

Lesson 05 Demo 02

Working with Functions and Arrays

Objective: To create functions and work with arrays by performing operations such as pushing, removing, and updating array elements

Tools required: Visual Studio Code and Node Package Manager

Prerequisites: JavaScript and knowledge of basic Linux and NPM commands

Steps to be followed:

1. Declare a simple function in JavaScript
2. Store the function reference in a variable
3. Declare the first-class function using the arrow operator
4. Declare a simple array in JavaScript
5. Push a new value to an array
6. Remove an element from an array
7. Update the value at a particular index in an array

Step 1: Declare a simple function in JavaScript

- 1.1 Create a function named `greetUser()` with an input argument in the `index.js` file

```
function greetUser(username) {  
  console.log(">>> Hello", username, '\b, How are you?');  
}
```



```
JS index.js  
JS index.js > ...  
1  function greetUser(username) {  
2    console.log(">>> Hello", username, '\b, How are you?');  
3  }  
4  
5  
6
```

1.2 Execute the function by passing a string name: `greetUser("Fionna")`

```
JS index.js > ...
1  function greetUser(username) {
2    console.log(">>> Hello", username, '\b, How are you?');
3  }
4
5  greetUser("Fionna")
6
```

1.3 Open the terminal window and execute the code using the command `node index.js` inside the project directory to view the output:

```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
>>> Hello Fionna, How are you?
```

Step 2: Store the function reference in a variable

2.1 Use the following to define a function named `greetUser()`, which takes a `username` parameter and logs a greeting message to the console:

```
const greetUser = function(username) {
  console.log(">>> Hello", username, '\b, How are you?');
}
```

```
JS index.js • {} launch.json
JS index.js > ...
1  const greetUser = function(username) {
2    console.log(">>> Hello", username, '\b, How are you?');
3  }
4
5
6
```

2.2 Use the variable name to execute the function by passing a suitable string as input: `greetUser("Fionna")`

```
JS index.js • {} launch.json
JS index.js > ...
1  const greetUser = function(username) {
2    console.log(">>> Hello", username, '\b, How are you?');
3  }
4
5  greetUser("Fionna")
6
```

- 2.3 Open a terminal window and execute the command **node index.js** inside the project directory to view the output:

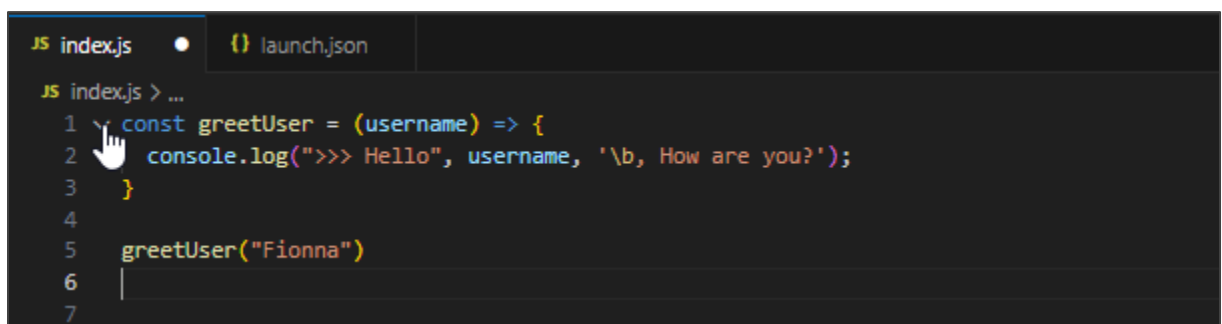
```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js  
>>> Hello Fiona, How are you?
```

Step 3: Declare the first-class function using the arrow operator

- 3.1 Use the following code to declare the first-class function using the arrow operators:

```
const greetUser = (username) => {  
  console.log(">>> Hello", username, '\b, How are you?');  
}
```

```
greetUser("Fionna")
```



```
JS index.js  {} launch.json  
JS index.js > ...  
1  const greetUser = (username) => {  
2    console.log(">>> Hello", username, '\b, How are you?');  
3  }  
4  
5  greetUser("Fionna")  
6  
7
```


- 3.2 Open a terminal window and execute the command **node index.js** inside the project directory to view the output:

```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js  
>>> Hello Fiona, How are you?
```

Step 4: Declare a simple array in JavaScript

4.1 Declare a simple array using the following code:

```
let nameArray = ["Fionna", "Jack", "John"]  
console.log(nameArray);
```

A screenshot of a code editor with a dark theme. The top bar shows two tabs: 'index.js' (active) and 'launch.json'. The editor content shows the following code:

```
JS index.js > ...  
1 let nameArray = ["Fionna", "Jack", "John"]  
2 console.log(nameArray);  
3
```

4.2 Open a terminal window and execute the command **node index.js** inside the project directory to view the output:

A screenshot of a terminal window. The prompt is 'demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4\$'. The command 'node index.js' has been executed, and the output is '['Fionna', 'Jack', 'John']'.

