Topic/Title: Introduction to Javascript

Keywords/Questions:

Turning off JavaScript

Interpreted language

Compiled language

Web apps: Javascript

IOS apps: Swift

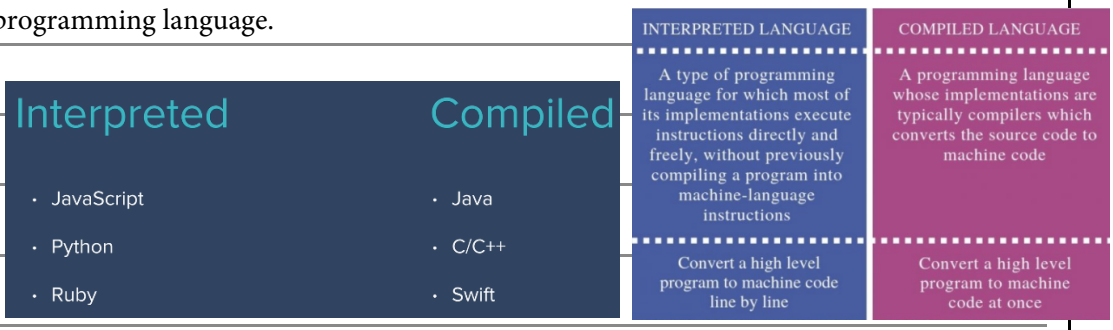
Android apps: Java

Notes:

-You can try turning off Javascript from your browser by going to settings>site settings> javascript. An advantage of this is that ads don't. A disadvantage is that some websites like YouTube and Netflix won't even load and websites that load lose some form of functionality. For example. when you type in twitter you won't see word count.

-The only reason why JavaScript was called JavaScript was because in the 90s the word Java was about so hot. Java and JavaScript have about as much in common as car and carpet.

JavaScript is an interpreted programming language whereas Java is a compiled programming language.



-In the olden days interpreted languages tend to be seen as almost like toy languages. They weren't so powerful. Now today modern JavaScript is used in all sorts of places, whereas traditionally it was a front end language that was meant to add some animations to your web site or allow a little bit of user interaction. But nowadays JavaScript can be seen in frameworks ranging from the front end to the back end and everything in between.

-Best languages for: -Web Apps:-JavaScript -Android Apps:-Java -IOS Apps:-Swift

Summary:

Turning off JavaScript:-settings >site/content settings>JavaScript>Don't allow

-this will cause some functionality issues -adds can't run

Javsscript:- is an interpreted language, the most popular programming language & used in both front & back end.

Interpreted language:-converted to machine code line by line

Compiled language:-converted to machine code as a whole

Best language for: - Web apps:JS IOS apps:Swift Android Apss:-Java

Topic/Title: JavaScript Alerts - Adding Behavior to Websites

Keywords/Questions:

alert("Hello");

alert('Hello");

alert(" Hello ");

Shift+Enter

inspect>console

inpect>sources>new snippet

Mdn>technologies>JS

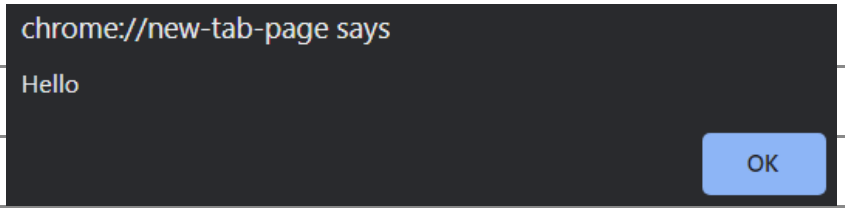
rwaldron/idiomatic.js

Notes:

alert("Hello");

-go to inspect then console then write the above code.

-this will create a pop up window on chrome.

