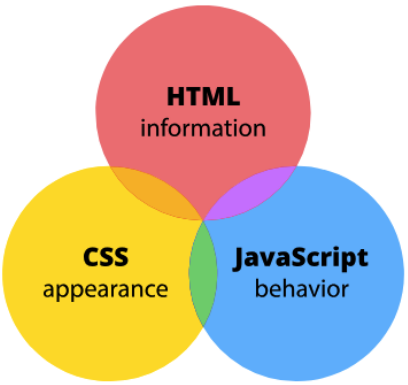
JavaScript

So far you have learned a bit about:

1. **HTML** - how it is used to **define the content** of web pages
2. **CSS** - how it is used to **specify the layout and style** of web pages

While HTML and CSS are languages necessary for getting content onto the internet,…



JavaScript is programming language that allows you to **control** the **behavior** of web pages

JavaScript is a **true** programming language.

Unlike HTML and CSS,

JavaScript allows you to create a website that can:

1. react to input,
2. perform calculations,
3. repeat things.
4. solve problems.

This is all necessary to create:

Applications
Games
Animation

Exercise#1

Copy and paste the following code to see how JavaScript helps a Website do stuff:

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction()
 {
    document.getElementById("demo").innerHTML = "Now some
different text! Amazing!";
 }
</script>
</head>
<body>
<h2>JavaScript in Head</h2>
<p id="demo">See some cool text.</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

Notice: the "programming" in JavaScript is contained in the **<script>** tags

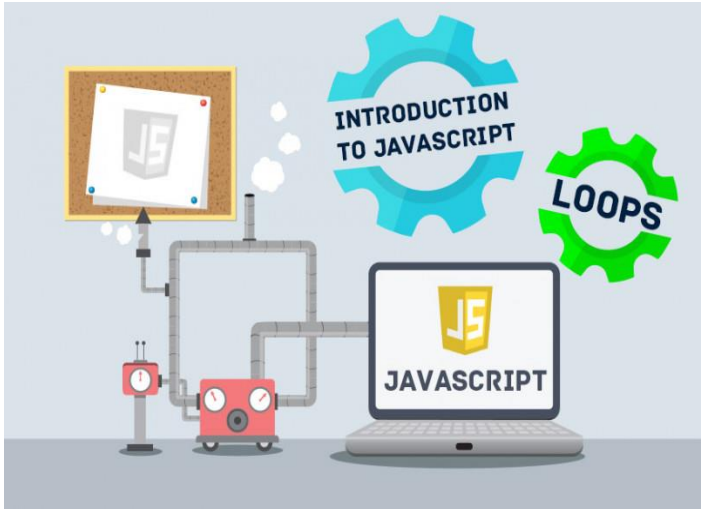
Exercise#2

Try the following example from w3schools:

https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_lightbulb

Javascript can do **all the things that other programming languages** can do:

- Create variables
- Use functions
- Perform calculations
- Create and manipulate lists
- (arrays)
- While Loops
- For Loops
- Conditionals
- Booleans



Exercise#3

Simple addition example: Cut and paste the following code into thimble or another online compiler. **Examine the code to make sure you know how it works and what it is doing.**

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Calculator</h2>

<p id="demo"></p>

<script>
function AddStuff() {
  var y = document.getElementById("txt1").value;
  var z = document.getElementById("txt2").value;
  var x = Number(y) + Number(z);
  document.getElementById("demo").innerHTML = x;
}
</script>
<p>
  Click the button to calculate x.
  <button onclick="AddStuff()">Try it</button>
</p>
<p>
  Enter first number:
  <input type="text" id="txt1" name="text1" value="1">
  Enter second number:
  <input type="text" id="txt2" name="text2" value="2">
</p>
<p id="demo"></p>


