

CALLBACKS

What are callbacks?

A callback is a function that is to be executed after another function (normally asynchronous) has finished executing—hence the name ‘call back’.

*A way for one part of our program to
communicate with another part, that it
has finished running.*

Example

Example

We have 2 functions

Example

We have 2 functions

Function 1 "getLanguage" 🧐

Example

We have 2 functions

Function 1 "getLanguage" 🧐

Function 2 "greeter" 😊

Function 1 "getLanguage" 🤖 - RUNS

Function 1 "getLanguage" 🤖 - RUNS

.....

Function 1 "getLanguage" 🧐 - RUNS

.....COMPLETES & CALLS

Function 2 "greeter" 😊

```
1 function getLanguage() {  
2 }  
3  
4 function greeter() {  
5 }
```

```
1 function getLanguage(callback) {  
2 }  
3  
4 function greeter() {  
5 }  
6  
7 getLanguage();
```

```
1 function getLanguage(callback) {  
2     callback();  
3 }  
4  
5 function greeter() {  
6 }  
7  
8 getLanguage(greeter);
```

Why

```
greeter
```

instead of

```
greeter()
```

?

```
1 function getLanguage(callback) {  
2     console.log("running getLanguage");  
3     callback();  
4 }  
5  
6 function greeter() {  
7     console.log("running greeter");  
8 }  
9  
10 getLanguage(greeter);
```


You can pass in information into a callback

```
1 function getLanguage(callback) {  
2     console.log("running getLanguage");  
3  
4     const language = "ENGLISH";  
5  
6     callback(language);  
7 }  
8  
9 function greeter(language) {  
10     console.log("running greeter");  
11  
12     if(language === "ENGLISH") {  
13         console.log("Hello!");  
14     }  
15 }
```

Why use callbacks?

- Some code might take longer to run - for example, if you need to contact a server
- Allows one part of the program to "wait" for another part of our program
- Behaviour is asynchronous, meaning we can continue to execute other code (non-blocking)