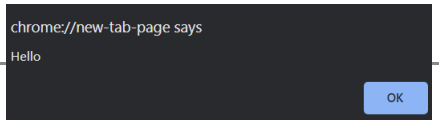


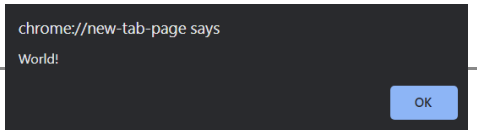
-If we touch OK the pop up will disappear.

Shift+Enter:- use it to move to the next line when you are writing more than one line inside console. For example

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

will result in 2 consecutive pop-ups:





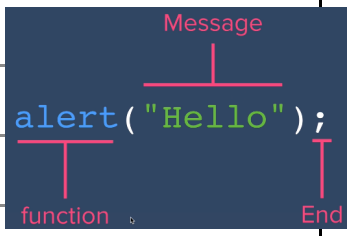
-when we hit ok on the first one the first will disappear and the second one will appear.

Inspect>Sources>New Snippet or ⋮ > More Tools>Developer Tools>Sources>New snippet

-after creating a new snippet you can write several lines of code as you do in atom and load it by writing Ctrl+Enter. You can name the snippet as you want but usually Angela names it as index.js.

Summary:

alert("Hello"); :-creates a pop up that says Hello. It disappears when you touch OK.



inspect>console:- writing javascript codes & running them in your browser

Shift+Enter:- moving to the next line when writing inside console

Inspect>Sources>New Snippet:- you can use it to write lots of lines of code and save it. It serves sort of like atom.

-you run your code by touching Ctrl+Enter

MDN>Technologies>JavaScript:-read it to know about javascript functions. search for functions in the search bar.

alert("hello"); =window.alert("hello")= alert('hello'); = alert(" hello "); :- all work but the standard one is alert("hello");

https://github.com/rwaldron/idiomatic.js/ :- tells us the standardized javascript style

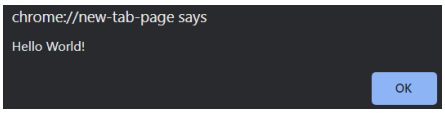
More Notes

Go to mdn>technologies>javascript

-You can read the whole documentation if you want to know more about javascript.

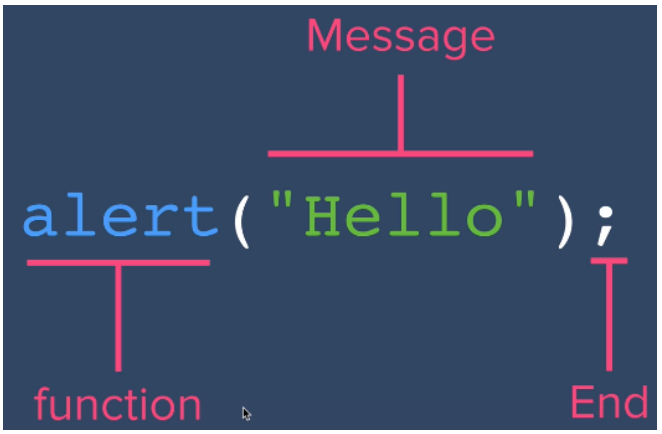
-For example if you wanna know about the alert(); function you can search alert in the search bar.

```
window.alert("Hello world!");
alert("Hello world!");
```



-both lines have the same result.

`window.alert()` instructs the browser to display a dialog with an optional message, and to wait until the user dismisses the dialog.



function:- is a keyword the browser knows.

-adds a little bit of behavior to the website.

message:- this is what we want to show up in the pop up.

-you can change it to anything you wish

End:- the semicolon denotes the end of your instruction.

```
alert(“Hello”);
```

-This is different from the above because in coding we use this(" ") quotation marks instead. This would be a problem when we type in word and copy paste our code. But, atom and snippet solve this problem automatically.

```
alert ( "Hello" ) ;
```

-Spacing doesn't matter. It doesn't change the functionality of your code. But, your it won't look good.

```
alert('Hello');
```

-Using single quotes is the same as using double quotes. But most programmers prefer double quotes.

https://github.com/rwaldron/idiomatic.js/

- document compiled by a lot of influential programmers
- it serves as a standard guide to how our code should look like aka its style
- at the current stage this document is a little bit too advanced, especially if you're learning programming for the first time. But after you complete this course, and as you start going on to build your own projects then this is a really handy reference guide to just look back at every so often to check to make sure that you are using the right syntax and the right structure.

