Initialization: *constructor()*

`constructor(param1, param2, ...)` is a special method in the body of a class that initializes the instance, it is called automatically when a class is initiated and it has to have the exact name **`constructor`** ,

In the following example the constructor sets the initial value of the field `name` :

```
 class User {
  constructor(name) {
    this.name = name;
  }
}
```

`User` 's constructor has one parameter `name` , which is used to set the initial value of the field `this.name` .

The arguments used to instantiate the class becomes the parameters of the constructor:

```

class User {
  constructor(name) {
   // Here name will be "Jon Snow"
    this.name = name;
  }
}

const user = new User("Jon Snow");
console.log(user.name)
```

```
 // Output
Jon Snow
```

`name` parameter inside the constructor has the value `"Jon Snow"` .

- ### Let's have another example

```
 class Country {
  constructor(name, yearFounded) {
    this.name = name
    this.yearFounded = yearFounded
  }
}

const america = new Country("The United States of America", 1776)

console.log(america.name);
console.log(america.yearFounded);
```

```
 // Output
The United States of America
1776
```

Note: If you don't define a constructor for the class, a **default** one is created. The default constructor is an empty function, which doesn't modify the instance.