

Notions

- ▶ **Objet JavaScript** : les objets JavaScript permettent de manipuler des informations sous forme de propriétés. Ils peuvent également contenir des fonctions.
- ▶ **this** : le mot clef « this » permet d'accéder à une propriété ou une fonction à l'intérieur même d'un objet.
- ▶ Il est possible d'ajouter dynamiquement des propriétés à un objet
- ▶ Pour accéder aux propriétés d'un objet, deux syntaxes sont possibles : objet[« propriété »] ou objet.proprete

Exemples

```
let personne = {
  nom : "matthieu",
  age : 31,

  retourneFormatageAge : function() {
    let maintenant = new Date();
    let annee = maintenant.getFullYear();
    return `j'ai ${this.age} ans - Année naissance : ${annee - this.age}`;
  },
  affichage : function() {
    console.log(`Mon nom est ${this.nom}, ${this.retourneFormatageAge()}`);
  },
  ajoutNombreAnnee : function(annee){
    this.age += annee;
  }
}

personne.ajoutNombreAnnee(10);
personne.affichage();
```

```
let perso = {
  nom: "Tya",
  prenom: "Milo",
}

perso.age = 30;
```

```
let perso = {
  nom: "Tya",
  prenom: "Milo"
}

console.log(perso.nom);
perso.nom = "Gaston";
console.log(perso["nom"]);
```

Notions

- **Assign(cible,source)** : permet de copier les propriétés d'un objet dans un autre. Si une propriété existe dans les deux objets, celle de la source sera prise en compte. **Cible** est l'objet qui reçoit les valeurs, **source** est l'objet qui envoie les valeurs
- **Keys(obj)** : permet de retourner les **noms** des propriétés d'un objet sous forme d'un tableau
- **Values(obj)** : permet de retourner les **valeurs** des propriétés d'un objet sous forme de tableau
- **Entries(obj)** : permet de retourner le couple [nom_propriété , valeur] d'un objet sous forme de tableau. Nous avons donc un tableau à 2 dimensions

Exemples

```
let perso = {  
  nom: "wawa",  
  classe: "Test",  
}  
  
let caracGuerrier = {  
  classe: "Guerrier",  
  force: 5,  
  agilité: 3,  
  intelligence: 2,  
}  
  
Object.assign(perso, caracGuerrier);  
console.log(perso);
```

Peut se faire
avec le Spread
opérateur

```
{ nom: 'wawa',  
  classe: 'Guerrier',  
  force: 5,  
  agilité: 3,  
  intelligence: 2 }
```

```
let perso = {  
  nom: "wawa",  
  classe: "Test",  
}  
  
let caracGuerrier = {  
  classe: "Guerrier",  
  force: 5,  
  agilité: 3,  
  intelligence: 2,  
}  
  
perso = {  
  ...perso,  
  ...caracGuerrier  
}
```

```
let perso = {  
  nom: "wawa",  
  classe: "Test",  
}  
  
const indexes = Object.keys(perso);  
const valeurs = Object.values(perso);  
const indexesValeurs = Object.entries(perso);  
  
console.log(indexes);  
console.log(valeurs);  
console.log(indexesValeurs);  
  
for (let [index, valeur] of indexesValeurs) {  
  console.log(`${index}: ${valeur}`);  
}
```

['nom', 'classe']

['wawa', 'Test']

[['nom', 'wawa'], ['classe', 'Test']]

nom: wawa
classe: Test