CS 50 Web Design

APCSP Module 2: Internet

Unit 3: JavaScript

LECTURE 12: JAVASCRIPT EVENT-DRIVEN PROGRAMMING

DR. ERIC CHOU IEEE SENIOR MEMBER

Objectives

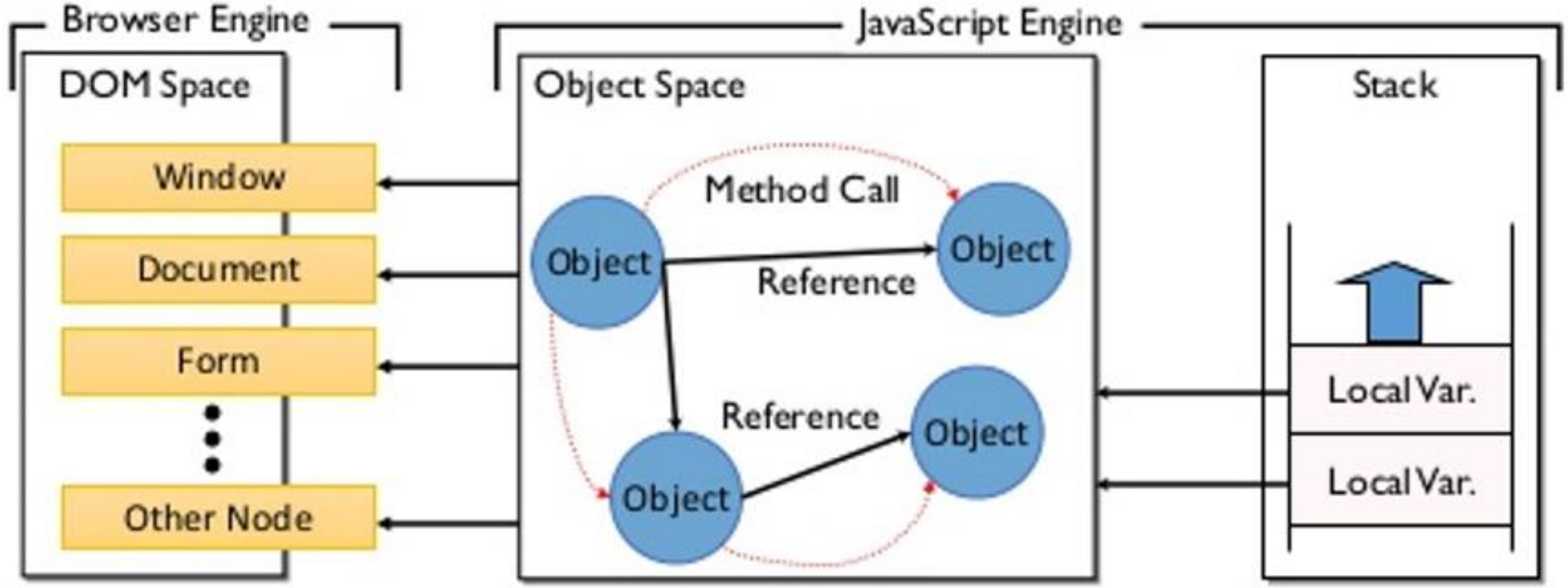
- JavaScript Programming Memory Model
- This Chapter is dedicated in teaching **HTML+CSS+JavaScript** with simple events.
- contains over Projects for JavaScript Units!
- Teaching AP CSP Students : Binary, Hexadecimal, ASCII, HTML Elements, CSS Properties, Variables, Arrays, Functions, Arithmetic Operators, If-Statements, Relational Operators, Loops, PC / Mobile Event Handlers, Random Math, Randomized Images, Video Games.