All code in any code-base should look like a single person typed it, no matter how many people contributed.

Topic/Title: Data Types

Keywords/Questions:

String: "Hello"

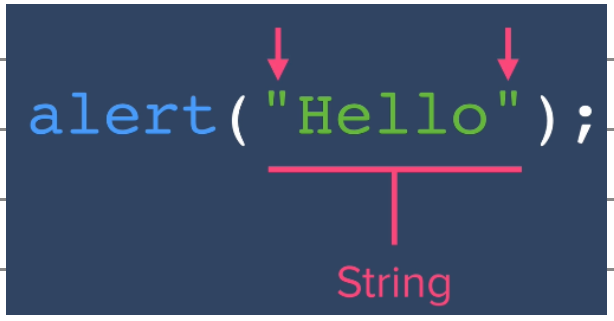
Number: 1

boolear: true or false

alert(2+3);

typeof();

Notes:



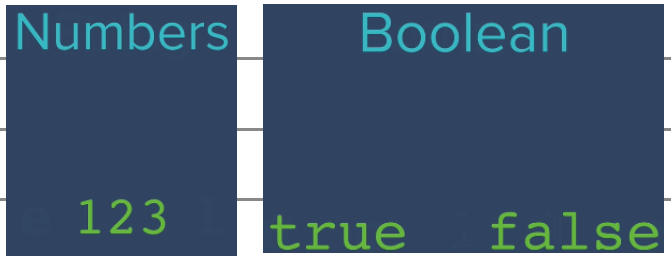
-Everything enclosed b/n quotation marks will be interpreted as a String by our browser.

"Hello" :- This is a string data type.

String :- text that's not meant to be interpreted as code

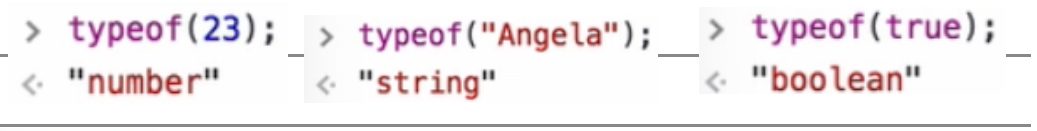
-just a string of characters.

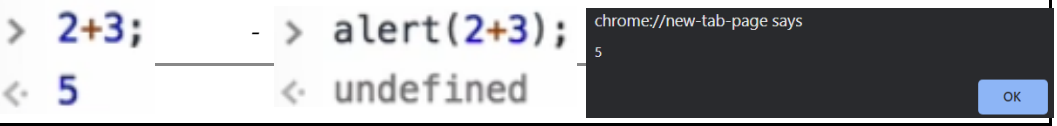
Other Data Types:-



-Variables

typeof() :-It will tell us the data type of whatever is in between the brackets.





Summary:

Data types:

-Numbers: are the no.s 1, 2, 3,...

-String: anything between " ". i.e:"Hello","hale",....

-Boolean: true & false

-variables

typeof():- tells us the type of our data type

alert(2+3); :- pops out 5. It pops out the computed result from our code.

Topic/Title: JavaScript Variables

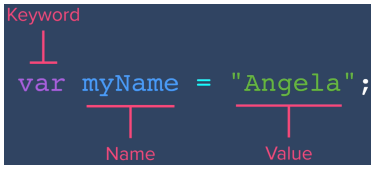
Keywords/Questions:

prompt();

var myName="Hale";

console.log();

var x=prompt("How're u?")

