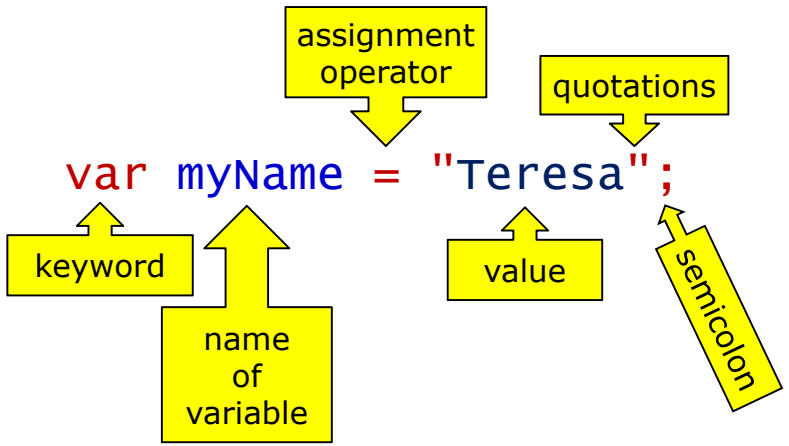
JavaScript and jQuery Course

Session 02

Variables

- an identifier that stores some kind of data
- var keyword (lowercase)
  - declares the variable
- = (assignment operator)
  - assigns a value
  - initializes the variable

Variables



Variables

```
var myName   = "Teresa";
var theTotal = 200;
```

- spaces do not matter around the equal sign
  - be consistent in how you use spaces
- quotations around a literal or string
  - but not around a number for calculating

Variables – How to name

- Begin with a letter, the $ character, or the underscore (_)
- Cannot begin with a number, but can contain a number
- Cannot contain spaces, punctuation, mathematical or logical operators
- Cannot be JavaScript reserved words

Variables – How to name

- camel-case syntax / underscore
- case sensitive

```
taxRate             tax_rate
calculateClick      calculate_click
emailAddress        email_address
firstName           first_name
futureValue         future_value
```


Variables – getting started

Declare

```
var myName;
```

Initialize (assign a value)

```
myName = "Teresa";
```


Variables – getting started

Declare and Initialize at same time

```
var myName = "Teresa";
```

Declare many variables at once

```
var myName, myAddr, myPhone;
```

comma

Pro's	Only have to type var once
Con's	Can be more difficult to find the definition of a variable

If you use this technique, **do not** type a list that extends beyond the right-side of the editor!

Declare and initialize many at once

```
var  myName = "Teresa", myAddr = "2 Main St",
     myPhone = "760-555-1212";
```

comma

Pro's	Only have to type var once
Con's	Can be more difficult to find the definition and initialization of a variable

If you use this technique, **do not** type a list that extends beyond the right-side of the editor!

Variable types

Examples of number values

```
15
-21
21.5
-124.82
-3.7e-9
```

Examples of string values

```
"JavaScript"
'String Data'
""
```

The two Boolean values

```
true
false
```


Variables

Scope

Variables exist where they are created

Global Variables - declared outside of a function

Can be used inside or outside of functions

Local Variables - declared inside a function

Can only be used in that function

Arithmetic Operators

+	Addition	`var a = 2 + 4;`
-	Subtraction	`var s = 6 - 2;`
*	Multiplication	`var m = 5 * 3;`
/	Division	`var d = 5 / 3;`
%	Modulus (remainder)	`var r = 43 % 10;` (result: 3)

Arithmetic Operators

Unary operators - means "only one factor"

++ Increment (adds one to the value)
-- Decrement (subtracts one from the value)

Examples

```
x++
n--
```


Shorthand Operators

Operator	Example	Equivalent to
=	`y = x + 3;`	
+=	`x += 3;`	`x = x + 3;`
-=	`x -= 3;`	`x = x - 3;`
*=	`x *= 3;`	`x = x * 3;`
/=	`x /= 3;`	`x = x / 3;`

Order of Precedence

Order	Operators	Direction
1	++	Left to right
2	--	Left to right
3	* / %	Left to right
4	+ -	Left to right

Examples of precedence and parentheses

```
3 + 4 * 5           // Result is 23
(3 + 4) * 5         // Result is 35

13 % 4 + 9          // Result is 10
13 % (4 + 9)        // Result is 0

1000 + 1000 * .05   // Result is 1050
1000 + (1000 * .05) // Result is still 1050
```

Concatenation

- Concatenate = Join
- + sign
  - will *add* numbers
  - will *concatenate* strings

Concatenation

```
var n = 1 + 1;                    // n is 2

var f = 3 + "fido";               // f is "3fido"

var ff1 = "fire" + "fox";         // ff1 is "firefox"

var ff2 = "fire" + " " + "fox";   // ff2 is "fire fox"
```


