

Objet en Javascript

Les objets sont similaires aux arrays, sauf qu'au lieu d'utiliser des index pour accéder et modifier leurs données, vous accédez aux données des objets via ce que l'on appelle des propriétés.

Ils sont utiles pour stocker des données de manière structurée et peuvent représenter des objets du monde réel, comme un chat, une maison, un avion..... .

1) Créer un objet en JS (Propriété / valeur)

```
var myDog = {  
  "name" : "foxy",  
  "legs" : 4,  
  "tails" : 1,  
  "friends":["cat","lion"]  
};
```

2) Accéder à une propriété(On accède avec un "." ou des "[]" en cas d'espace.

```
var testObj = {  
  "hat": "ballcap",  
  "shirt": "jersey",  
  "shoes": "cleats"  
};  
  
var hatValue = testObj.hat;  
var shirtValue = testObj.shirt;
```

- 3) accéder à une propriété avec une variable(On créer une variable et on inclus la propriété que l'on souhaite accéder. Ensuite on créer une autre variable et on demande à ce que notre objet stock la variable contenant notre propriété).

```
// Setup
var testObj = {
  12: "Namath",
  16: "Montana",
  19: "Unitas"
};

// Only change code below this line

var playerNumber = 16;      // Change this line
var player = testObj[playerNumber]; // Change this line
console.log(player)
```

- 4) Modifier un objet

```
// Setup
var myDog = {
  "name": "Coder",
  "legs": 4,
  "tails": 1,
  "friends": ["freeCodeCamp Campers"]
};

myDog.name = "Happy Coder";
console.log(myDog); //Output Happy Coder
```

- 5) ajouter une propriété à notre objet

```
var myDog = {
  "name": "Happy Coder",
  "legs": 4,
  "tails": 1,
  "friends": ["freeCodeCamp Campers"]
};

myDog.bark = "woof"
```

- 6) supprimer une propriété

```
var myDog = {  
  "name": "Happy Coder",  
  "legs": 4,  
  "tails": 1,  
  "friends": ["freeCodeCamp Campers"],  
  "bark": "woof"  
};  
  
delete myDog.bark;
```

- 7) vérifier si une propriété existe ou pas. Ici on vérifie si dans la fonction obj se trouve la propriété checkProp.

```
function checkObj(obj, checkProp) {  
  if (obj.hasOwnProperty(checkProp)){  
    return obj[checkProp];  
  } else{  
    return "Not Found";  
  }  
}
```

- 8) Manipuler des objets complexes.

```
var myMusic = [  
  {  
    "artist": "Billy Joel",  
    "title": "Piano Man",  
    "release_year": 1973,  
    "formats": [  
      "CD",  
      "8T",  
      "LP"  
    ],  
    "gold": true  
  }  
];
```

```

    },
    {
      "artist" : "aerosmith",
      "title" : "dream on ",
      "release_year": 1956,
      "formats":["cd","youtube"],
      "gold": true
    }
  ];

```

9) accéder aux objets imbriqués

```

// Setup
var myStorage = {
  "car": {
    "inside": {
      "glove box": "maps",
      "passenger seat": "crumbs"
    },
    "outside": {
      "trunk": "jack"
    }
  }
};

var gloveBoxContents = myStorage.car.inside["glove
box"];
console.log(gloveBoxContents) //Output = maps

```

10) accès aux tableaux imbriqués

```
var myPlants = [  
  {  
    type: "flowers",  
    list: [  
      "rose",  
      "tulip",  
      "dandelion"  
    ]  
  },  
  {  
    type: "trees",  
    list: [  
      "fir",  
      "pine",  
      "birch"  
    ]  
  }  
];  
  
var secondTree =myPlants[1].list[1];
```