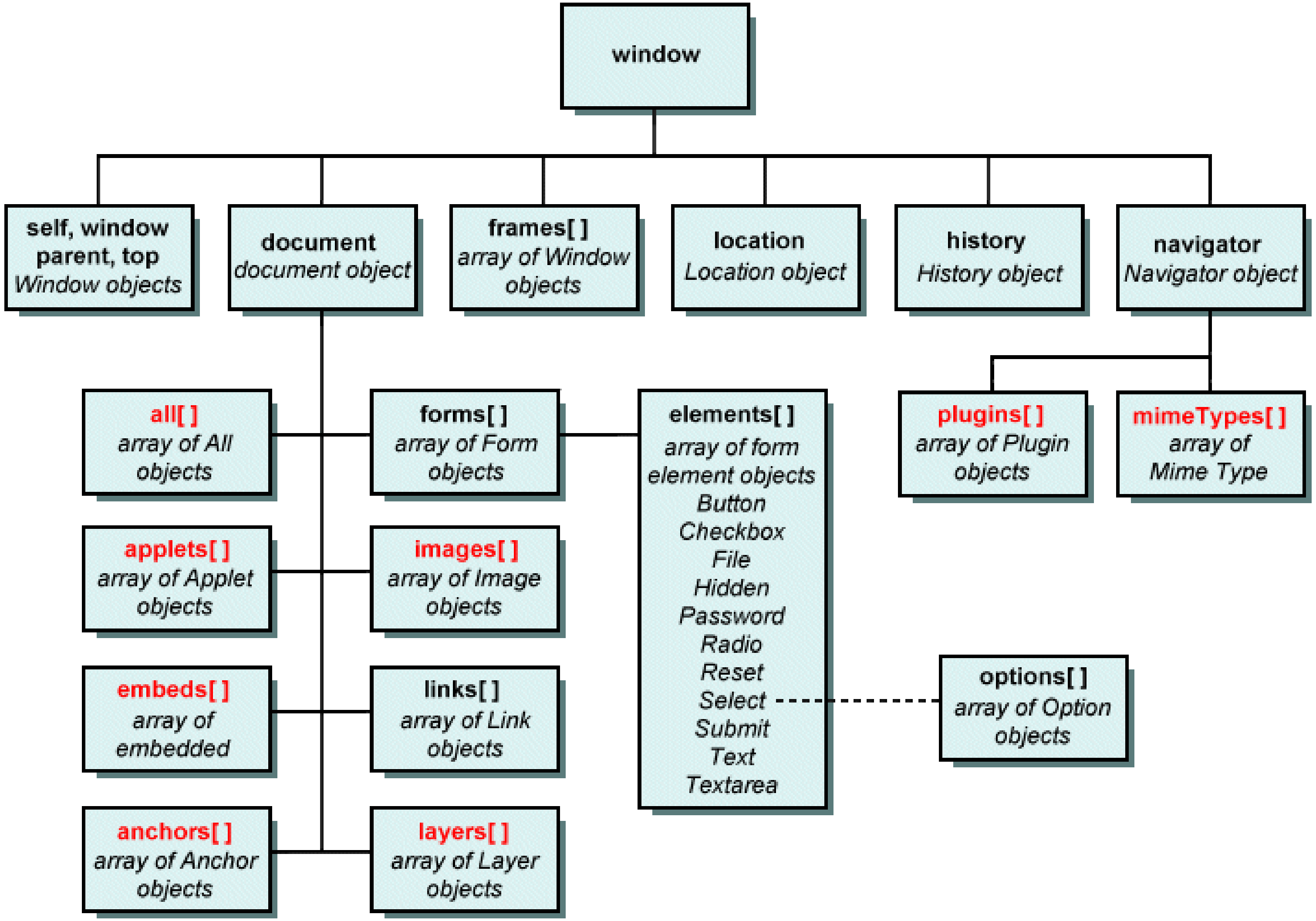
JavaScript Programming Memory Model

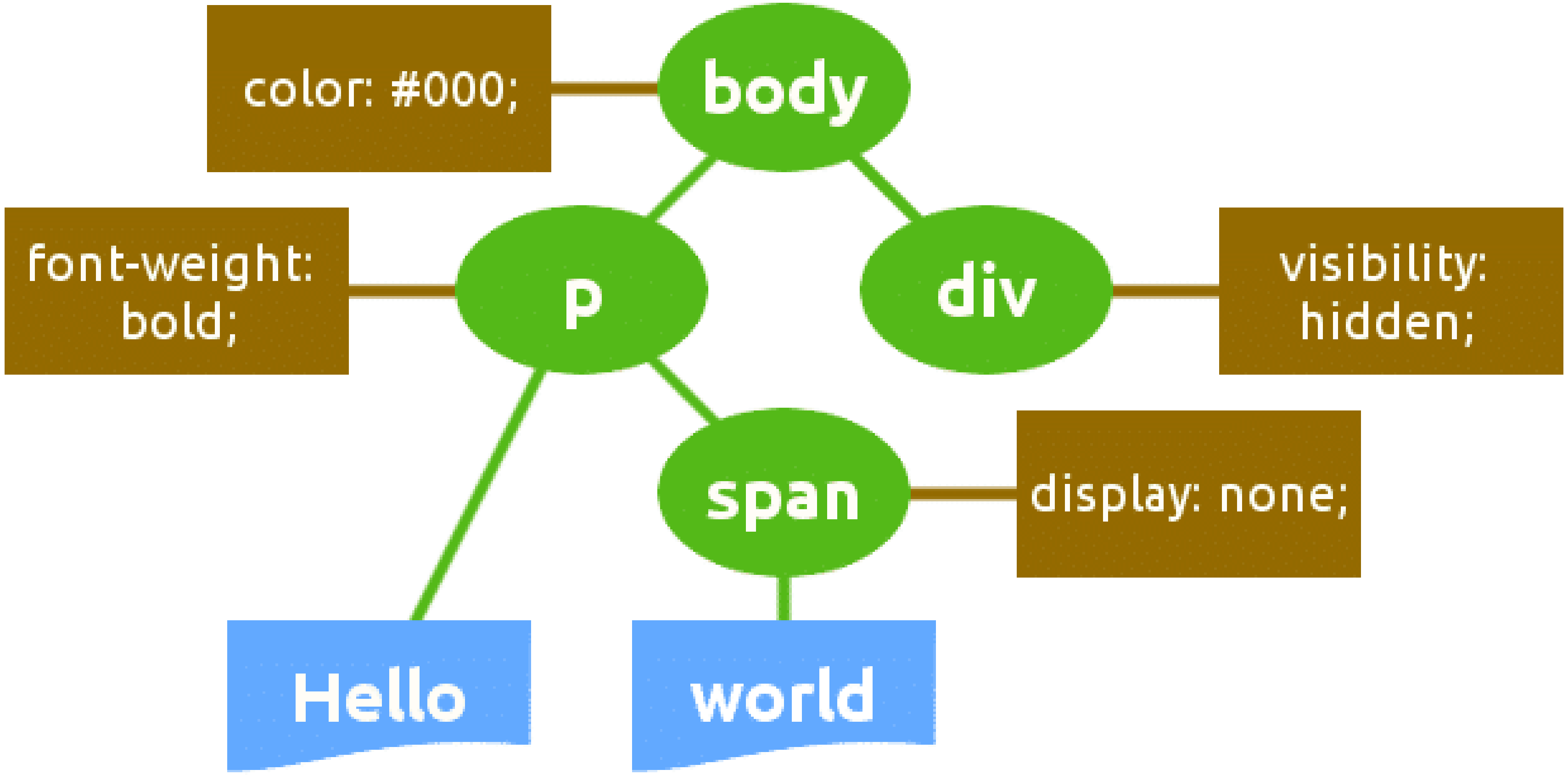
SECTION 1

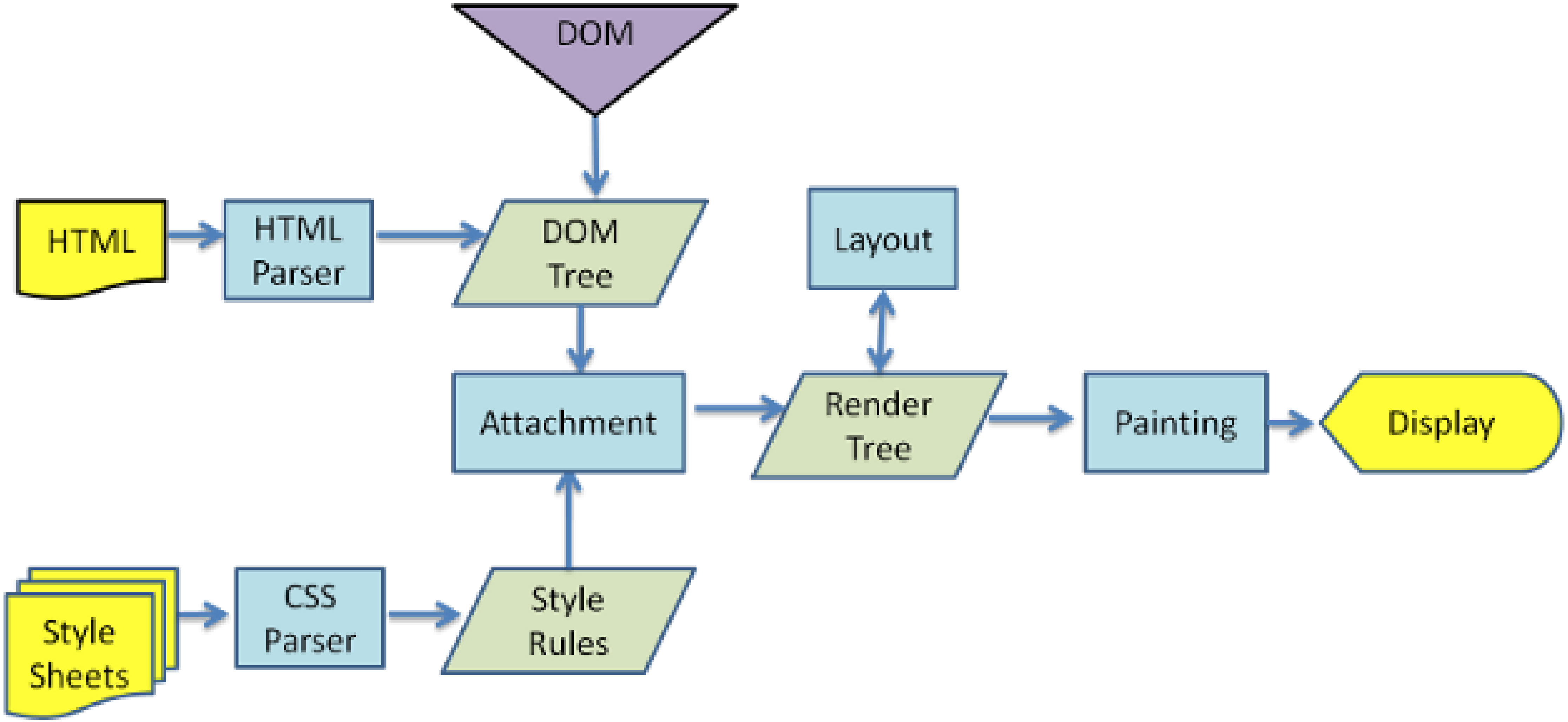
JavaScript Memory Model

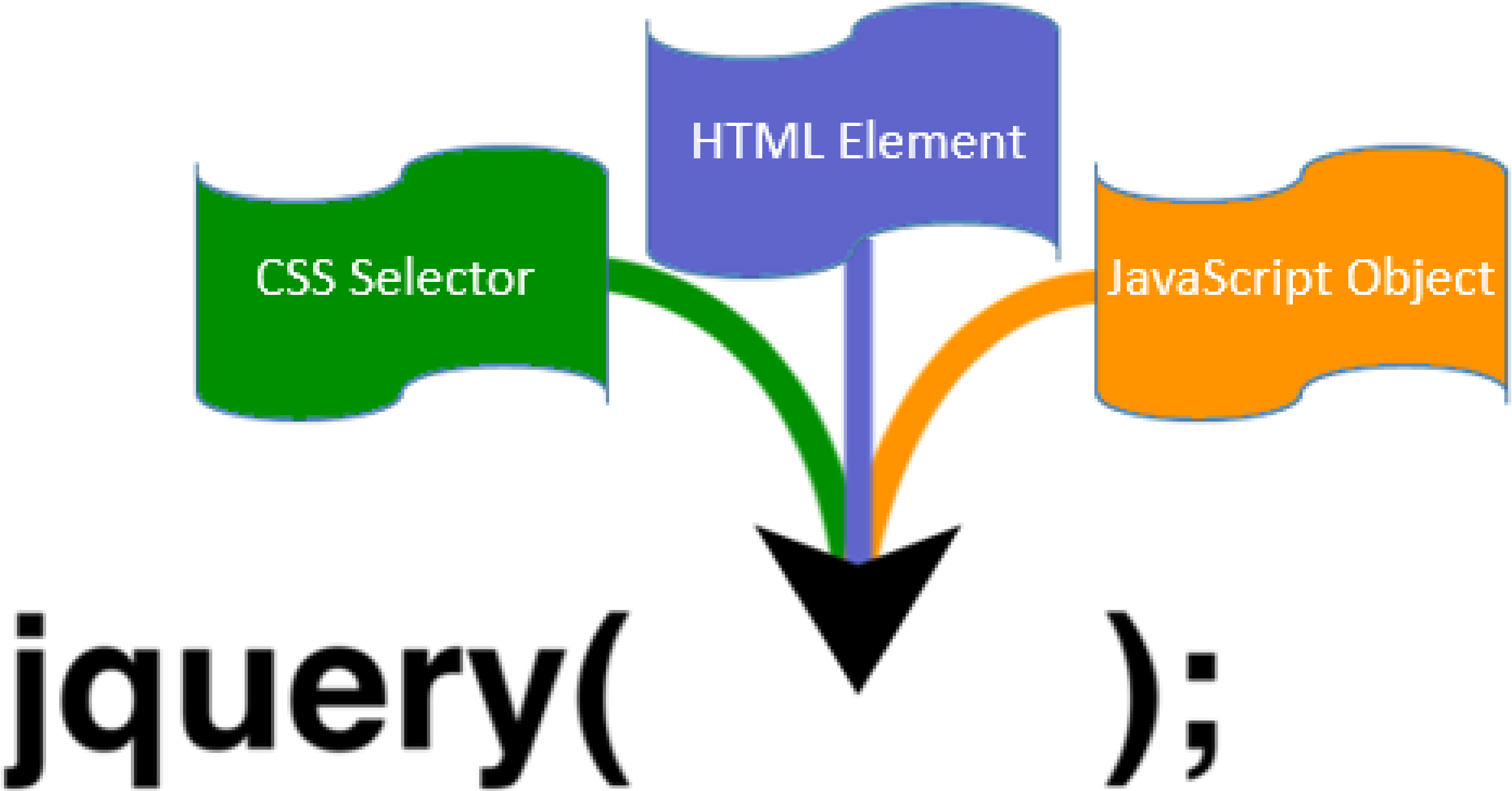
- DOM Space: the space where the Document Object Model representing the HTML's layered structure is represented.
- Object Space: the space where all JavaScript objects are located.
- Stack: short-term memory




window
self, window parent, top Window objects
document document object
frames[ ] array of Window objects
location Location object
history History object
navigator Navigator object
all[ ] array of All objects
forms[ ] array of Form objects
elements[ ] array of form element objects Button Checkbox File Hidden Password Radio Reset Select Submit Text Textarea
plugins[ ] array of Plugin objects
mimeTypes[ ] array of Mime Type
applets[ ] array of Applet objects
images[ ] array of Image objects
embeds[ ] array of embedded
links[ ] array of Link objects
options[ ] array of Option objects
anchors[ ] array of Anchor objects
layers[ ] array of Layer objects


color: #000;
body
font-weight: bold;
p
div
visibility: hidden;
span
display: none;
Hello
world


