



# WHAT IS CALLBACK

MANI TAHRIRI

# WHAT IS CALLBACK

- In JavaScript, function is an object. Because of that, each function can hold another function as argument. As the result, we can write nested function. The function that have another function inside is called higher-order function this is similar to parent-child relationship.

# WHEN TO USE CALLBACK

- JavaScript is an *event-driven language*, meaning that the list of function will be execute one after another. This characteristic of JavaScript is not a problem unless we want to work with another API.

```
function a(){
  // create waiting time for function A
  setTimeout( function(){
    console.log("This is function A");
  }, 2000 );
}

function b(){
  console.log("This is function B");
}

a();
b();
```

This is function B	<a href="#">callback.js:20</a>
This is function A → execute with delay	<a href="#">callback.js:15</a>

>



```
function a(){  
  // create waiting time for function A  
  setTimeout( function(){  
    console.log("This is function A");  
  }, 2000 );  
}
```

→ callback

→ callback();

```
function b(){  
  console.log("This is function B");  
}
```

a();

~~b();~~

→ a(b);

```
// THIS EXAMPLE WILL RUN FUNCTION ONE FIRST  
function a(callback){  
  setTimeout( function(){  
    console.log("This is function A");  
    callback();  
  }, 2000 );  
}  
  
function b(){  
  console.log("This is function B");  
}  
  
a(b);|
```

# HOW TO PASS MORE THAN ONE ARGUMENT

```
function StartEngine(engine, callback) {  
  setTimeout( function(){  
    console.log(`Starting the ${engine} now.`);  
    callback();  
  }, 2000);}  
  
function setGear(){  
  setTimeout( function(){  
    console.log('go to first gear');  
  }, 1000 );  
}  
StartEngine('BMW', setGear);
```



# BLOG POST EXAMPLE

```
const posts = [
  {title: 'Post One', body: 'This is post one'},
  {title: 'Post Two', body: 'This is post two'}
];

function createPost(post, callback) {
  setTimeout(function() {
    posts.push(post);
    callback();
  }, 2000);
}

function getPosts() {
  setTimeout(function() {
    let output = '';
    posts.forEach(function(post){
      output += `<li>${post.title}</li>`;
    });
    document.body.innerHTML = output;
  }, 1000);
}

createPost({title: 'Post Three', body: 'This is post three'}, getPosts);
```