</body>
</html>
```

Question:

What does the function **`Number()`** do in JavaScript?

Exercise#4

While loop example: Cut and paste the following code into thimble or another online compiler. **Examine the code to make sure you know how it works and what it is doing.**

sample while loop

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript <em>While</em> Loop</h2>
<script>
var text = "";
var i = 0;
while (i < 10)
      {
      text = "<br>The number is " + i;
      document.write(text);
      i++;
      }
</script>
</body>
</html>
```

Exercise#5

For loops in JavaScript example: cut and paste the following code into thimble or another online compiler. Examine the code to make sure you know **how it works** and **what it is doing.**

For loop

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript For Loops</h2>

<script>

var text = "";
var i;

for (i = 0; i<5; i++)
      {
      text = "<br>The number is " + i;
      document.write(text);
      }
</script>
</body>
</html>
```

Question:

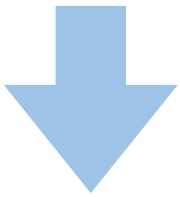
What does `i++` mean???

Exercise#6

Use JavaScript to create a **number guessing game** where the user guesses a number and the computer program will let them know if they are lower, higher, or right on! You should make your user interface part of a **simple** webpage.

- Use can use a `<form>` tag to collect information (if you want).
- The **`parseInt()`** function can change a string into an integer.

Sample Solution below:
(Try to do it on your own first before looking at the solution)



```
<!DOCTYPE html>
<html>
<body>
<input type="text" id="my_input1" />
<input type="text" id="my_input2" />
<input type="button" value="Add Them Together" onclick="doMath();" />

<script type="text/javascript">
    function doMath()
    {
        // Capture the entered values of two input boxes
        var my_input1 = document.getElementById('my_input1').value;
        var my_input2 = document.getElementById('my_input2').value;

        // Add them together and display
        var sum = parseInt(my_input1) + parseInt(my_input2);
        document.write(sum);
    }
</script>

</body>
</html>
```

Animation and Functional, Dynamic Web Pages:

JavaScript is the **programming language** of the internet. It does most of the number crunching and is responsible for the dynamics you see in the front end of a website.

We can also use JavaScript to make simple animations and games. Lots of the "dynamic" things a webpage can do involves moving stuff around the screen. Sooo…let's move some stuff around the screen using JavaScript!

Canvas:

HTML **Canvas** is *one way* JavaScript can be used to move things around the screen.

Exercise#7

- **Read** the following section of w3schools.
- Do the Try it Yourself » exercises.

https://www.w3schools.com/html/html5_canvas.asp

(Do up to and including "**Draw a Circle**")

Exercise#8

Animation

Cut and paste the following code into thimble or another online compiler. Examine the code to make sure you know **how it works** and **what it is doing.**

- Change the 30 to 10,.. then 5,... then 60 to see what the `setInterval()` function is doing.
- Look up `setInterval()` to see how it works
- Note: that **`position++`** is the same as thing as **`position=position+1`**

```
<!DOCTYPE html>
<html>
<head>
<title>Canvas Animation</title>
</head>
<body>
<canvas id="canvas" width="200" height="200">