DOM
HTML
HTML Parser
DOM Tree
Layout
Attachment
Render Tree
Painting
Display
Style Sheets
CSS Parser
Style Rules

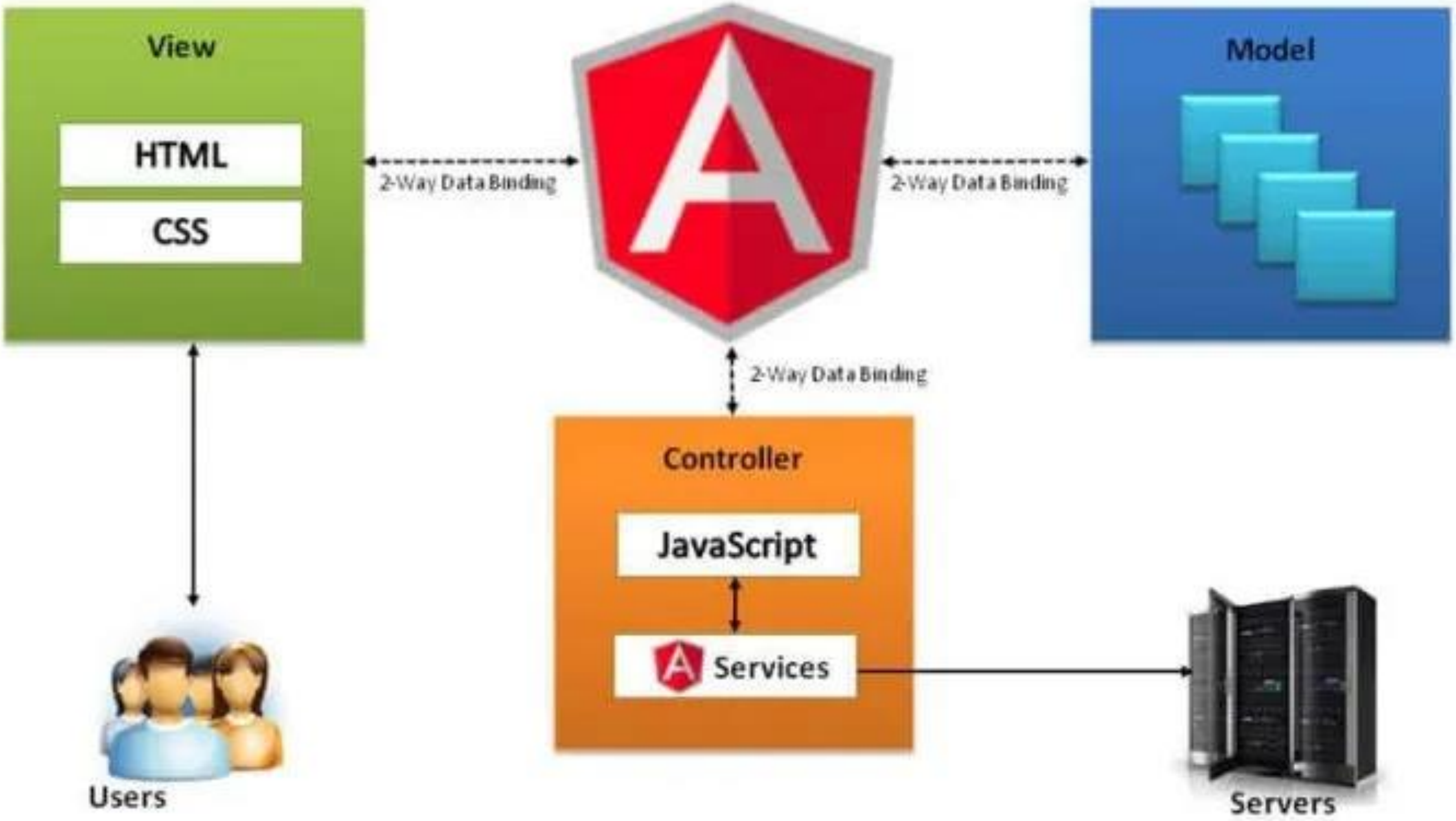


CSS Selector

HTML Element

JavaScript Object

jquery( );


View
HTML
CSS
2-Way Data Binding
2-Way Data Binding
Model
2-Way Data Binding
Controller
JavaScript
Services
Users
Servers

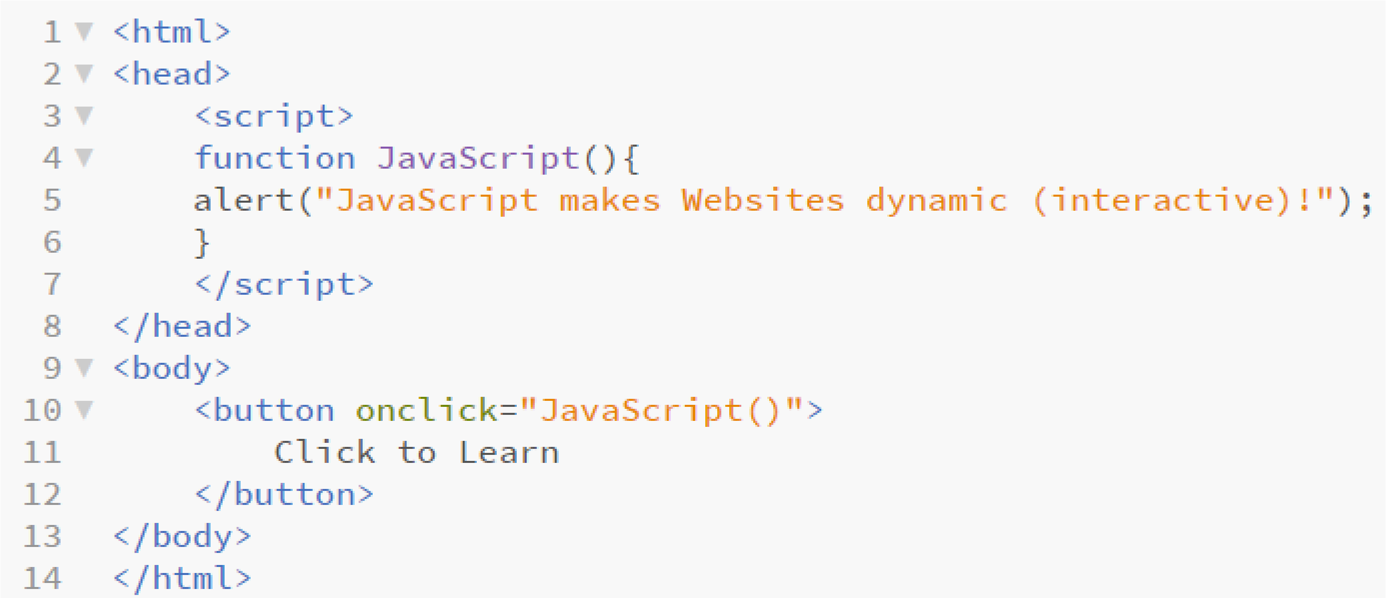
React JS

Basic Projects [Optional]

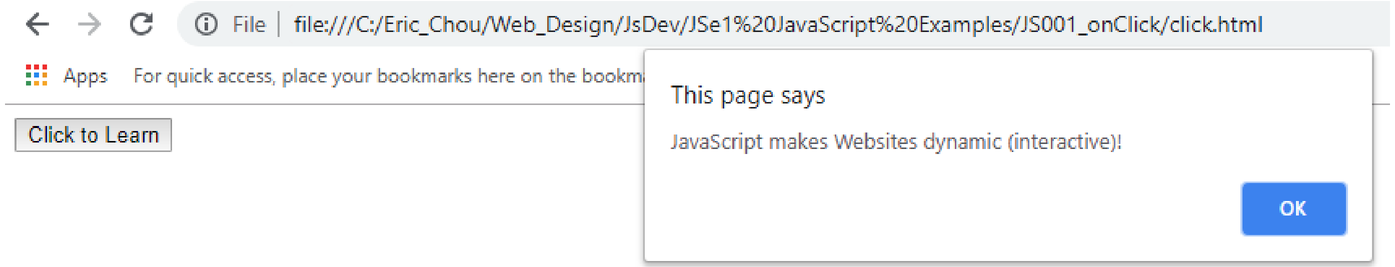
SECTION 1

Project 1: onclick

Demo Program: click.html



```
<html>
<head>
    <script>
    function JavaScript(){
    alert("JavaScript makes Websites dynamic (interactive)!");
    }
    </script>
</head>
<body>
    <button onclick="JavaScript()">
        Click to Learn
    </button>
</body>
</html>
```



file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20Examples/JS001_onClick/click.html

