

Functions



David Tucker

CTO Consultant

@_davidtucker_ | davidtucker.net

Function Declaration

index.js

```
// Function declaration
```

Function Declaration

index.js

```
// Function declaration  
function isValidDepartment (input) {  
  
}
```

Function Declaration

index.js

```
// Function declaration
function isValidDepartment (input) {
  const departmentNames = [ "Marketing",
    "Engineering", "Human Resources" ];
  return departmentNames.includes(input);
}
```

```
// Using function
isValidDepartment("Engineering"); // true
isValidDepartment("Finance"); // false
```



Function Declarations



Function Expressions



Arrow Functions

Arrow Functions

An arrow function expression is a compact alternative to a traditional function expression, with some semantic differences and deliberate limitations in usage.

Arrow Function Syntax

index.js

```
// Function expression
```

```
const isDavid = function (name) {  
    return (name === "David");  
}
```

```
// Multiline arrow function
```

```
const isDavid = (name) => {  
    return (name === "David");  
}
```

```
// Single line arrow function with a parameter
```

```
const isDavid = (name) => name === "David";
```

Arrow Function Syntax

index.js

```
// Single line arrow function with no parameters
const lineBreak = () => console.log("");

// Arrow function with multiple parameters
const logArray = (item, index) => {
  console.log(`${index}: ${item}`);
}
```



| Iterating Over Arrays with Functions



Filtering Arrays with Functions



Recursion

Recursion

In programming recursion is when you call a function within itself. In doing this, an infinite loop can occur, so this must be done in a way where a function will not call itself continually.



Higher-order Functions

Higher-order Functions

In JavaScript, a higher-order function takes one or more parameters and returns another function as its return value.