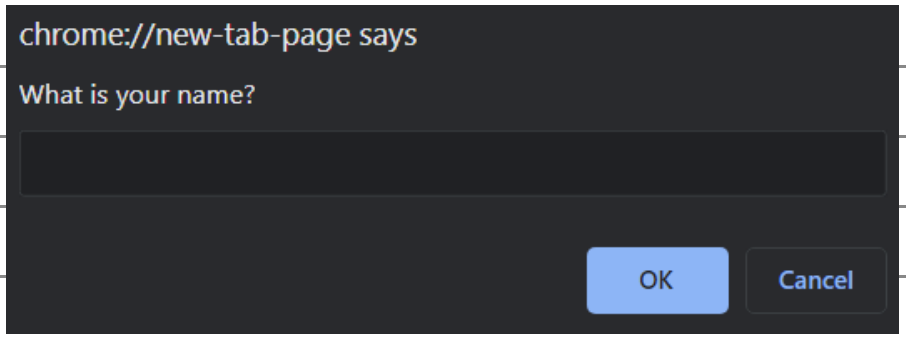


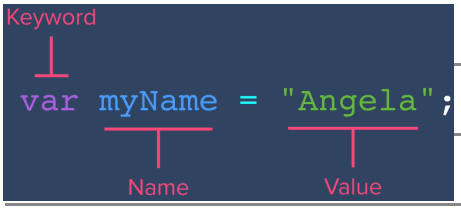
-you need to use var only for the first time.

Notes:

prompt(); :-used to ask for information by the use of pop ups.

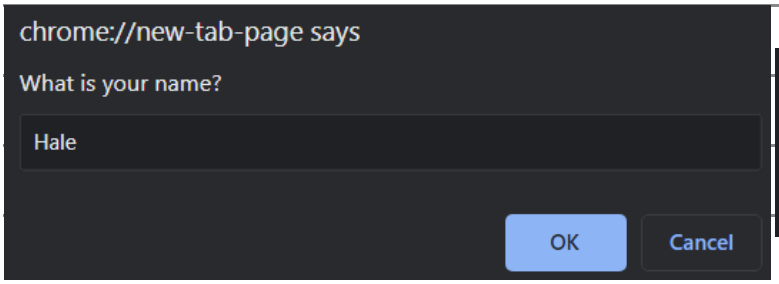
```
prompt("What is your name?");
```

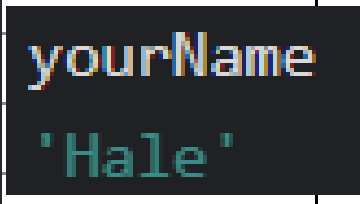




-The keyword var indicates that it is a variable. The value of the name can be changed. For example, if we type myName="Hale". The value assigned to ***Name will be "Hale" instead of "Angela. We need to use var only for the first time when we assign a Value to a Name or when we're constructing a new variable.***

```
var yourName=prompt("What is your name?");
```





-whatever we write in the blank before hitting OK will be assigned to yourName

Summary:

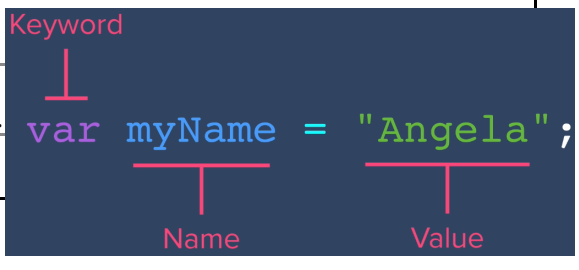
prompt(); :-used to ask for information by the use of pop ups. If we decide to store the information, it will be stored. For example we can store the information inputed by the use of variables.

var myName="Hale"; :- method to declare variables. We need to use var only for the first time.