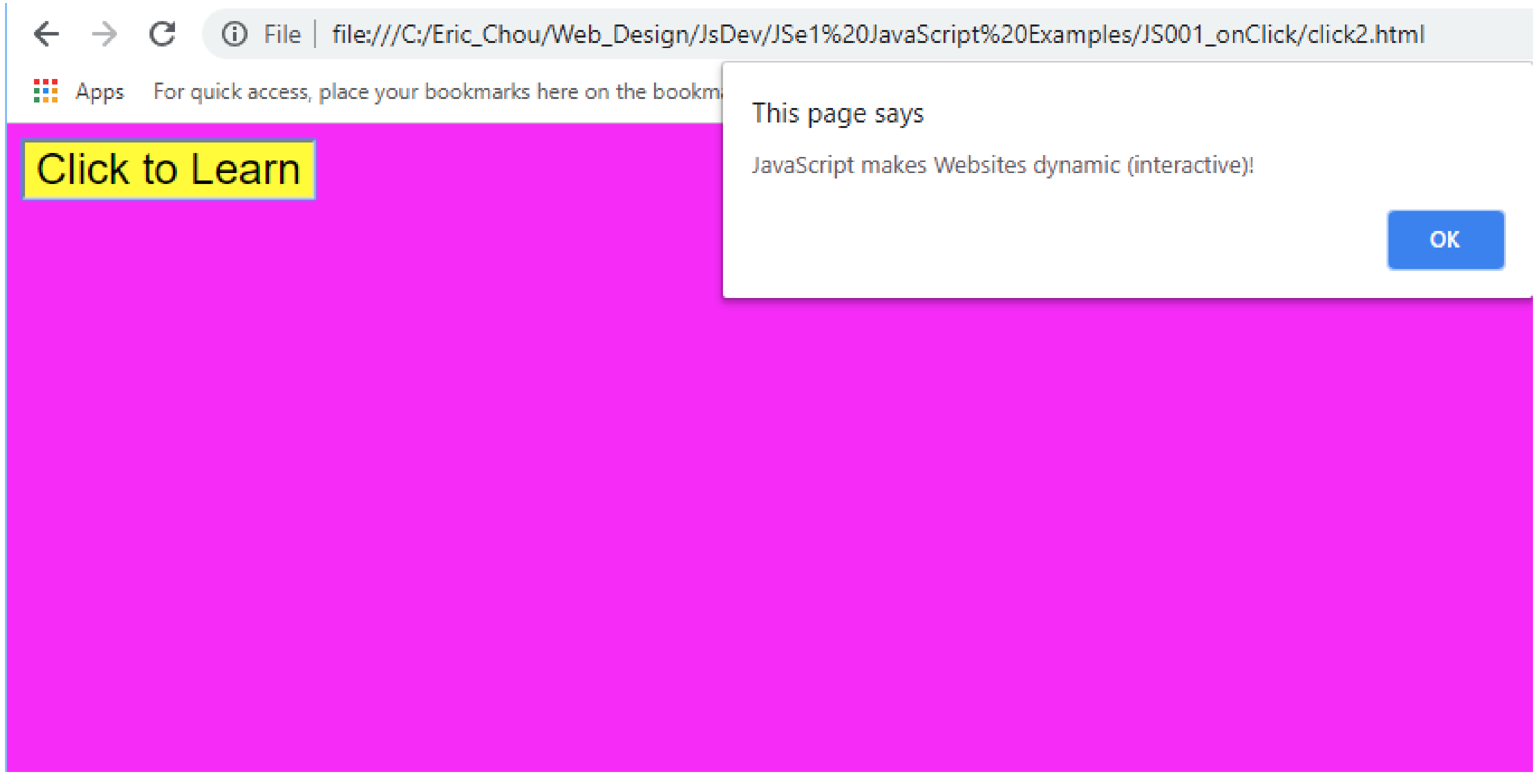
For quick access, place your bookmarks here on the bookm

This page says

JavaScript makes Websites dynamic (interactive)!

Project 1: onclick

Demo Program: click2.html



file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20Examples/JS001_onClick/click2.html

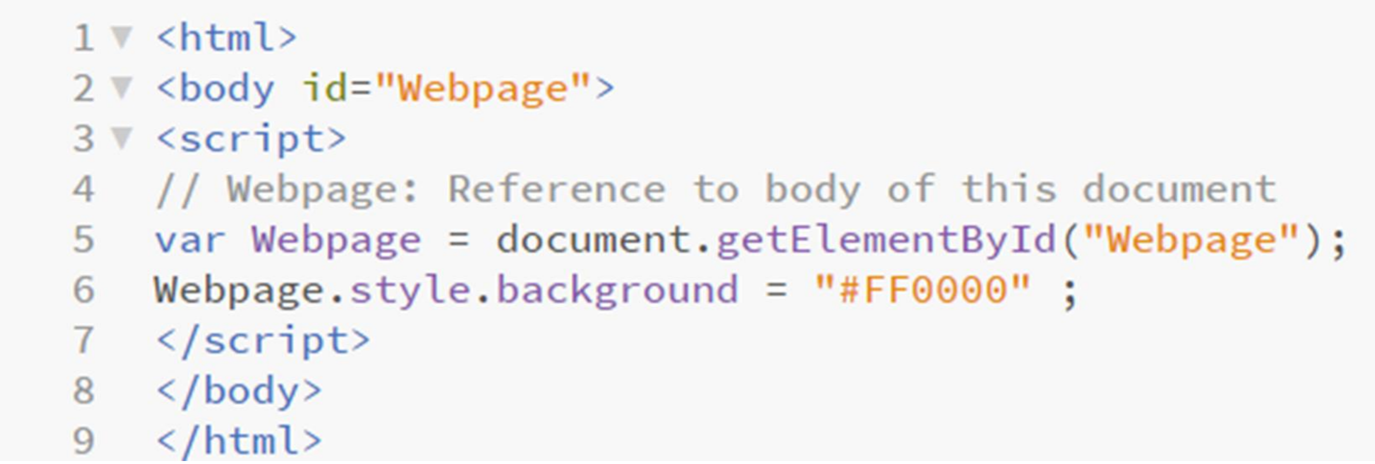
This page says

JavaScript makes Websites dynamic (interactive)!

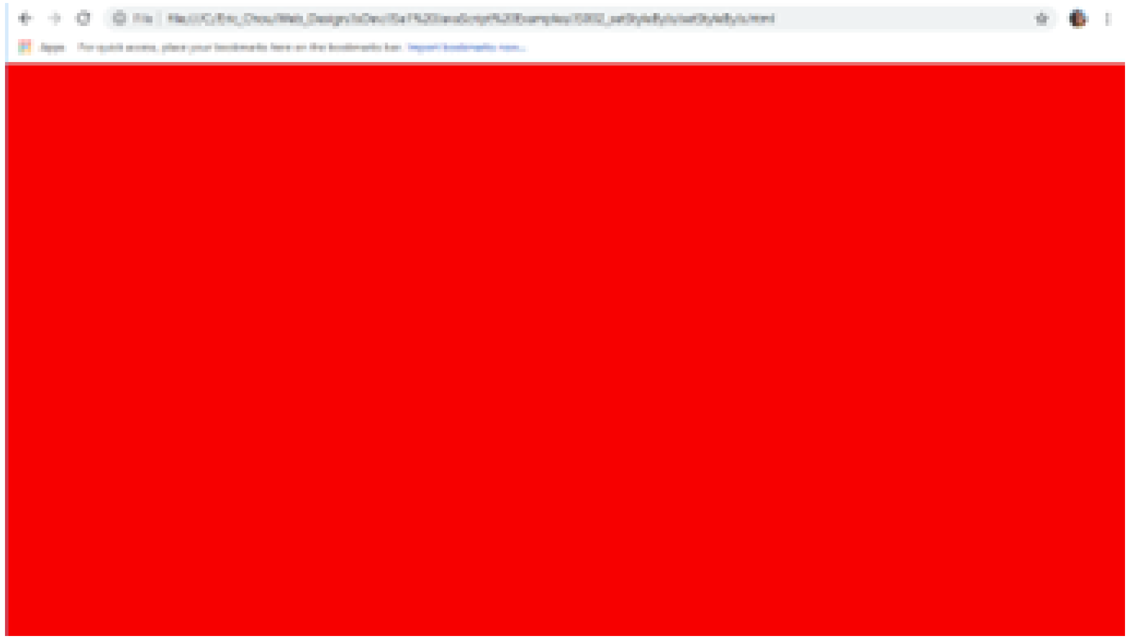
Project 2: Set Style by JavaScript

Demo Program: setStyleByJs.html

- Get the body object reference from DOM
- Set its style property by changing the object’s style data field.

