JavaScript Basics

Q1: Week 5

written by Airielle Randolf D. Dela Cruz

to tell you the truth...

You haven't *actually* been programming this whole time...

to tell you the truth...

You haven't *actually* been programming this whole time...

This is where JavaScript comes in.

if HTML,

allows you to structure your web page.

and if CSS,

allows you to lay out and design your web page.

then JavaScript,

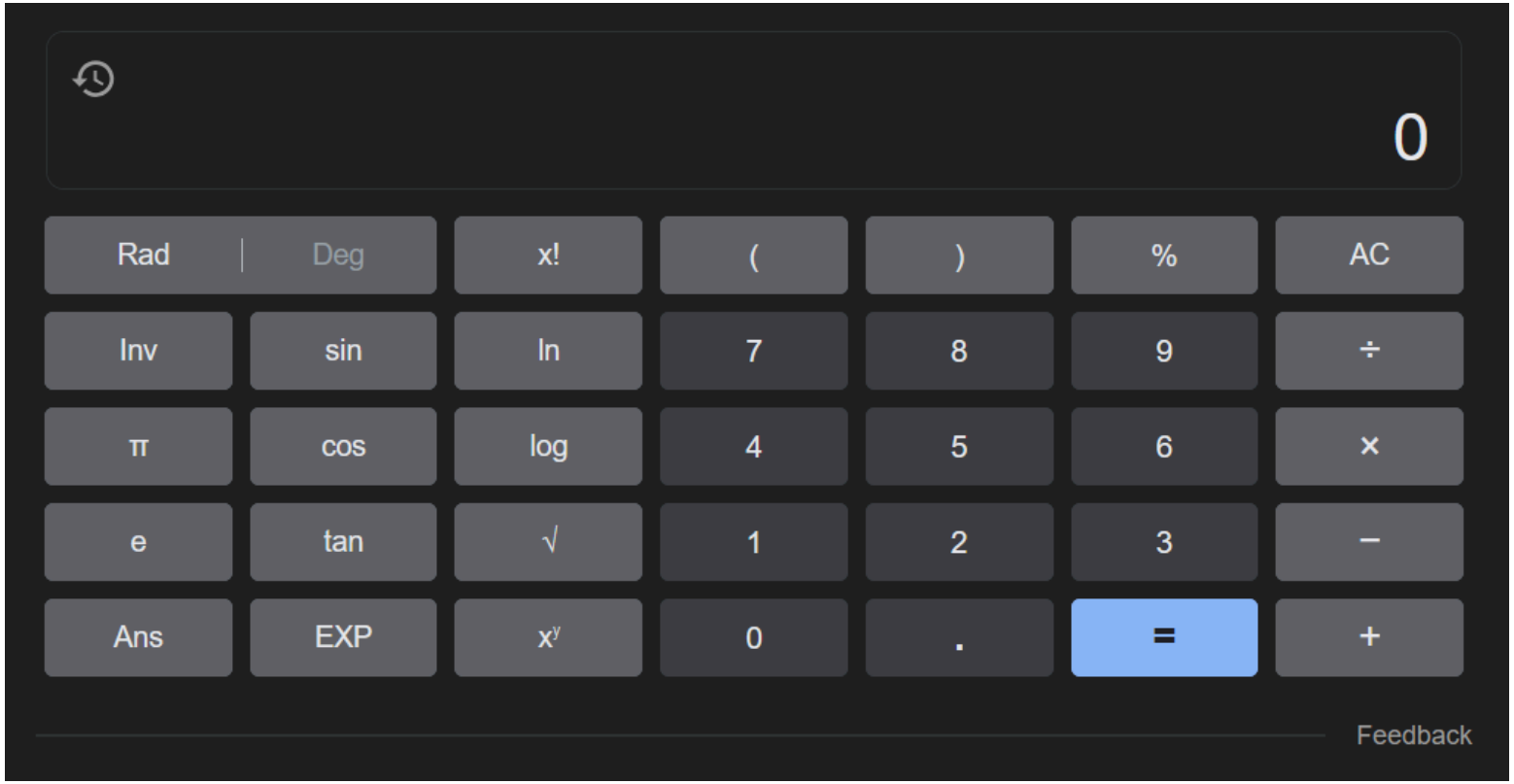
allows you to program your web page.

what is JavaScript?

JavaScript (or JS) is the programming language of the web. It allows web developers to make use of computations and methods to give their websites complex functionality.

why JavaScript?

