

JavaScript

- Consists of statements
- Very specific set of rules
- All statements end with a semi-colon ";"

Functions:

- Anything that is a word followed by a set of parentheses: moveForward(10);
- The word is the name of the function. In this case the function name is moveForward
- Computer must know how to run the function; hence it is important to define the function
- There is a set of built-in functions as well

Parameters:

- Anything inside the parentheses are called the parameters of the functions. Here, the number "10" is a parameter.
- Parameters need not be numbers all the time.
- There is a data type called strings that are like words. All strings must be put in double quotes.

Example:

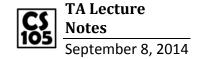
- say("hello");
 - Here the function name is "say".
 - o Parameter is "hello", it is a string.
- You can even add two strings together:
 - o say("Hello" + "World");

Variables:

- The variables in JavaScript are like the ones in Scratch but need to have the word "var" in front of them for the computer to know it is a variable.
- Example: var name;
- This is required only when we use the variable for the first time in each program
- You can set the value of the variable by using an equal sign.
 - $0 \quad x = x + 1;$ $0 \quad x = x * 4;$
 - o All basic arithmetic operations are possible

Built-in functions:

- If you try to call a function that is not built-in, the computer throws an error message –
 "The function xyz is not defined"
- print("whatever");



- o print takes the parameter passed and prints it on the screen.
- readline();
 - readline() will read a line from the console and return it back to you. This
 means that we can assign this line to a variable. The variable can be called
 anything.
 - o Example: var xyz = readline();
 - When readline() is used, the console waits for us to input a line.
- Consider the following code snippet for a more sensible action:

```
print("What is your name?");
var name = readline();
print("Hello" + name);
```

Output:

```
What is your name?
Wade (This is our input)
Hello Wade
```

If conditionals:

Consider this addition to the existing code.

```
if (name == "Wade")
{
    print( "...shouldn't you be in class?")
}
```

New output:

```
What is your name?
Wade (This is our input)
Hello Wade
...shouldn't you be in class?
```

If we give a different input, output will be different.

```
What is your name?
Alex (This is our input)
Hello Alex (Note: no "shouldn't you be in class")
```