

Functions



Overview

```
1  /*
2    - What is a function?
3    - Why use functions?
4    - Creating a function
5    - Calling a function
6    - Parameters and arguments
7    - Returning values
8    - Functions and TDD
9    - Looking ahead
10 */
```



What is a function?

```
1  /* a function is a reusable block of code */
2
3  /* functions work sort of like equations: you can
4     give them inputs and they can produce outputs */
5
6
7
8
9
10
```



Why use functions?

```
1  /* because we can run the same block of code whenever
2     we want, functions are an important part of
3     control flow */
4
5  /* it's often helpful in larger programs to use
6     multiple smaller, simple functions than have one
7     monolithic block of code */
8
9  /* breaking code into functions makes the code easier
10     to test */
```



Creating a function

```
1  /* A function declaration includes at least:
2     - function keyword
3     - function name
4     - parens
5     - line(s) of code between curly braces */
6  function imAFunction() {
7      console.log('I am inside a function');
8  }
9
10
```



Creating a function

```
1 // Notice the code inside our function is not running
2 function imAFunction() {
3     console.log('I am inside a function');
4 }
5
6 // imAFunction was declared above but never invoked
7
8
9
10
```



Invoking a function

```
1  // Use parentheses to invoke a function
2
3  function imAFunction() {
4      console.log('I am inside a function');
5  }
6  imAFunction();
7
8
9
10
```



before the function
I am inside a function
after the function

Invoking a function

```
1 // When called, the code inside the function will run
2
3 function imAFunction() {
4     console.log('I am inside a function');
5 }
6
7 console.log('before the function');
8 imAFunction();
9 console.log('after the function');
10
```




Parameters and arguments

```
1  /* An important feature of functions is that they
2     can receive inputs */
3
4  /* We use parameters when we define a function as
5     placeholders for inputs */
6
7  function hello(name) {
8     console.log('Hello', name);
9  }
10
```



Parameters and arguments

```
1  /* Arguments are the values we pass into a function
2     as inputs when we invoke a function. */
3
4  /* Inside the function, the arguments are assigned
5     to the parameters we defined when we declared
6     the function */
7
8
9
10
```



Parameters and arguments

```
1  // name is a parameter
2
3  function hello(name) {
4      console.log('Hello', name);
5  }
6
7  hello('Sadie'); // 'Sadie' is an argument
8
9
10
```



Hello David
Hello Nimit
Hello world

Parameters and arguments

```
1  // we can call functions multiple times
2  function hello(name) {
3      console.log('Hello', name);
4  }
5
6  hello('David');
7  hello('Nimit');
8
9  let notAName = 'world';
10 hello(notAName);
11
```



Parameters and arguments

```
1 // functions can have more than one argument
2
3 function hello(firstName, lastName) {
4     console.log('Hello', firstName, lastName);
5 }
6
7 // pass arguments in the same order as the parameters
8 hello('Gaby', 'Medina');
9
10
```



Parameters and arguments

```
1  // what happens if we don't pass in an argument?
2
3  function hello(firstName, lastName) {
4      console.log('Hello', firstName, lastName);
5  }
6
7  hello('Gaby');
8
9
10
```



Parameters and arguments

```
1  // what happens if we pass in too many arguments?
2
3  function hello(firstName, lastName) {
4      console.log('Hello', firstName, lastName);
5  }
6
7  hello('Gaby', 'Medina', 'the Great');
8
9
10
```



Returning values

```
1  /* functions take inputs and return outputs,  
2     remember? */  
3  
4  /* we can use a return statement to output a value  
5     out of the function */  
6  
7  
8  
9  
10
```




Returning values

```
1  function getANum() {  
2      return 10;  
3  }  
4  
5  let aNum = getANum();  
6  console.log(aNum);  
7  
8  
9  
10
```



Returning values

```
1  /* the expression to the right of the return keyword
2     will be evaluated before it is returned */
3  function getFullName(firstName, lastName) {
4     return firstName + ' ' + lastName;
5  }
6
7  let first = 'Alan';
8  let last = 'Turing';
9  let fullName = getFullName(first, last);
10 console.log(fullName);
11
```



Returning values

```
1  /* execution of a function stops when it gets to a
2     return statement */
3
4  function getFullName(firstName, lastName) {
5     return firstName + ' ' + lastName;
6     console.log('I will never run');
7 }
8
9 let fullName = getFullName('Anita', 'Borg');
10 console.log(fullName);
```



fullName: Julia Liuson
returnedValue: undefined

Returning values

```
1  // what if we forget to return?
2
3  function getFullName(firstName, lastName) {
4      let fullName = firstName + ' ' + lastName;
5      console.log('fullName:', fullName);
6  }
7
8  let returnedValue = getFullName('Julia', 'Liuson');
9  console.log('returnedValue:', returnedValue);
10
```



Recap

```
1  /*
2    - What is a function?
3    - Why use functions?
4    - Creating a function
5    - Calling a function
6    - Parameters and arguments
7    - Returning values
8    - Functions and TDD
9    - Looking ahead (next few slides)
10 */
```



Looking Ahead

```
1  /* this is our last pre-work lecture together! */
2
3  /* there is a sixth set of practice problems for you to finish after you
4     complete the problems for this lecture */
5
6  /* the sixth set of problems will review everything we've covered up to
7     this point */
8
9
10
```



Looking Ahead

```
1  /* this is our last pre-work lecture together! */
2
3  /* there is a sixth set of practice problems for you to finish after you
4     complete the problems for this lecture */
5
6  /* the sixth set of problems will review everything we've covered up to
7     this point */
8
9
10
```



Looking Ahead

```
1  /* if you're continuing with async BCP, we'll see you throughout the rest
2     of the course! */
3
4
5
6
7
8
9
10
```