Les fonctions constructeur

Fonction constructeur d'objet sans prototype, l'inconvénient c'est que a chaque fois que l'on créer une instance, on instancie également la fonction this.present()

```
function Personne(name, age){
  this.name = name
  this.age = age
  this.present = () => {console.log("Hello my name is "+ this.name)}
}


let matt = new Personne("Matt", 32);
let franck = new Personne("Franck", 21);

console.log(matt);
console.log(franck);
```

Fonction constructeur d'objet avec prototype, l'avantage c'est que la méthode this.present() est créer qu'une seul fois dans le prototype de l'objet Person et est donc utilisable par toute les instances de celle-ci.

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

Person.prototype.present =function(){
  console.log("Hello my name is "+ this.name);
}


let matt = new Person("Matt", 32);
let franck = new Person("Franck", 21);

console.log(matt);
console.log(franck);
```