-we can write whatever we want instead of myName & "Hale"

console.log(); :-used to print

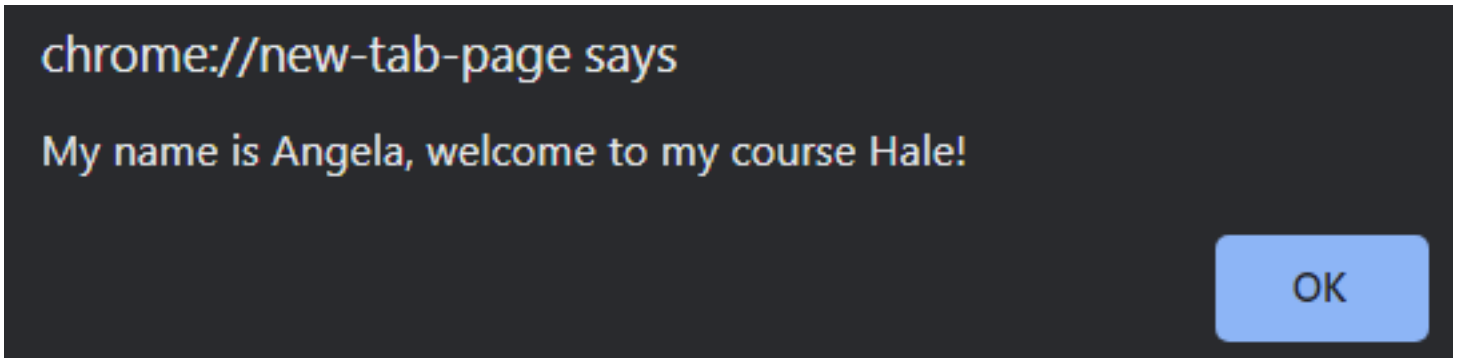
var x=prompt("Hello"); :-the value of x will be whatever you input.

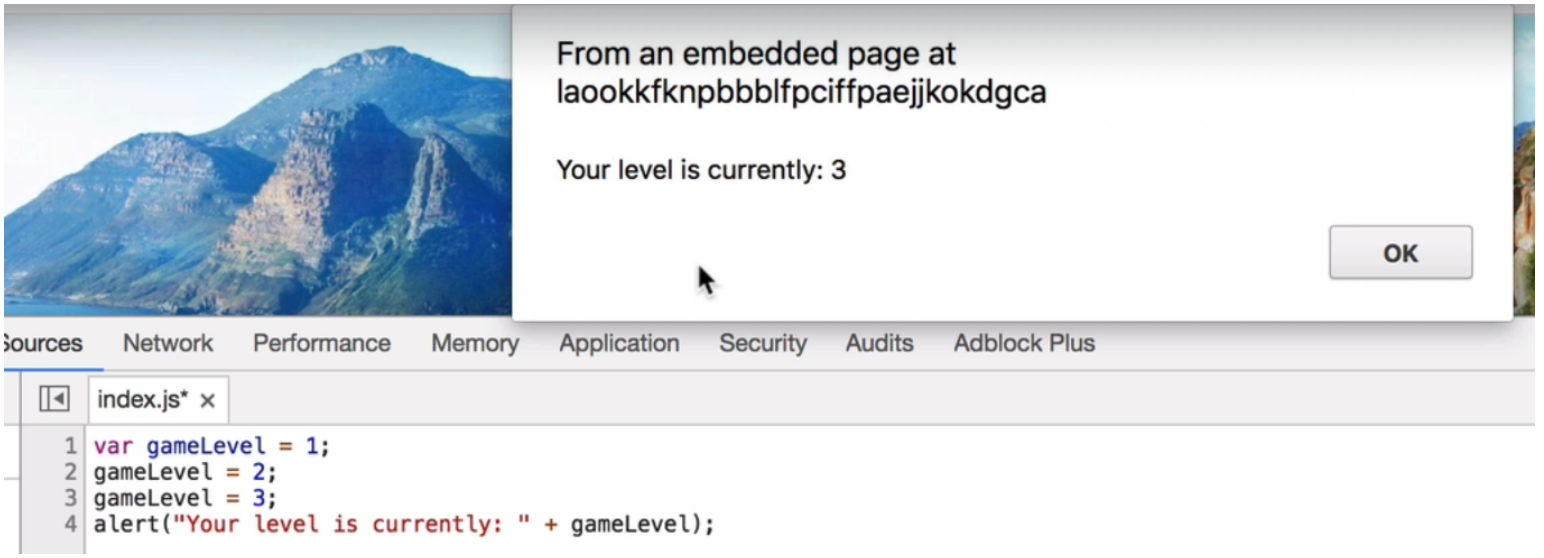


More Notes

```
var myName="Angela";
var yourName=prompt("What is your Name?");
alert("My name is " + myName +", welcome to my course " +yourName+"!");
```

-The 2nd line will prompt us to input a value for yourName and we inputed Hale. Then, the 3rd line will result in:





console.log();

-this is an equivalent of print.

