

# JavaScript Introduction Part 2

---

Reference: Burdette, L. (2010). The JavaScript PocketGuide. PeachPit Press

## Objects

- A collection of values called properties
- Properties that are functions are called methods

<code>var half = {};</code>	Declaring and defining an empty object
<code>half.numerator = 1;</code>	Declaring and defining a property
<code>half.denominator = 2;</code>	
<code>half.decimal = function(){     return this.numerator / this.denominator; }</code>	Declaring and defining a method

## Loops

- Supports: while, for, do...while

## Conditional Logic

- Supports: if, if-else, if-else-if

## Exceptions

```
try {  
    // An exception may be thrown here  
} catch(exception){  
    // Handles the exception  
} finally {  
    // Always executes whether or not the exception was thrown  
}  
  
// How to throw an exception  
throw new Error("something bad happened");
```

## Logical Operators

- Logical Not: !
- Logical And: &&
- Logical Or: ||

## Ternary Expression

```
result = expression ? return this if true : return this if false
```

E.g.  
`value = someVar > 5 ? "D" : "d";`

# Writing JavaScript

---

## A Simple Tool for Testing Your JavaScript Code

Using coding jargon, we call the module being developed the system under test (SUT). The idea is to test as we develop. In some development processes, the test is written before the SUT.

### SutTester.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf8" />
  <title>SUT Tester</title>
</head>
<body>
  <header>
    <h1>SUT Tester</h1>
  </header>
  <script src="Sut.js"></script>
  <script src="SutTester.js"></script>
</body>
</html>
```

### Sut.js

```
function getAreaOfRectangle(length, width) {
  return length * width;
}
```

### SutTester.js

```
function assert(condition, message) {
  if (!condition) {
    throw (message || "Assertion") + " " + "Failed!";
  } else {
    console.log((message || "Test") + " " + "Passed");
  }
}

try {
  testGetAreaOfRectangle();
} catch (exception) {
  console.log(exception);
}

function testGetAreaOfRectangle() {
  var area = getAreaOfRectangle(2, 100);
  assert(area === 200, "Testing length 2 width 100");
}
```

### Console output from Browser:

```
Testing length 2 width 100 Passed
```

Here the SUT represents whatever module is under development. It should be renamed to match your module.

## JavaScript Introduction Part 2

---

### Simple Regular Expressions

See: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions)

- Regular expression literal: `/<pattern>/`, where `<pattern>` is the regular expression
- Constructor function: `new RegExp("<pattern>")`

```
var pattern = /version \d.\d/;
var string1 = "Some string";
var string2 = "version 1.2";
var string3 = "version 1";
var string4 = "here is version 1.4";
console.log(string1.search(pattern));
console.log(string2.search(pattern));
console.log(string3.search(pattern));
console.log(string4.search(pattern));
```

```
-1
0
-1
8
```