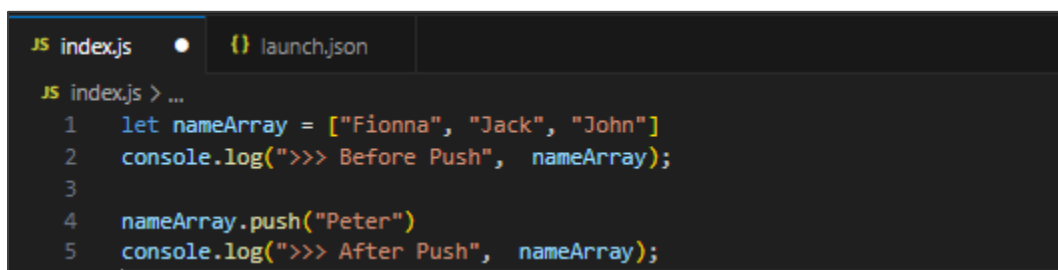
```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js  
[ 'Fionna', 'Jack', 'John' ]
```

Step 5: Push a new value to an array

5.1 Use the following code to declare an array and push a new value in it:

```
let nameArray = ["Fionna", "Jack", "John"]  
console.log(">>> Before Push", nameArray);
```

```
nameArray.push("Peter")  
console.log(">>> After Push", nameArray);
```

A screenshot of a code editor with a dark theme. The top bar shows two tabs: 'index.js' (active) and 'launch.json'. The editor content shows the following code:

```
JS index.js > ...  
1 let nameArray = ["Fionna", "Jack", "John"]  
2 console.log(">>> Before Push", nameArray);  
3  
4 nameArray.push("Peter")  
5 console.log(">>> After Push", nameArray);
```

5.2 Open a terminal window and execute the command **node index.js** inside the project directory to view the output:

```
demopython1yopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
>>> Before Push [ 'Fionna', 'Jack', 'John' ]
>>> After Push [ 'Fionna', 'Jack', 'John', 'Peter' ]
```

Step 6: Remove an element from an array

6.1 Use the following code to declare an array with some values and remove a value from it:

```
JS index.js  {} launch.json
JS index.js > ...
1  let nameArray = ["Fionna", "Jack", "John", "Peter"]
2  console.log(">>> Before Pop", nameArray);
3
4  nameArray.pop("Jack")
5  console.log(">>> After Pop", nameArray);
6
```

6.2 Open a terminal window and execute the command **node index.js** inside the project directory to view the output:

```
demopython1yopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
>>> Before Pop [ 'Fionna', 'Jack', 'John', 'Peter' ]
>>> After Pop [ 'Fionna', 'Jack', 'John' ]
```

Step 7: Update the value at a particular index in an array

7.1 Use the following code to declare an array with some values and update a value at a particular index:

```
let nameArray = ["Fionna", "Jack", "John", "Peter"]  
console.log(">>> Before Updation", nameArray);
```

```
nameArray[1] = "Jack Petersons"  
console.log(">>> After Updation", nameArray);
```

A screenshot of a code editor with two tabs: 'index.js' and 'launch.json'. The 'index.js' tab is active, showing the following code:

```
JS index.js > ...  
1 let nameArray = ["Fionna", "Jack", "John", "Peter"]  
2 console.log(">>> Before Updation", nameArray);  
3  
4 nameArray[1] = "Jack Petersons"  
5 console.log(">>> After Updation", nameArray);  
6
```

7.2 Open a terminal window and execute the command **node index.js** inside the project directory to view the output:

A screenshot of a terminal window with the following text:

```
demopython1yopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js  
>>> Before Updation [ 'Fionna', 'Jack', 'John', 'Peter' ]  
>>> After Updation [ 'Fionna', 'Jack Petersons', 'John', 'Peter' ]
```

By following these steps, you have successfully created functions and worked with arrays by performing operations such as pushing, removing, and updating array elements.