



- ✓ 1. Intro to Functions
- ✓ 2. Function Example
- ✓ 3. Declaring Functions
- ✓ 4. Function Recap
- ✓ 5. Quiz: Laugh it Off 1 (5-1)
- ✓ 6. Quiz: Laugh it Off 2 (5-2)
- ✓ 7. Return Values
- ✓ 8. Using Return Values
- ✓ 9. Scope
- ✓ 10. Scope Example
- ✓ 11. Shadowing
- ✓ 12. Global Variables
- ✓ 13. Scope Recap
- ✓ 14. Hoisting
- ✓ 15. Hoisting Recap
- ✓ 16. Quiz: Build a Triangle (5-3)
- ✓ 17. Function Expressions
- ✓ 18. Patterns with Function Expressions
- ✓ 19. Function Expression Recap
- 20. Quiz: Laugh (5-4)
- 21. Quiz: Cry (5-5)
- 22. Quiz: Inline (5-6)
- 23. Lesson 5 Summary

### What you've learned so far:

**Function Expression:** When a function is assigned to a variable. The function can be named, or anonymous. Use the variable name to call a function defined in a function expression.

```
// anonymous function expression
var doSomething = function(y) {
  return y + 1;
};
```

```
// named function expression
var doSomething = function addOne(y) {
  return y + 1;
};
```

```
// for either of the definitions above, call the function like this:
doSomething(5);
```

Returns: 6

You can even pass a function into another function *inline*. This pattern is commonly used in JavaScript, and can be helpful streamlining your code.

```
// function declaration that takes in two arguments: a function for displaying
// a message, along with a name of a movie
function movies(messageFunction, name) {
  messageFunction(name);
}
```

```
// call the movies function, pass in the function and name of movie
movies(function displayFavorite(movieName) {
  console.log("My favorite movie is " + movieName);
}, "Finding Nemo");
```