
const contact = { name: 'Michael', age: 56 };

const c1 = Object.create(contact, {
  name: {
    value: ['Jaco', 'Yes'],
    writable: true,
    configurable: false,
    enumerable: false
  },
  age: { value: 53, enumerable: true },
  email: { value: 'michael@mail.fr', enumerable: true },
});
Object.defineProperty(c1, 'name', { value: 'Fernando', writable: false });
const c2 = Object.create(contact);

c1.name = 'Franco'; // do not work !
console.log(Object.keys(c1)); // [ 'age', 'email' ]
console.log(Object.keys(Object.getPrototypeOf(c1))); // [ 'name', 'age' ]

console.log(c1.name, c2.name); // Fernando, Michael
```

Accesseur / mutateur

```
const book = {
  title: 'Il était une fois'
}

Object.defineProperty(book, 'description', {
  get: function () {
    return this.title;
  },
  // set: function (title) {
  //   this.title = title;
  // }
})

book.description = "Bonjour d'antan";
console.log(book.description); // Il était une fois
```

FIN