Type Conversions

`parseInt()` function

parses a string/number and returns an integer

`parseFloat()` function

parses a string/number and returns a floating point number (decmial)

`toString()` method

converts a number to a string

Type Conversion

`parseInt()` function

parses and returns an integer (whole number)

Example

```
var num = 123.4567;
num = parseInt(num);

alert(num);
```

Type Conversion

`toString()` method

converts a number to a string

Example

```
var num1 = 1;
num1 = num1.toString();

var num2 = 2;
num2 = num2.toString();

alert(num1 + num2);
```

Formatting Numbers

Do this last!
It is formatting

`.toFixed()` method of the `number` object

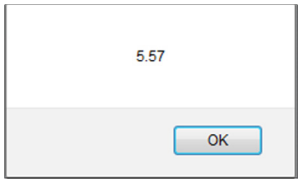
`number` object is part of the JavaScript language

Converts a number into a **string**, keeping the number of decimals specified in parentheses

Example

```
var num = 5.56789;
num = num.toFixed(2);

alert(num);
```



Testing a Variable

- Firebug Add-on for Firefox

```
console.log(variable_or_expression);
```

- Editor / Browser – How to Write code

```
alert(variable_or_expression);

alert("the result is: " + variable);
```


Math Object

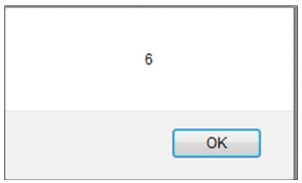
`Math` object is part of the JavaScript language
uppercase "M"

`Math.ceil()` – rounds upward to the nearest integer

Example

```
var num = 5.56789;
num = Math.ceil(num);

alert(num);
```



Math Object

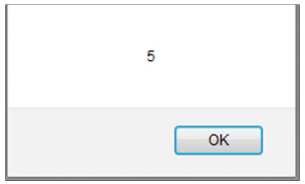
`Math.floor()`

rounds downward to the nearest integer

Example

```
var num = 5.56789;
num = Math.floor(num);

alert(num);
```



Math Object

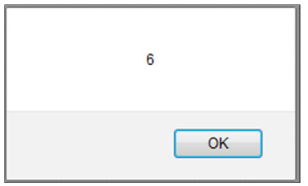
`Math.round()`

rounds to the nearest integer
can be up or down, whichever is closest

Example

```
var num = 5.56789;
num = Math.round(num);

alert(num);
```



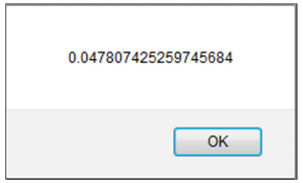
Math Object

`Math.random()`

generates a random decimal number
between 0 and 1 *but not including 1*

Example

```
var num = Math.random();
alert(num);
```



Math Object

`Math.floor()` and `Math.random()`

generate a whole number within a range

Example

```
var num = Math.floor(Math.random() * 11);
alert(num);
```

String Object

`length` property

returns the length of a string

Example

```
var myString = "Hello World!";
alert(myString.length);
```

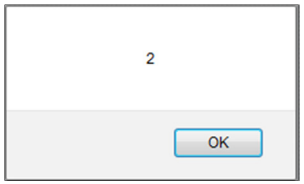
String Object

`indexOf()`

returns the position of the first occurrence of a specified value – begins at 0

Example

```
var myString = "Hello World!";
alert( myString.indexOf("l") );
```



String Object

`lastIndexOf()`

returns the position of the last occurrence of a specified value

`charAt()`

returns the character at the specified index

String Object

`toLowerCase()`

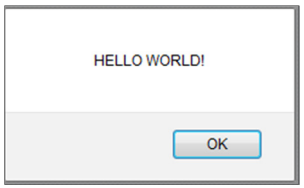
converts a string to lowercase letters

`toUpperCase()`

converts a string to uppercase letters

Example

```
var myString = "Hello World!";
alert( myString.toUpperCase() );
```



String Object

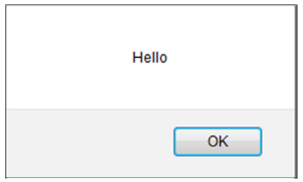
`slice()`

extracts a part of a string and returns a substring

takes the starting and ending position, or just the start position (will go to the end of string)

Example

```
var myString = "Hello World!";
alert( myString.slice(0,5) );
```



Escape Characters

Operator	Description
`\n`	Starts a new line in a string.
`\"`	Puts a double quotation mark in a string.
`\'`	Puts a single quotation mark in a string.

Example

```
alert("Hello \"Big\" World!");
```

Displays: `Hello "Big" World!`

Alternating Quotations

Example

```
alert( 'Hello "Big" World!' );
```

Displays: `Hello "Big" World!`