</canvas>
<script>

var canvas = document.getElementById("canvas");
var ctx = canvas.getContext("2d");
var position = 0;
setInterval(function ()
{
ctx.clearRect(0, 0, 200, 200);
ctx.fillRect(position, 0, 20, 20);
position++;
if (position > 200)
     {
     position = 0;
     }
y }, 30);
</script>
</body>
</html>
```

Check out:
https://www.w3schools.com/jsref/met_win_setinterval.asp
For more details on how `setInterval()` works

Exercise#9

Controlling a **Canvas** drawing with Keys

Cut and paste the following code into thimble or another online compiler. Examine the code to make sure you know **how it works** and **what it is doing.** After entering the code and trying it out answer the question below:

Questions:

- What does a `switch case` do?
- Why is there an **`if statement`** `if (y > 0)` in each `case: statement` (what are the if statements doing)?
- What the following line of code do?
  `window.addEventListener('keydown',doKeyDown,true);`

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8" />
<title>Canvas Test</title>
</head>
<body>
<section>

<div>
<canvas id="canvas" width="300" height="200">
<script>
var canvas;
var ctx;
var x = 150;
var y = 100;
var WIDTH = 300;
var HEIGHT = 200;

function circle(x,y,r)
{
ctx.beginPath();
ctx.arc(x, y, r, 0, Math.PI*2, true);
ctx.fill();
}

function rect(x,y,w,h)
{
ctx.beginPath();
ctx.rect(x,y,w,h);
ctx.closePath();
ctx.fill();
ctx.stroke();
}
```

```
function clear()
{
ctx.clearRect(0, 0, WIDTH, HEIGHT);
}

function init()
{
canvas = document.getElementById("canvas");
ctx = canvas.getContext("2d");
return setInterval(draw, 60);
}

function doKeyDown(evt){
switch (evt.keyCode)
{
case 38:  /* Up arrow was pressed */
if (y > 0)
{
y = y-5;
}
break;
case 40:  /* Down arrow was pressed */
if (y < HEIGHT)
{
y = y+5;
}
break;
case 37:  /* Left arrow was pressed */
if (x > 0)
{
x= x-5;
}
break;
case 39:  /* Right arrow was pressed */
if (x < WIDTH)
{
x =x+5;
}
break;
}
}

function draw() {
clear();
ctx.fillStyle = "white";
ctx.strokeStyle = "black";
rect(0,0,WIDTH,HEIGHT);
ctx.fillStyle = "purple";
circle(x, y, 10);
}

init();
window.addEventListener('keydown',doKeyDown,true);
</script>
</section>
</body>
</html>
```

Exercise#10

Controlling **an image** with Keys

The HTML **canvas** can be limiting at times and often will involve drawing objects from scratch. Another technique is to simply move "images" around the screen.

Copy and paste the code below into thimble, and move around the cat.

```
<html>
  <head>
    <title>JavaScript - Move Object with Arrow Keys using JavaScript
Function.</title>
    <script type="text/javascript">
      //init object globally
      var objImage= null;
      function init(){
        objImage=document.getElementById("image1");
        objImage.style.position='relative';
        objImage.style.left='0px';
        objImage.style.top='0px';
      }
      function getKeyAndMove(e){
        var key_code=e.which||e.keyCode;
        switch(key_code){
          case 37: //left arrow key
            moveLeft();
            break;
          case 38: //Up arrow key
            moveUp();
            break;
          case 39: //right arrow key
            moveRight();
            break;
          case 40: //down arrow key
            moveDown();
            break;
                      }
      }
      function moveLeft(){
        objImage.style.left=parseInt(objImage.style.left)-100 +'px';
      }
      function moveUp(){
        objImage.style.top=parseInt(objImage.style.top)-100 +'px';
      }
      function moveRight(){
        objImage.style.left=parseInt(objImage.style.left)+100 +'px';
      }
      function moveDown(){
        objImage.style.top=parseInt(objImage.style.top)+100 +'px';
      }
      window.onload=init;
```

```
    </script>
  </head>

  <body onkeydown='getKeyAndMove(event)'>
    <h1>JavaScript - Move Object with Arrow Keys using JavaScript
Function.</h1>
    <h2>Use Arrow keys to move object left, top, right and down.</h2>
    <img id="image1" src="https://i.imgflip.com/18mv7q.jpg"
alt="includehelp.com"/>
  </body>
</html>
```

Modify the code above so that: *an image of a mouse is also on the screen and when the cat touches the mouse a message is displayed on the screen that indicates the cat has caught the mouse.* Feel free to extend this exercise into a fun game.

Exercise#11

Classic programming tasks with JavaScript.

Test you skills with JavaScript by complete the following

Complete the following tasks in JavaScript by creating simple programs that are executable in Thimble.

a) Get 3 numbers from the user and display them in ascending order.

b) Use a `for loop` to display the numbers 0 through 5, each in a separate alert window or on a separate html line on a webpage. Get each number to be printed out after 3 second intervals if possible.

c) Use a `for loop` and a modulus operator (%) to calculate and display the numbers in the range 0 to 50 that are multiples of 3.

d) Add up all the numbers in that are input into an **array** by the user.