Google's calculator would not be possible without JavaScript's programming powers!



JavaScript invented math?

General Programming Fundamentals

Before we discuss JavaScript, we need to know the fundamentals of programming that apply to all programming languages.

What is Programming?

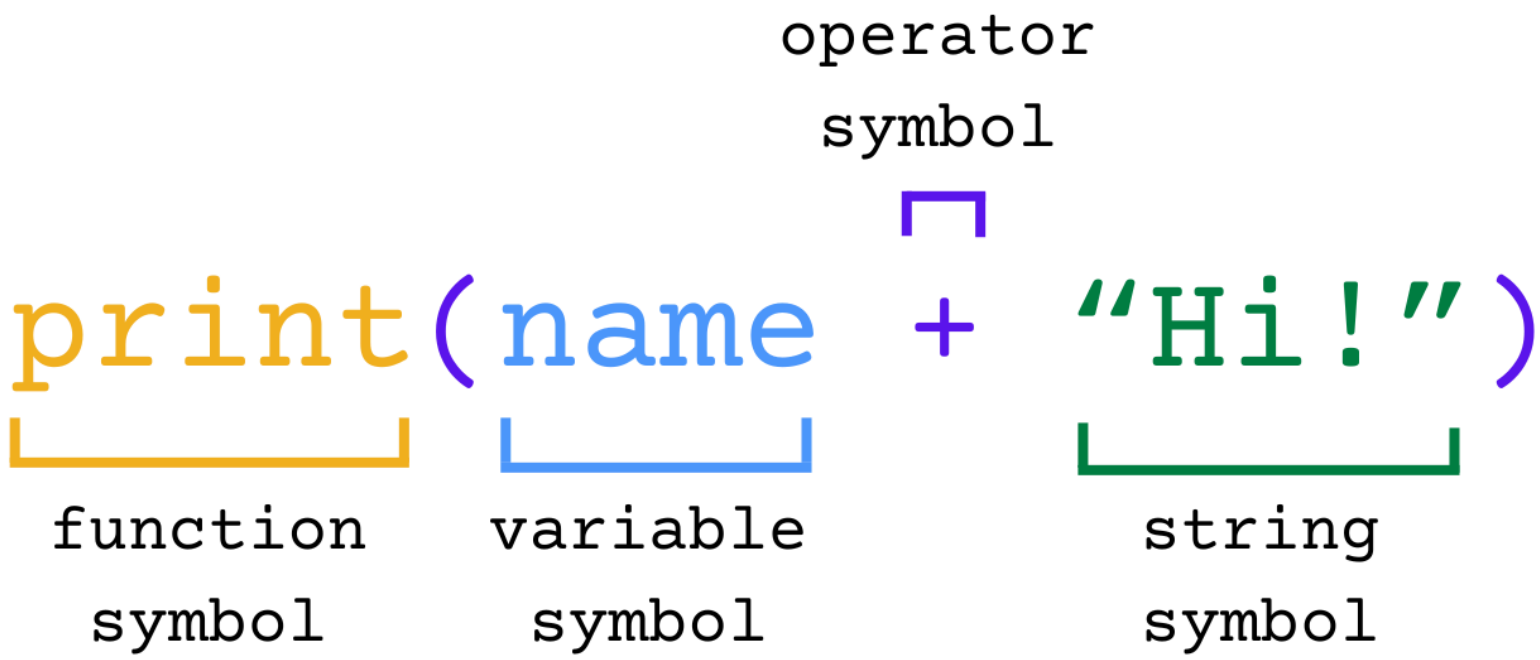
Programming is often defined as **writing a set of instructions that the computer can eventually execute.**

```
var x = 10;
function thisFunction(var num) {
    return num + 5;
}
for(let i = 0; i < 7; i++) {
    x = thisFunction(x)
}
```

The problem is: **HOW DO WE READ THIS?**

Reading Code Symbols

When it comes to READING code, you can think of code as a series of **symbols** that all mean something specific.



Operators

The easiest kind of symbol to understand are the operators and numbers. These are at the forefront in performing computations when programming.

```
6 + 9
-7 * 2
3 / 2
```

Most programming languages have special operators, like exponentiation, modulo, and operators for bit logic.

```
2 ** 2      // exponent
```

Keywords

Programming languages usually have a set of keywords that mean something specific. Keywords in JavaScript include `var`, `for`, `function`, and many more.