-console.log("Hale");

-this prints Hale.

```
function test() {
    var a = "3";
    var b = "8";

/***********Do not change the code above 👆*******/
//Write your code on lines 7 - 9:
    var c = a;
    a = b;
    b = c;
/***********Do not change the code below 👇*******/

    console.log("a is " + a);
    console.log("b is " + b);
}
```

-this code exchanges the value of the variables a and b.

Topic/Title: Naming and Naming Conventions for JavaScript Variables

Keywords/Questions:

Ctrl+L

Right Click Reload Button

Empty Cache and Hard Reload

Camel Casing

meaningful names

no space

no keyword

no starting with no.

abc..,123..,$,_

Notes:

Ctrl + L :- to clear your console on windows. But, this doesn't clear information stored in the browser like values of variables. To delete hard coded information like this we need to right click on the reload button  and click "Empty Cache and Hard Reload". But this won't delete what we wrote in sources>snippets. Even if we open a new tab and reload we will find it.

Common rules programmers will follow while naming variables:

-Give meaningful names to your variables. -Use Camel Casing

-We can't name our variable as var or any other keyword. This will result in an error. But you can call it for example as myvar.

-Your variable name can't begin with a number. For example, you can't name it 1hale. `var 123 = 123;` But, you can include a number in your variable name. `var my123 = 123;`

-You can't have a space within your variable name

-Your variables can contain only: `var abc123$_`

-letters -numbers -the dollar sign ($) -underscore(_)

Camel casing:-writing the first word in your variable as small letters & all other words as capital -this will be used throughout the course. It will solve problems that might arise like therapistFinder.com which might be read as the rapist... `var userScoreFinal = 12;`

Summary:

Ctrl+L:-makes the console empty. But, doesn't delete stored info

Right Click reload button>Empty Cache and Hard Reload :- deletes stored info in the console like variables

-It doesn't delete info in sources>snippets(information inside them appears even in new window).

Camel Casing:- E.g.: var userScoreFinal="Hale"; -the first letter of the name will be in lowercase but the others will be in uppercase.

Variable naming rules:-1)have meaningful names 2)don't start with numbers 3)don't use spaces 4)don't use keywords 5)use only variables, numbers, $, _ 6)use camel casing

Question 1:

Which variable name is not a valid in Javascript?

- `var myage = 12;`
- `var my age = 12`
- `var myAge = 12`
- `var my_age = 12`

Question 2:

Which is the best name for a Javascript variable that will represent student 1's score?

- `1studentScore`
- `s1Score`
- `studentonescore`
- `student1Score`

Topic/Title: Strings
String Lengths and Retrieving the Number of Characters
Slicing and Extracting Parts of a String
Challenge: Changing Casing in Text
Challenge: Changing String Casing Solution

Keywords/Questions:

Concatenation

word.length;

word.slice(lower,upper);

word.toUpperCase();

word.toLowerCase();

Notes:

We can combine two or more strings with a + sign.

