JavaScript Reference Sheet #1

Learn Teach Code

Comments

```
// A one-line comment begins with two slashes.

/* And this is a multi-line comment, for when you have lots of notes!
   Remember to use LOTS of comments to organize your code :)
*/
```

Variables * *Note: we will use **let**, not **var**. We will also use **const**, but more on that later ☺*

Declaring and setting variables:

```
let myString = "String of text";
let myNumber = 5;
let myBoolean = true;
```

You can also declare variables without values:

```
let myUndefinedVariable;
```

A useful shortcut: you can declare multiple variables at once, separated by commas:

```
let thingOne, thingTwo, thingThree;
let myName = "Snuffleupagus", myAge = 32;
```

You only need to use "let" when you FIRST define a variable. After that, you just use its name like this:

```
let myVariable = false;   // defining my variable
myVariable = true;        // changing its value
```

Console.log() * *Displays information in the console. Inputs can be any type of data!*

```
console.log("Read this message!");
console.log("The value of myVariable is: " + myVariable);
console.log(thingOne, thingTwo, thingThree); // multiple inputs are possible too!
```

Math Operators (the most common ones)

Gluing strings together (concatenation):	`"hello, " + "world!"`
Addition:	`2 + 30`
Subtraction:	`7 - 1`
Multiplication:	`10 * 3`
Division:	`25 / 100`
Order of operations:	`(25 + 5) * (23 - 13) / 3`
Incrementor:	`myNum++;` `myNum += 1;` `myNum = myNum + 1;`
Decrementor:	`myNum--;` `myNum -= 1;` `myNum = myNum - 1;`

Conditional (if and else) statements

```
let healthPoints = 15;

// An "if" statement. Here, the console will show nothing because the condition is false:
if ( healthPoints > 90 ) {
  console.log("You feel great!");
}

// An "if/else" statement. The console will show "Your health is low!"
if ( healthPoints > 90 ) {
  console.log("You feel great!");
} else {
  console.log("Your health is low!");
}

// An "if" + "else if" + "else" statement. The console will show "You're OK! "
if ( healthPoints > 90 ) {
  console.log("You feel great!");
} else if ( healthPoints > 10 ) {
  console.log("You're OK!");
} else {
  console.log("You're almost dead!");
}
```

Comparison and Logical Operators

Strict equality: `3 === 5`
Lazy equality: `3 == 5` ← ***Don't use it until you know what you're doing :)***

Strict inequality: `3 !== 5`
Lazy inequality: `3 != 5` ← ***Don't use it until you know what you're doing :)***

Greater than: `3 > 5`
Greater than or equal to: `3 >= 5`

Less than: `3 < 5`
Less than or equal to: `3 <= 5`

```
Logical AND:    true && false    // evaluates to false
Logical OR:     true || false    // evaluates to true
Logical NOT:    !true            // evaluates to false
```

Defining and Calling Your Own Functions

```
// Defining a function that takes TWO inputs and returns an output:
function greetByName(greeting, name) {
    return greeting + " to you, " + name;
}

// Calling the function, saving its output, then printing the output to the console:
let myMessage = greetByName("Hello", "Bobby");
console.log(myMessage);            // ****The console will show: "Hello to you, Bobby"
```

```
// A function with no inputs that returns an output:
function outputHello() {
    return "Hello there!";
}

// Printing the result of calling the function:
console.log( outputHello() );
```

Turning a website element into a JavaScript variable

You can assign a unique name to any element of a website using HTML like this:

```
<button id="start"> Start! </button>
```

Then you can save that particular HTML element as a JavaScript variable like this:

```
let startButton = document.getElementById("start");
```

*Notice that **"start"** -- the unique name we picked for this button on our website -- needs to **perfectly match** in your HTML and your JavaScript for this to work**!***

Now we can do things with this button in JavaScript using the variable "startButton".

***Note:** there are many other ways to access parts of your HTML using JavaScript, but to keep this beginner's class simple, we'll only be using this method for now.*

Change the text of an element

To change the text of an HTML element, you can do this:

```
startButton.textContent = "This will replace the text in the start button from earlier!";
```

** Note: the one exception is when you're working with *form elements* like an input box; for those, you need to use **.value** instead!

Get user input from a text box on your website

If you have a text input box in your HTML like this:

```
<input id="myinput">
```

And if you turned the input box into a JavaScript variable like this:

```
let myTextBox = document.getElementById("myInput");
```

Then you can save a string value from what the user wrote inside that text input box like this:

```
let userInput = myTextBox.value;
```

Change the color and style of an element

You can change how an element looks with JavaScript like this:

```
startButton.style,color = "red";
startButton.style,font = "italic 50px Helvetica";
```

Here's a useful website for custom color codes (called "hex codes"): color-hex.com/

Do something when a user clicks an element on your website

Example of telling ***the entire web page*** to ***start listening for clicks*** using an event listener:

```
document.addEventListener("click", myFunction);
```

Or, once you have an element saved as a JavaScript variable, you can tell it to ***listen for clicks*** like this:

```
startButton.addEventListener("click", myFunction);
```

You have include the name of a function that you want to run whenever the user clicks on the given element.

In the above example, the function named **myFunction** will run when the user clicks the **startButton** (whichever HTML element that refers to.)

** So you need to **define myFunction** somewhere in your code for this to work! See the previous section in this sheet on "Defining and Calling Your Own Functions".