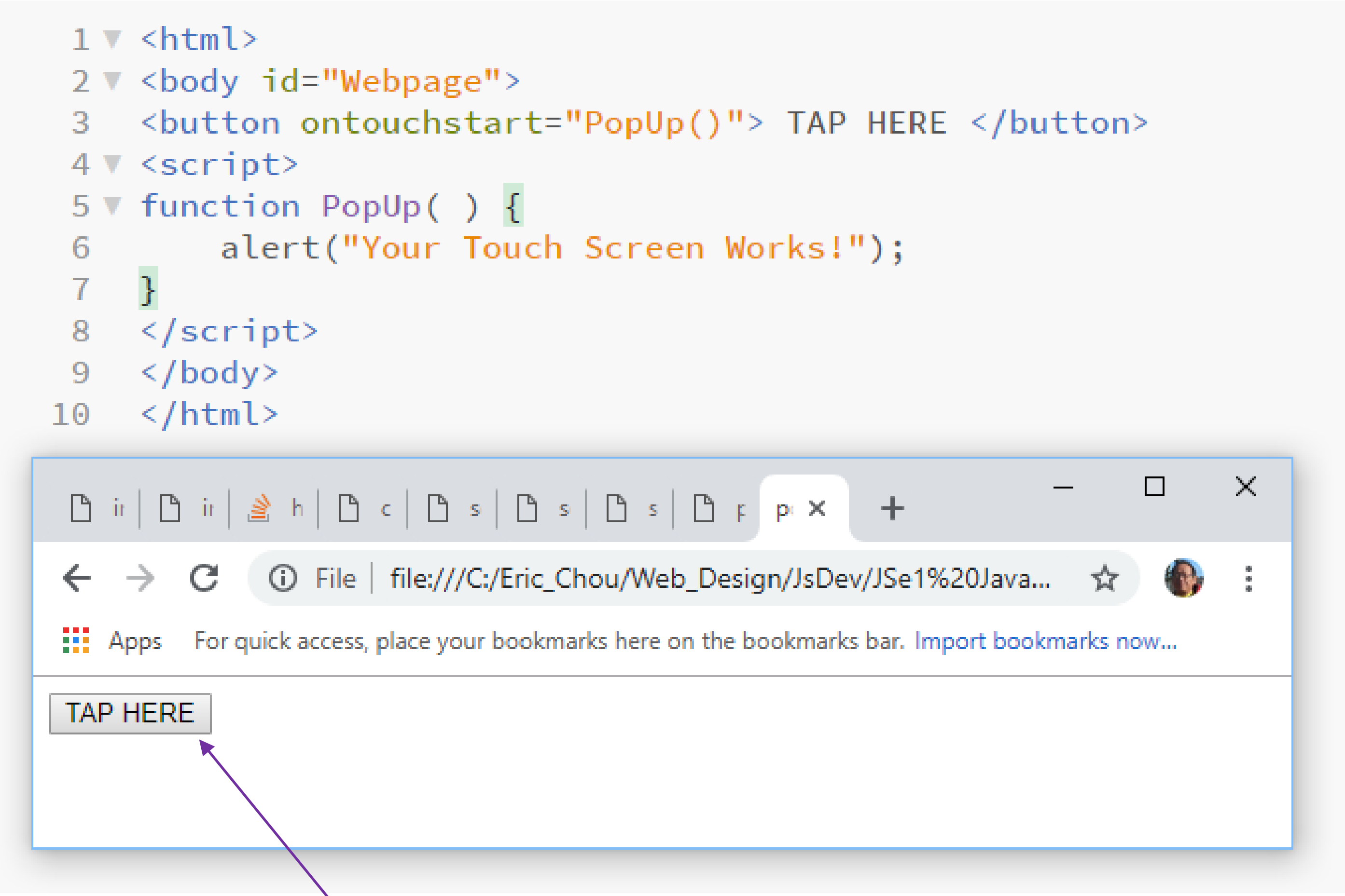


```
<html>
<body id="Webpage">
<script>
// Webpage: Reference to body of this document
var Webpage = document.getElementById("Webpage");
Webpage.style.background = "#FF0000" ;
</script>
</body>
</html>
```



Project 3: On Touch Start

Demo Program: PopUp.html



```
<html>
<body id="Webpage">
<button ontouchstart="PopUp()"> TAP HERE </button>
<script>
function PopUp( ) {
     alert("Your Touch Screen Works!");
}
</script>
</body>
</html>
```

file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20Java...

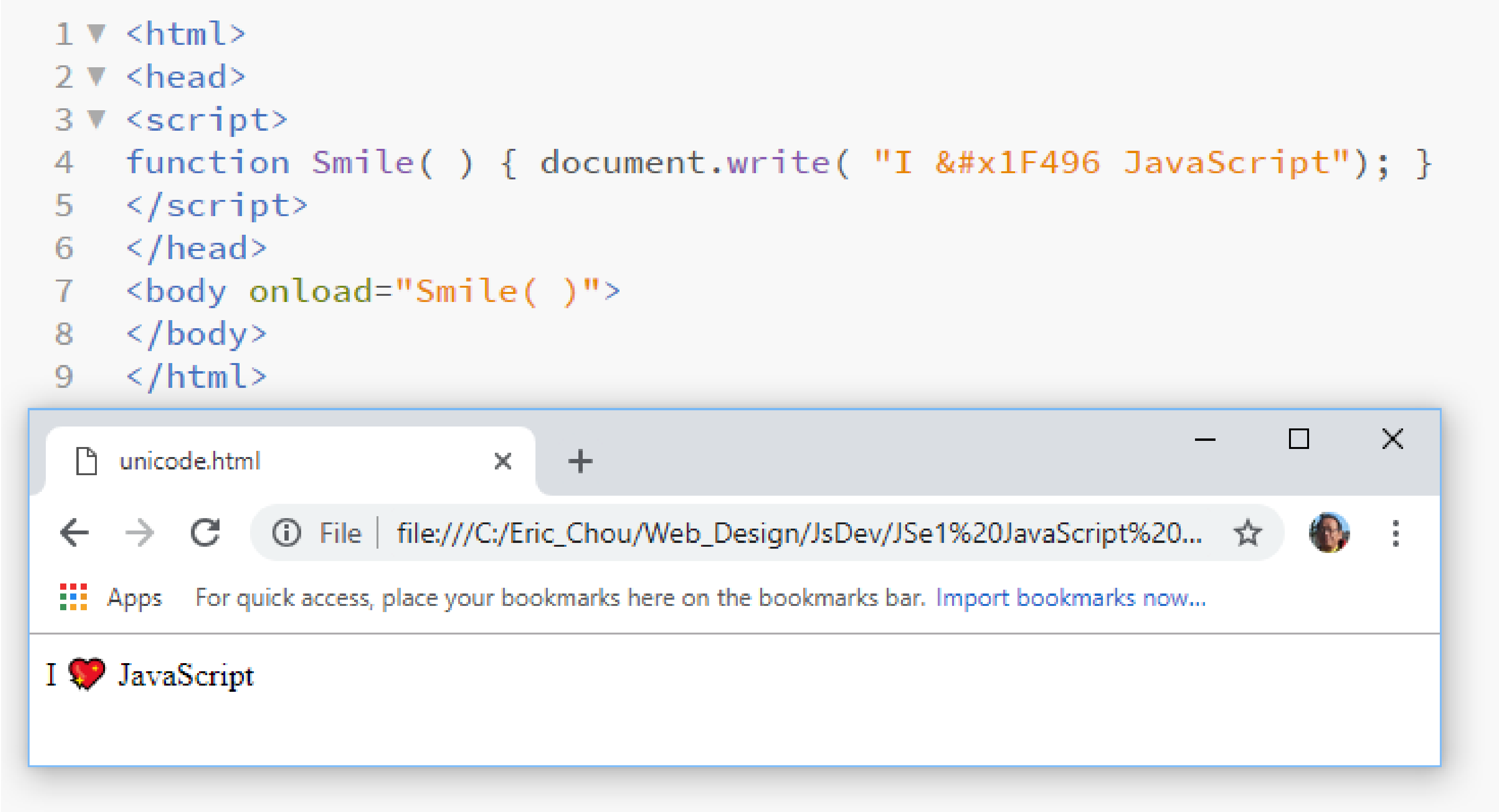
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Take Action on Touch!
No Action on Mouse Click!
Sent by emails.
Try on cell phone.

Project 4: Unicode

Demo Program: unicode.html

- Display a Unicode code symbol on a page.



```
<html>
<head>
<script>
function Smile( ) { document.write( "I &#x1F496 JavaScript"); }
</script>
</head>
<body onload="Smile( )">
</body>
</html>
```

unicode.html
File
file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20...
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

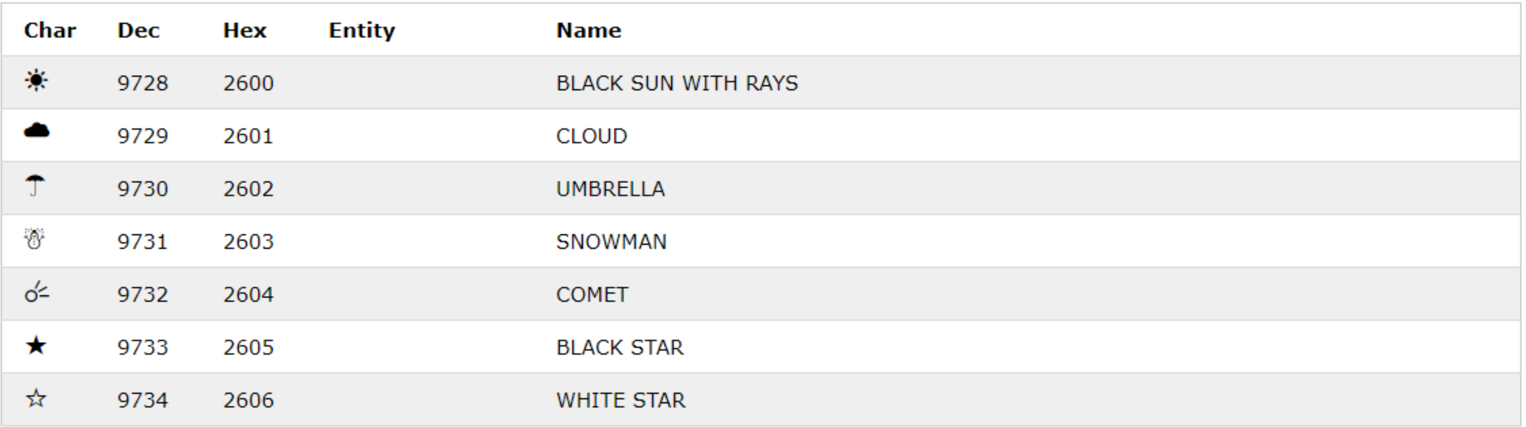
```
I 💖 JavaScript
```

HTML Literals

https://www.w3schools.com/charsets/ref_utf_misc_symbols.asp

Range: Decimal 9728-9983. Hex **2600-26FF**.