"a" + "b" = "ab"

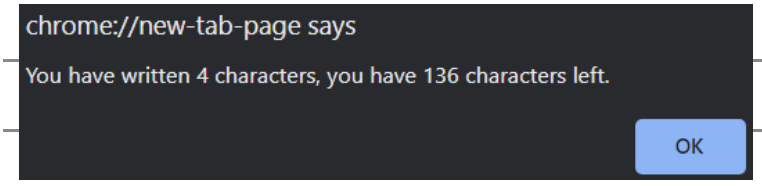
Concatenation:

-the process of using a + sign to combine two or more strings.

word.length; :- used to find the length or the number of characters of a string.

```
var name = "Angela";
name.length;
```
<- 6

```
var tweet=prompt("Compose your tweet:");
alert("You have written " + tweet.length + " characters, you have " + (140-tweet.length) + " characters left.");
```



-Let us say I typed Hale in the prompt.

word.slice(); :-used to slice a word. For example "hale".slice(0,2); is "ha".

```
var name = "Angela";
name.slice(0,1);
```
<- "A"

-JavaScript counting starts from 0 that means "A" is in the 0th position i

```
var name = "Angela";
name.slice(5,6);
```
<- "a"

```
var name = "Angela";
name.slice(0,3);
```
<- "Ang"

Summary:

Concatenation:- adding two or more strings together. We use t he + sign.

word.length:- used to find out the length of a string.

word.slice(lowerbound, upperbound); :- used to slice strings.

word.toUpperCase(); :- converts all the letters in a string to uppercase.

word.toLowerCase(); :-converts all the letters in a string to lowercase.

More Notes

word.slice(); continued:

-word.skuce(lower bound, upper bound);

-to find out the number of characters we are slicing we can use the formula:

slice length = upper bound−lower bound

```
var name = "Angela";
name.slice(1,5);
```

-we get 4 characters sliced.

```
alert(prompt("Compose your tweet:").slice(0,140));
```

-cuts down whatever we have written in the prompt to 140 characters and displays it.

word.toUpperCase(); :-turns every single character in the string to the uppercase version of it.

```
var name = "Angela";
name = name.toUpperCase();
```

```
> name
< "ANGELA"
```

```
> "Angela".toUpperCase();
< 'ANGELA'
```

word.toLowerCase(); :-turns every single character in the string to the lowercase version of it.

-just does the opposite of word.toLowerCase();

```
var name=prompt("What is your Name?");
name=name.slice(0,1).toUpperCase()+name.slice(1,name.length).toLowerCase();
```

-This line of code Capitalized the first letter of every word we enter while keeping all the other letters in lowercase. For example, if we input the word "hAlE", the final value of name would be "Hale".

Topic/Title: Basic Arithmetic and the Modulo Operator in JavaScript

Increment and Decrement Expressions

Keywords/Questions:

Addition(+) +=

Subtraction(-) -=

Multiplication(*) *=

Division(/) /=

Modulo(%) %=

Pre/Post-Increment ++x x++

Pre/Post Decrement --x x--

BODMAS

parseInt();

//

Notes:

Addition:- made by using the + sign. `var a = 2 + 3; //5`

Subtraction: made by using the - sign. `var b = 10 - 2;`

Multiplication:- made by using the * sign. `var c = 3 * 3;`

Division:- made by the / sign. `var d = 6 / 2;`

Modulo:- made by the % sign. Used to compute reminder. `var e = 9 % 6; //3`

//:- used for commenting out.

JavaScript operations follow the BODMAS rule.

parseInt(string):- used to convert a String to an Integer.

-For example, parseInt("1") converts the string "1" to an Integer.

```
var dogAge=prompt("What is the age of your dog?");
var humanAge=4*(dogAge-2)+21;
alert(humanAge);
```

-For example, if we input 2 in the prompt, the alert will be 21.

Increment Expression:- ++ `var x = 5; x++; //6` -equivalent to x=x+1

Decrement Expression:- -- `var x = 5; x-- ; //4` -equivalent to x=x-1

+= :used to increase the value of our variable. `var x = 5; x += 2 ; //7`

-= :used to decrease the value of our variable.

```
var x = 5;
var y = 3;
x += y ; //8
```

+=

-=

*=

/=

Summary:

JavaScript Operations:-Addition(+) -Subtraction(-) -Multiplication(*) -Division(/) -Modulo(%)

-Post-Increment Expression:x++ Post-Decrement Expression:x--

-Pre-Increment Expression:++x Pre-Decrement Expression:--x

JavaScript Operations follow the BODMAS rule.

x+=5, x-=5, x/=5, x*=5 :-can be used instead of x=x+5, x=x-5, x=x*5, x=x/5 respectively.

parseInt(String); :-converts a string to an integer.