Exercise#12

Final JavaScript Challenges:

Do one or more of the following JavaScript challenges:

1. Calculator Challenge:

Create HTML/JavaScript calculator that involves the user clicking on an image (like one ones below) and will output the correct answer in the appropriate area on the calculator.

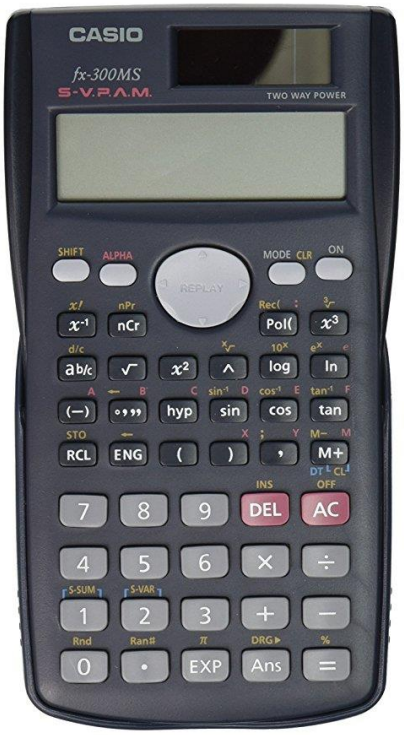
You have two options you can use:

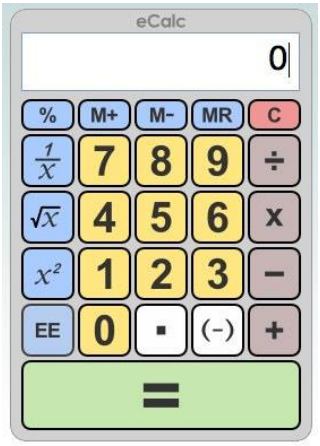
a) "image mapping":

https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_areamap

Cool Tool: https://www.image-map.net/

b) Or you can use HTML **canvas and** HTML elements to draw you calculator from scratch and buttons from scratch.





2. Rock Paper Scissors:

Simple Rock/Paper/Scissors game:

- Ask the user, "Pick 0 for Rock, 1 for Paper, or 2 for Scissors (you'll have to use ParseInt on this one).
- Then generate a random number between 0 and 3
- Compare User number and Random. Check to see who won.
- Print result to the screen.
- Add fun images if you wish.

3. Fish up or Down?

The following is a University of Waterloo Contest Question. Try to solve the following problem using an array.

Problem Description
A fish-finder is a device used by anglers to find fish in a lake. If the fish-finder finds a fish, it will sound an alarm. It uses depth readings to determine whether to sound an alarm. For our purposes, the fish-finder will decide that a fish is swimming past if:

- there are four consecutive depth readings which form a strictly increasing sequence (such as 3 4 7 9) (which we will call "Fish Rising"), or
- there are four consecutive depth readings which form a strictly decreasing sequence (such as 9 6 5 2) (which we will call "Fish Diving"), or
- there are four consecutive depth readings which are identical (which we will call "Constant Depth").

All other readings will be considered random noise or debris, which we will call "No Fish."

Your task is to read a sequence of depth readings and determine if the alarm will sound.

Input Specification
The input will be four positive integers, representing the depth readings. Each integer will be on its own line of input.

Output Specification
The output is one of four possibilities. If the depth readings are increasing, then the output should be `Fish Rising`. If the depth readings are decreasing, then the output should be `Fish Diving`. If the depth readings are identical, then the output should be `Fish At Constant Depth`. Otherwise, the output should be `No Fish`.

Sample Input 1
```
30
10
20
20
```

Output for Sample Input 1
```
No Fish
```

Sample Input 2
```
1
10
12
13
```

Output for Sample Input 2
```
Fish Rising
```