We will go over these as we learn more about JavaScript.

```
var
let
for
while
```

Symbol Definition and Referencing

In programming, it's very common for YOU to define your own symbols, and then reference YOUR symbols later in the code. It's part of what makes programming so powerful.

```
var my_symbol       // definition

// ...

my_symbol + 2       // reference
```

Variables

Variables are things that store values. When creating a variable, we define a symbol for the variable, which can be referenced later in code. The syntax in JavaScript for creating variables looks like this:

```
var variable_symbol = value
```

Notice how we're using a keyword, a defined symbol, and an operator, just in this one line of code.

Example of Variables

In the example below, a variable with the symbol `skibidi` is defined to store the value `50`. Then, `skibidi` is referenced later in the code to be printed to the screen.

```
var skibidi = 50

print(skibidi)      // displays the value of skibidi
```

Take note that the symbol `skibidi` was used twice. The first line was to define the value of `skibidi`. The second line references the *value stored* by `skibidi`, which is 50. That's the difference between a symbol definition and a symbol reference.

Let's JavaScript!

finally

How do you start writing JavaScript?

One way is to add a `<script></script>` tag inside your page's `<head>` tag.

```
<html>
    <head>
        <script>
            // JavaScript here.
        </script>
    </head>

    <body></body>
</html>
```

How do you start writing JavaScript?

Another way is to create an external JavaScript file (ending in `.js`) and linking it to the HTML page using the `<script>` tag's `src` attribute.

```
<html>
    <head>
        <script src="my_script.js"></script>
    </head>

    <body></body>
</html>
```

Play Along!

In your `my_page.html` (hopefully you still have it), create a `<script>` tag and play along with the JavaScript lesson.

You may also create an external `.js` file and link it to your page.

JS Basics: The DOM

Let's learn how to DOM-inate.

The DOM

The **Document Object Model** (or DOM) is the way we represent HTML elements in JavaScript as objects.

This is essentially the interface that lets JavaScript code access HTML elements.

The "document" symbol

The `document` symbol has many properties that we can use to access parts of our HTML.

One such property is `.getElementById()`, which obtains an HTML element by its ID.

```
document.getElementById("my_id")
```

> Object properties can be accessed by using the dot `.` operator after the object's name.

HTML Elements can be stored in variables

HTML elements obtained through the `document` can be stored in variables as objects themselves.

```
var my_paragraph = document.getElementById("paragraph1")
```

In the code above, the variable `my_paragraph` stores the HTML element with the ID `paragraph1`. This HTML element can now be modified in JS.

Modifying an HTML variable

After accessing HTML elements through the DOM, they can then be modified in JavaScript using their various properties.

An example property is `.innerHTML`, which changes what's "inside" the element. It could be text, or other elements.

```
var my_div = document.getElementById("my_div")

my_div.innerHTML = "Hello!"
```

Multiple HTML elements in one variable

Multiple HTML elements can be queried and stored in one variable in the form of a list. Document properties that can select multiple HTML elements include `.getElementsByClassName()` and `.querySelectorAll()`.

> If you don't know what a list is, don't worry too much, as we won't be using this for now.

```
var paras = document.getElementsByClassName("my_paragraphs");

paras[0].innerHTML = "Hello!"
```

The "style" property

You can also modify the CSS properties of an HTML element in JavaScript using the `.style` property, followed by the name of the CSS property.

```
var my_div = document.getElementById("my_div")

my_div.style.color = "red"
my_div.style.textAlign = "center"
```

Summary

That's about it for now! We just wanted to give you the basics of JS. In summary:

- Programming is about giving instructions to the computer.
- Coding is just writing a bunch of symbols that mean specific things.
- Symbols can include operators, keywords set by the programming language, or self-defined symbols that can refer to variables (or later on, functions).
- JavaScript can be integrated into an HTML page using a `<script>` tag inside `<head>`.

Summary (2)

- The DOM is the interface used by which we access HTML elements in JS.
- HTML elements can be accessed using various `document` methods, like `.getElementById()` or `.getElementsByClass()`.
- HTML elements are expressed as objects, can be stored in JS variables, and have various properties that can be modified in code.

Let's give it a try! (~10 mins.)

Given the basics of JS that you've learned, try to do the following!:

- Open `my_page.html`
- Use a `script` tag to write JavaScript (or link an external `.js` file).
- Change the content of your footer using JavaScript.

> Hint: use `.getElementById()` to access the element.

- Change the text color of the heading

> Hint: give the heading an ID!

Don't be afraid to ask for help!