- If you want any of these characters displayed in HTML, you can use the HTML entity found in the table below.
- If the character does not have an HTML entity, you can use the decimal (dec) or hexadecimal (hex) reference.



Char	Dec	Hex	Entity	Name
☀	9728	2600		BLACK SUN WITH RAYS
☁	9729	2601		CLOUD
☂	9730	2602		UMBRELLA
☃	9731	2603		SNOWMAN
☄	9732	2604		COMET
★	9733	2605		BLACK STAR
☆	9734	2606		WHITE STAR

HTML Literals

https://www.w3schools.com/charsets/ref_utf_dingbats.asp

Example

<p>I will display &spades;<p>
<p>I will display &#9824;<p>
<p>I will display &#x2660;<p>

Will display as:

I will display ♠
I will display ♠
I will display ♠

File: unicode2.html

```
<!DOCTYPE html>
<html>
    <body>
        <p>I will display &spades;</p>
        <p>I will display &#9824;</p>
        <p>I will display &#x2660;</p>
    </body>
</html>
```

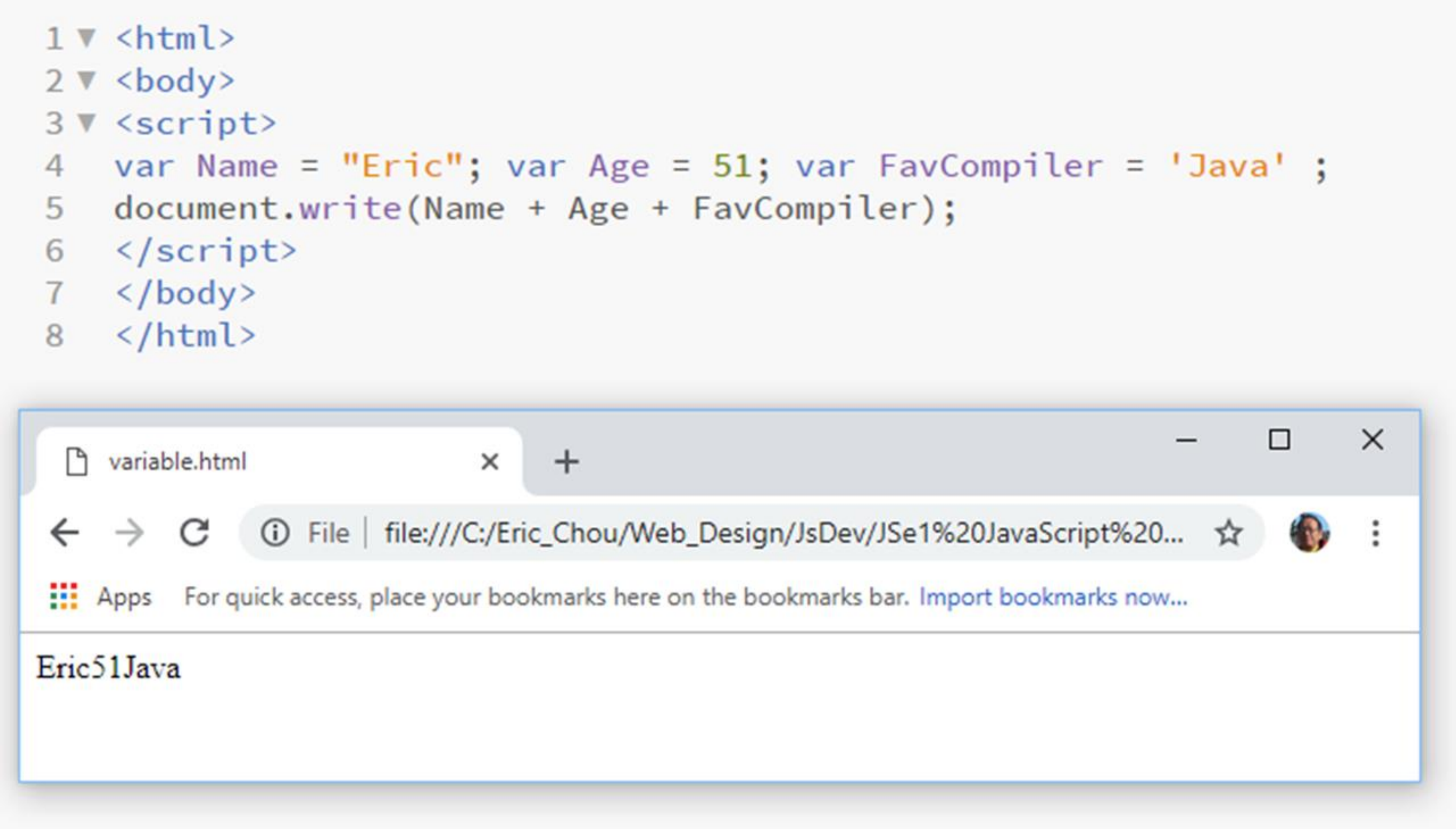
Project 5: Variables

Demo Program: variable.html

- All JavaScript Variables are "Objects".
- Data Types are specified by the values assigned to each Variable.
- To declare a Variable, just type "var" and then "Name" it something before setting = Value (#, 'A', "Word").
- Each JavaScript command is separated with ; (semi-colons).

Project 5: Variables

Demo Program: variable.html



```
<html>
<body>
<script>
var Name = "Eric"; var Age = 51; var FavCompiler = 'Java' ;
document.write(Name + Age + FavCompiler);
</script>
</body>
</html>
```

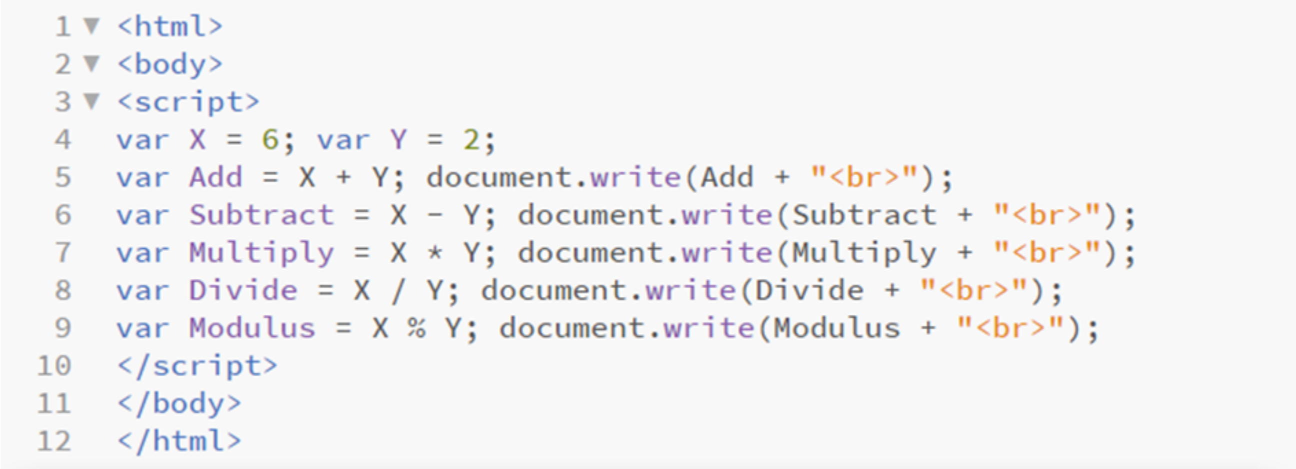
variable.html
File
file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20...
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

```
Eric51Java
```

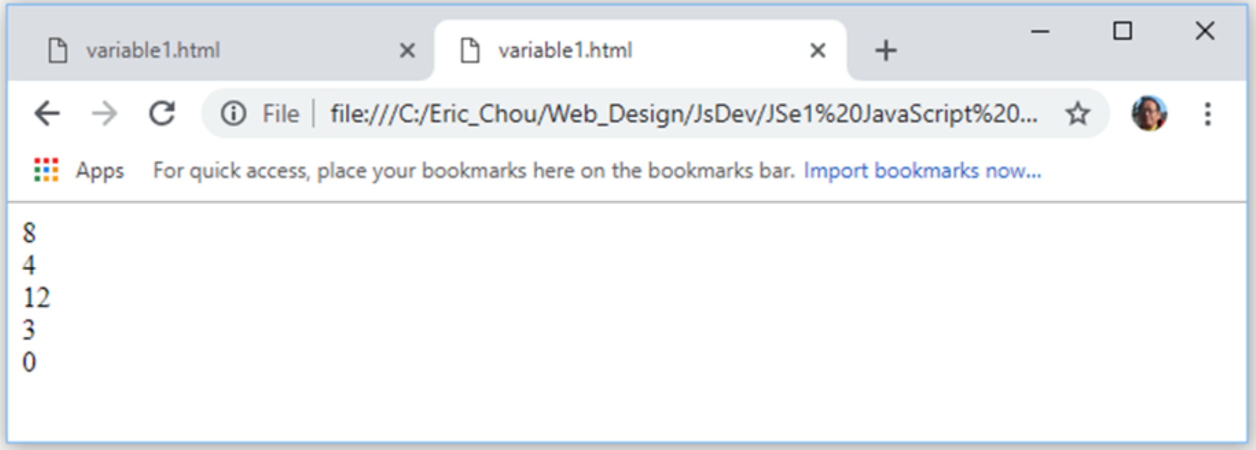
Project 5: Variables

Demo Program: variable1.html

- JavaScript uses the same Arithmetic Operators as other high-level interpreters.
- + **"<br>"** lets you go down one line when printing to the webpage.



```
<html>
<body>
<script>
var X = 6; var Y = 2;
var Add = X + Y; document.write(Add + "<br>");
var Subtract = X - Y; document.write(Subtract + "<br>");
var Multiply = X * Y; document.write(Multiply + "<br>");
var Divide = X / Y; document.write(Divide + "<br>");
var Modulus = X % Y; document.write(Modulus + "<br>");
</script>
</body>
</html>
```


variable1.html
variable1.html
File
file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20...
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

```
8
4
12
3
0
```

