Functions

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions

Function Declaration

```
function playSound() {
  // ...
}


assertEquals(typeof playSound, "function");
```

Declared directly with the ***function*** keyword - no use of ***var***

Function Expression

```
var addNumbers = function() {
  // ...
};
```

Assigned as the value of a declared variable

Anonymous Function

```
var addNumbers = function () {
    // ...
};

function playSound() {
    // ...
}

assertEquals(addNumbers.name, "");
assertEquals(playSound.name, "playSound");
```

Named Function Expression

```
var playSound = function playSound() {
  // ...
};


assertEquals(playSound.name, "playSound");
```

Named Function Expression

```
var doThing = function playSound() {
    // ...
};


assertEquals(doThing.name, "playSound");
```

The name is based on the ***function***, not the ***variable***

Implicit Return Values

```
function helloWorld() {
  return undefined;
}

assertEquals("undefined", typeof helloWorld());
```

Arguments

```
function sum(a, b) {
  return a + b;
}

assertEquals(sum(1, 2), 3);
```

Rest parameter

```
function sum(...args) {
  // args[0] == 1
  // args[1] == 2
  // args.length == 2
}

sum(1, 2);
```

Rest parameter

```
function sum(...args) {
  return args[0] + args[1];
}

assertEquals(sum(1, 2), 3);
```

Can be used to make **variadic functions**

(functions that accepts a variable number of arguments)

Variadic function

```
function sum(...args) {
  var s = 0;
  for (var i = 0; i < args.length; i++) {
    s += args[i];
  }

  return s;
}


assertEquals(sum(2, 3, 4, 5), 14);
```

Assign to variable

```
var sayHello = function () {};
```

Assign to object properties

```
var person = {
   sayHello: function () {}
};


person.sayGoodbye = function () {};
```

Pass as arguments

```
var sum = function (a, b) {
  return a + b;
};

var multiply = function (a, b) {
  return  a * b;
}

var manipulateNumbers = function (a, b, operat
  return operation(a, b);
}

manipulateNumbers(2, 3, sum);
// => 5

manipulateNumbers(2, 3, multiply);
// => 6
```

Look up dynamically

```
var sayHello = person.sayHello;

sayHello();
```

They have properties

```
var sum = function sum(a, b, c) { ... };

assertEquals(sum.length, 3);
assertEquals(sum.name, "sum");
```

Hoisting

```
/*
var doSomething = function doSomething(a) {
  // ...
};
*/

doSomething();

function doSomething(arg) {
   // ...
}
```

Hoisting

- Variable hoisting (only the declaration!)
- Function declaration (also the body)

Primitives passed by value

```
var num1 = 1;
var num2 = 2;

function doSomething(x, y) {
  x = 10;
  y = 20;
}

doSomething(num1, num2);

assertEquals(num1, 1);
assertEquals(num2, 2);
```

Objects passed as reference

```
function doSomething(obj) {
  obj.num = 100;
}

var myObj = { num: 5 };

doSomething(myObj);



assertEquals(myObj.num, 100)
```

All non-primitives behave like this!

Function Exercises

Canvas: exercises-functions.zip