// :-used to comment out.

*=:- applied similarly to +=.

/=:-applied similarly to +=.

What does `y` equal?

```
1  var x = 3;
2  var y = x++;
3  y += 1;
```

Incorrect answer. Please try again.

In this line: var y = x++ the value of x is assigned to y before x is incremented, so y equals 3 on line 2, while x equals 4. There fore on line 3, y now equals 4 instead of 5.

Pre-increment/decrement --> The current value of the variable is used, ***before*** the increment/decrement

```
1  var y = --x + z;  // x is decremented by 1, then the result of the expression
2                    // "x + z" (using this new value of x) is assigned to y.
```

Post-increment/decrement --> The value ***after*** the increment/decrement operation is used.

```
1  var y = x++ + z;  // the expression "x + z" is evaluated (using the current
2                    // value of x) and assigned to y, then x is incremented.
```

Topic/Title: Functions Part 1: Creating and Calling Functions
Functions Part 2: Parameters and Arguments
Functions Part 3: Outputs & Return Values

Keywords/Questions:

function(name){
//commands
}

parameters

arguments

console.log();

return x;

Math.floor();

Math.round();

Math.pow();

**

Notes:

-All naming a variable rules apply to naming a function.

-All lines of code inside the function should be indented.

function hale() { alert("My name is Hale."); }

-We don't need ; at the end of the }

-We call the function by simply writing **hale();**

Ctrl+F:-We can use it to replace the names of all our variables. We first highlight the variable

-We can touch Ctrl+F. Then,  . Then, we touch the Replace All button.

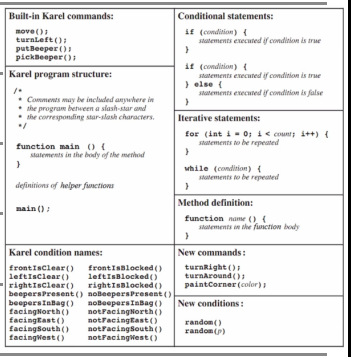
console.log(); :-used to print.

-It will be printed inside console.

Karel Stanford Robot:- very similar to HUBO.

-can be used for programming practice.

-http://stanford.edu/~cpiech/karel/ide.html



Using parameters:

```
function getMilk (bottles) {
   var cost = bottles * 1.5;
   //Do something with cost
}
```

-Here the parameter is bottles

getmilk(5); :-the argument here is 5.

Parameter is variable in the declaration of function. Argument is the actual value of this variable that gets passed to function.

Math.floor(); :- gives us the largest integer <= a given number.

-Math.floor(2.5); is 2 -Math.floor(2.2); is 2 -Math.floor(2.8); is 2 -Math.floor(2); is 2

Summary:

All the variable naming rules apply to naming functions. When declaring functions, we indent the codes inside

function name(parameters){//commands} :- declaring a function

- name(argument); :-calling the function

Ctrl+F:- used for highlighting and replacing all occurrences of a particular variable in snippet/console.

console.log(); :-used to print. -return x; :-used to return outputs of a function.

Math.floor(argument); :-used to round to the lowest integer

Math.round(argument); :-just rounds the number to the nearest integer

Math.pow(number, exponent); and number**exponent :- both are used to raise a number to a particular exponent.

return 5;

- The return keyword is used with functions. It gives the function an output.

```
function getMilk (money) {
  return money % 1.5;
}
var change = getMilk(4);
```

  - The output of getMilk(4) is 1 so the value of the variable change will be 1.

Math.pow(5, 2);

- this means 5 the power of 2.
- it raises the first number to the value of the second number.
- similar to the exponentiation expression(**).
  - 5**2=25

Exponentiation expression(**):

- similar to Math.pow(arg1, arg2);
- 5**2=25

Math.round();

- rounds a number to the nearest whole number