Project 6: Basic I/O

Demo Program: io.html

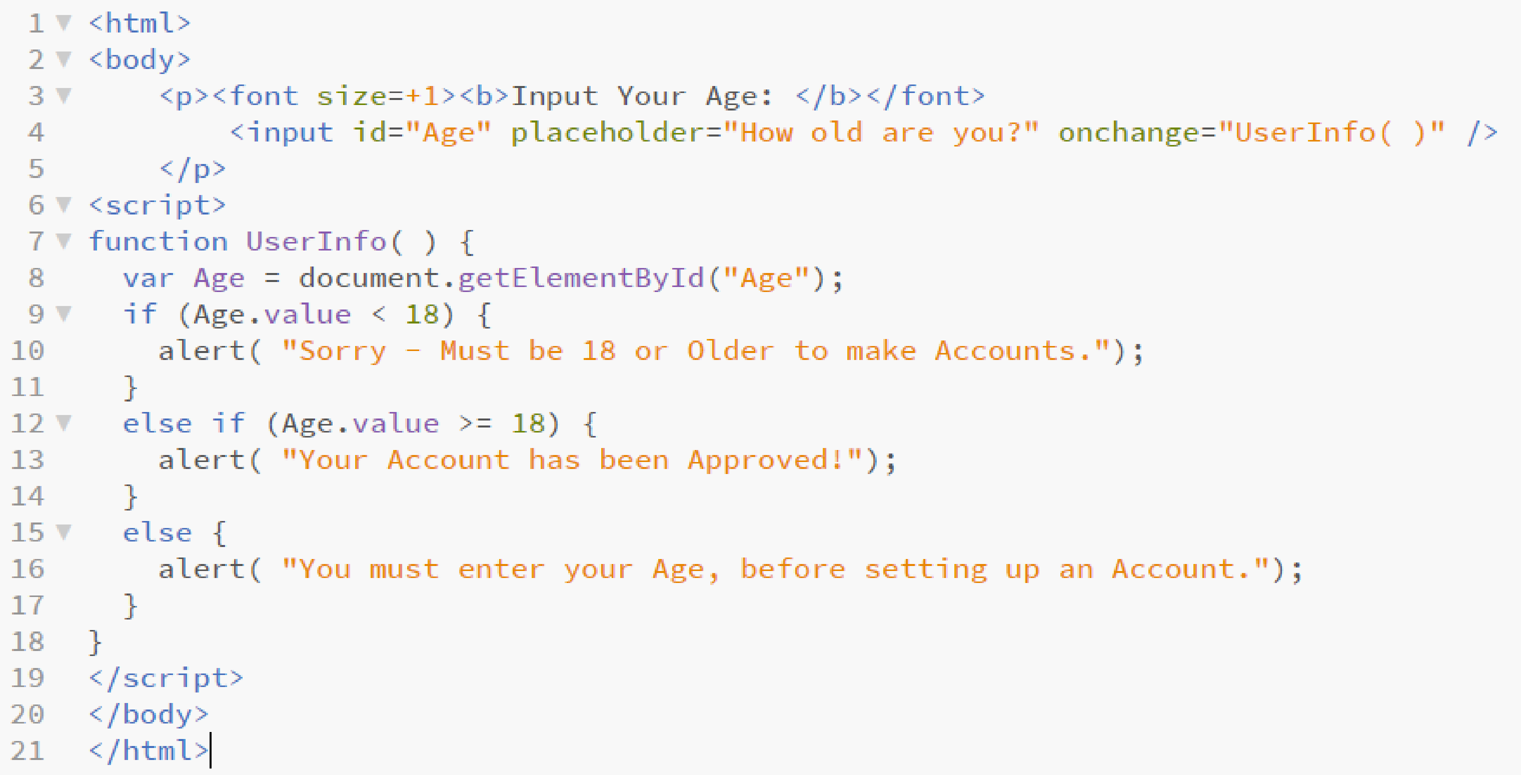
- Input Form for input box
- Output to alerts.
- <b></b>: bold face
- <font size=+1></font>: font-size increased

Project 6: Basic I/O

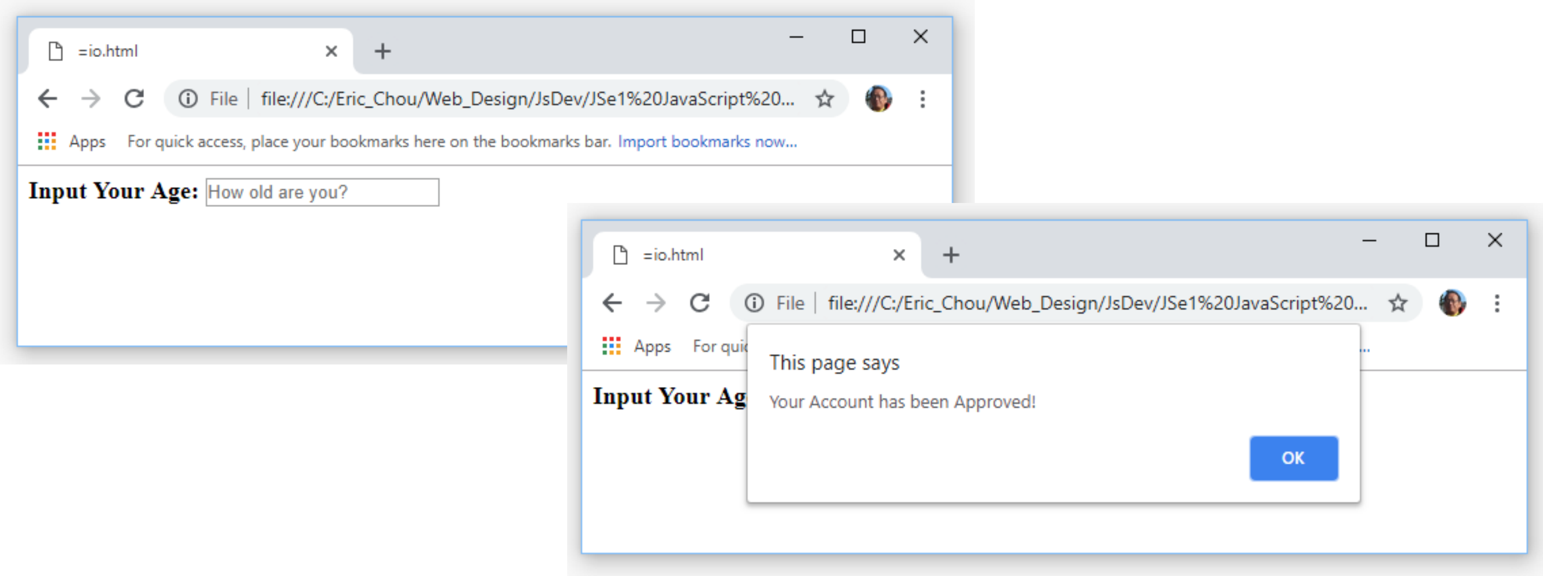
Demo Program: io.html

- A Conditional Statement in JavaScript begins with " if ", or " else if ", or " else ".
- " If " statements use Logic to check for true values of variables or class properties.
- " Else If " statements use Logic to check for other values that might also be true.
- " Else " statements use Logic to run a function task if there are no true values found for variables.

Project 6: Basic I/O



```
<html>
<body>
    <p><font size=+1><b>Input Your Age: </b></font>
        <input id="Age" placeholder="How old are you?" onchange="UserInfo( )" />
    </p>
<script>
function UserInfo( ) {
  var Age = document.getElementById("Age");
  if (Age.value < 18) {
    alert( "Sorry - Must be 18 or Older to make Accounts.");
  }
  else if (Age.value >= 18) {
    alert( "Your Account has been Approved!");
  }
  else {
    alert( "You must enter your Age, before setting up an Account.");
  }
}
</script>
</body>
</html>
```



=io.html

file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20...

For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Input Your Age: How old are you?

This page says

Your Account has been Approved!

