

# CONSTRUCTOR FUNCTIONS

---

The regular `{...}` syntax allows us to create an object

But often we need to create many similar objects, like multiple users or menu items and so on

This can be done using `constructor functions` and the "new" operator



purpose: to create objects

# CONSTRUCTOR FUNCTIONS

---

The regular `{...}` syntax allows us to create an object

But often we need to create many similar objects, like multiple users or menu items and so on

This can be done using `constructor functions` and the "new" operator



purpose: to create objects

# CONSTRUCTOR FUNCTIONS

---

Technically, any function can be a constructor function

But there are a few nuances when working with constructor functions. The 2 most important ones are:

1. Good practice to define your function with a **capital letter**
2. You should generally require the **new** keyword to be used when executing the constructor function

Why use the new keyword?

# CONSTRUCTOR FUNCTIONS

---

Basically the **new** keyword does a few things for us

1. Creates a new empty object and assigns it to **this**
2. Assigns properties to **this**
3. Returns the **this** object

# CONSTRUCTOR FUNCTIONS

---

Basically the **new** keyword does a few things for us

1. Creates a new empty object and assigns it to **this**

```
this = {};
```

2. Assigns properties to **this**

```
this.name = "Wally"  
this.admin = false
```

3. Returns the **this** object

```
return this
```

We then put this entire code block into a variable

```
let user1 = {name: "Wally", admin: false}
```