

Notions

- ▶ **Class** : le mot clef « class » permet de créer une classe (concept de la Programmation Orientée Objet - POO).
- ▶ **Classe (POO)** : une classe est un « moule » qui permet de définir la structure générale d'objets ayant les mêmes caractéristiques (attributs / fonctions)
- ▶ **Constructeur** : une classe peut-être instanciée (générée), en utilisant le mot clef « new »
- ▶ **this** : le mot clef « this » permet d'accéder à une propriété ou une fonction à l'intérieur même d'un objet.
- ▶ **Héritage** : une classe peut utiliser les caractéristiques d'une autre classe au travers de l'héritage, mis en œuvre avec le mot clef « extends »
- ▶ **super** : le mot clef « super » permet de faire référence à un attribut ou une fonction d'une classe mère dont une classe hérite.

Exemples

```
class Personne {
  constructor(nom,age){
    this.nom = nom;
    this.age = age;
  }
  anniversaire() {
    this.age++;
  }
  affichagePerso(){
    return `Nom : ${this.nom} - Age : ${this.age}`;
  }
}

class Guerrier extends Personne{
  constructor(nom,age,pseudo){
    super(nom,age);
    this.pseudo = pseudo;
    this.force = 5;
    this.agilite = 3;
    this.intelligence = 2;
  }
  affichagePerso(){
    let txt = "*****\n";
    txt += super.affichagePerso();
    txt += ` - Pseudo : ${this.pseudo}\n`;
    txt += `Force : ${this.force} - Agilite : ${this.agilite} - Intelligence : ${this.intelligence}`;
    return txt;
  }
}

let perso1 = new Guerrier("Milo",25,"Papy");
console.log(perso1.affichagePerso());
perso1.anniversaire();
console.log(perso1.affichagePerso());
```

instanciation

```
*****
Nom : Milo - Age : 25 - Pseudo : Papy
Force : 5 - Agilite : 3 - Intelligence : 2
*****
Nom : Milo - Age : 26 - Pseudo : Papy
Force : 5 - Agilite : 3 - Intelligence : 2
```