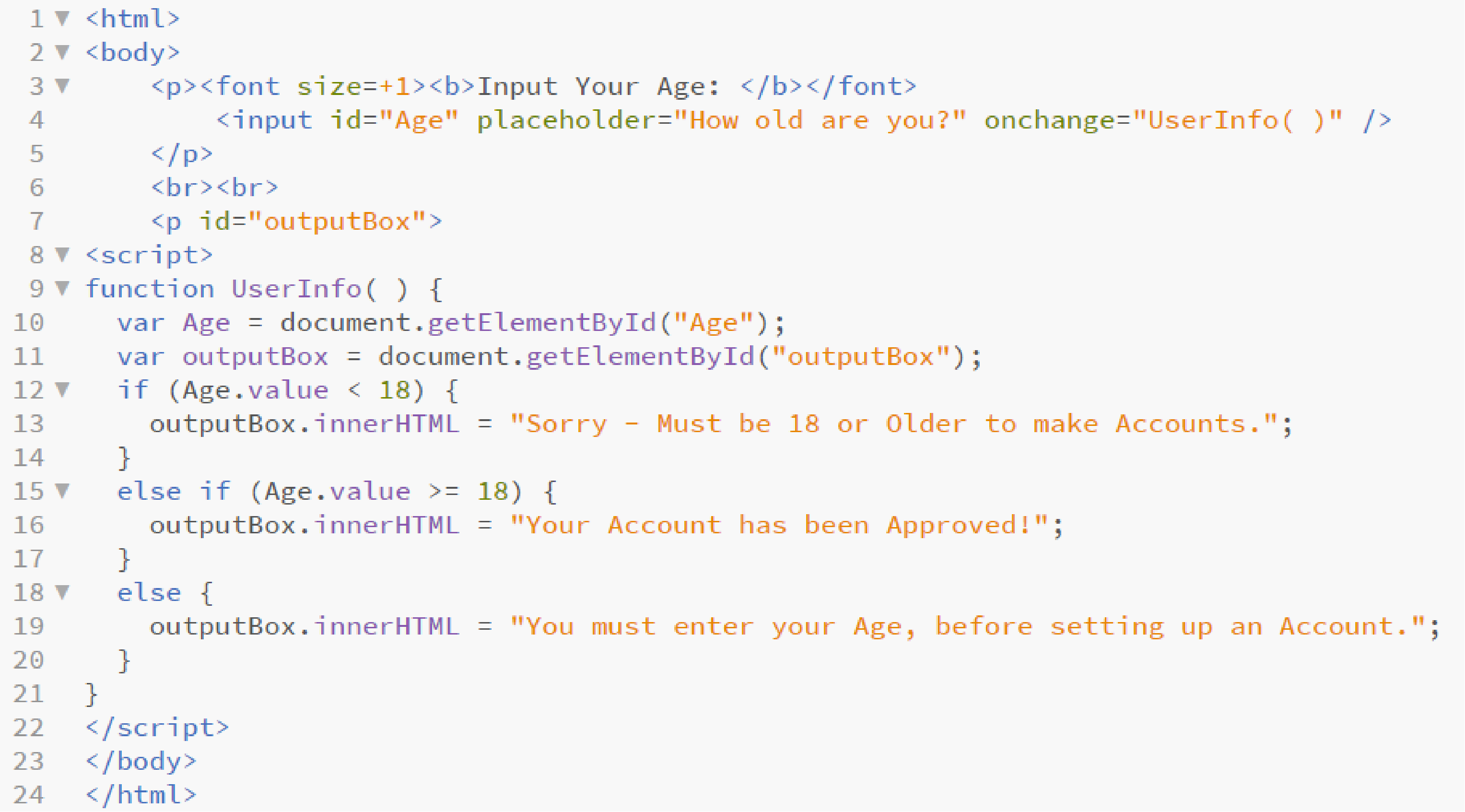
OK

Project 6: Basic I/O

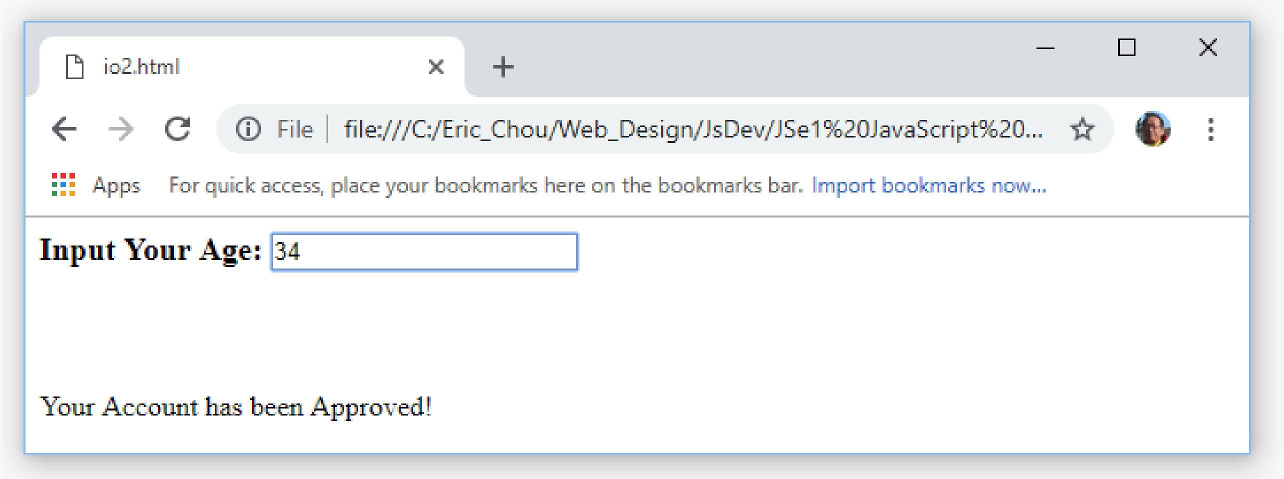
Demo Program: io2.html

- Input Form for input box
- Paragraph <p id=“output_text_label_name”></p>

Project 6: Basic I/O



```
<html>
<body>
    <p><font size=+1><b>Input Your Age: </b></font>
        <input id="Age" placeholder="How old are you?" onchange="UserInfo( )" />
    </p>
    <br><br>
    <p id="outputBox">
<script>
function UserInfo( ) {
  var Age = document.getElementById("Age");
  var outputBox = document.getElementById("outputBox");
  if (Age.value < 18) {
    outputBox.innerHTML = "Sorry - Must be 18 or Older to make Accounts.";
  }
  else if (Age.value >= 18) {
    outputBox.innerHTML = "Your Account has been Approved!";
  }
  else {
    outputBox.innerHTML = "You must enter your Age, before setting up an Account.";
  }
}
</script>
</body>
</html>
```


io2.html
File
file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20...
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...

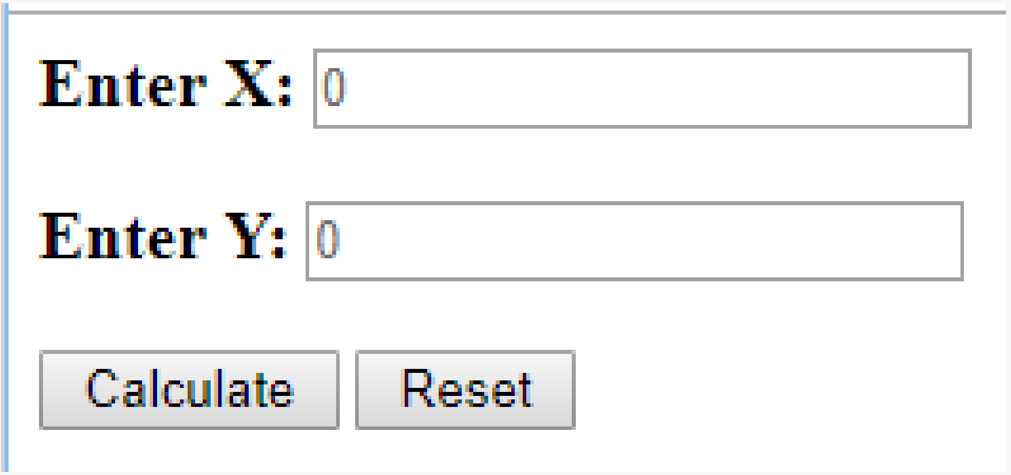
Input Your Age: 34

Your Account has been Approved!

Project 6: Basic I/O

Demo Program: io3.html

- 2 input forms for variable x, y
- 3 text paragraphs for outputs.
- 2 buttons for Calculate and Reset
- 4 handler for getX(), getY(), Calculate() and reset().



Enter X: 0

Enter Y: 0

```
<body>
    <p><font size=+1><b>Enter X: </b></font>
        <input id="XX" placeholder="0" onchange="getX( )" /><br><br>
       <font size=+1><b>Enter Y: </b></font>
        <input id="YY" placeholder="0" onchange="getY( )" /><br><br>
        <button onclick="Calculate()">&nbsp;Calculate&nbsp;</button>
        <button onclick="reset()">&nbsp;Reset&nbsp;</button>
    </p>
    <br><br>
    <p id="XXX"></p>
    <p id="YYY"></p>
    <p id="outputBox"></p>
</body>
```

```
    <script>
        // data model
        var x=0;
        var y=0;
        // handler x changed
        function getX( ) {
            var xstr = document.getElementById("XX").value;
                x = parseInt(xstr);
            var outputBox = document.getElementById("XXX");
                outputBox.innerHTML = "X is "+x;
        }
        // handler y changed
        function getY( ) {
            var ystr = document.getElementById("YY").value;
                y = parseInt(ystr);
            var outputBox = document.getElementById("YYY");
                outputBox.innerHTML = "Y is "+y;
        }
        function Calculate(){
            var outputBox = document.getElementById("outputBox");
            var z = x + y;
            outputBox.innerHTML = "The sum is "+z;
        }
        function reset(){
            var outputBox = document.getElementById("outputBox");
                outputBox.innerHTML = ""; z=0;
                outputBox = document.getElementById("XXX");
                outputBox.innerHTML = ""; x=0;
                document.getElementById("XX").value = "0";
                outputBox = document.getElementById("YYY");
                outputBox.innerHTML = ""; y=0;
                document.getElementById("YY").value = "0";
        }
</script>
```

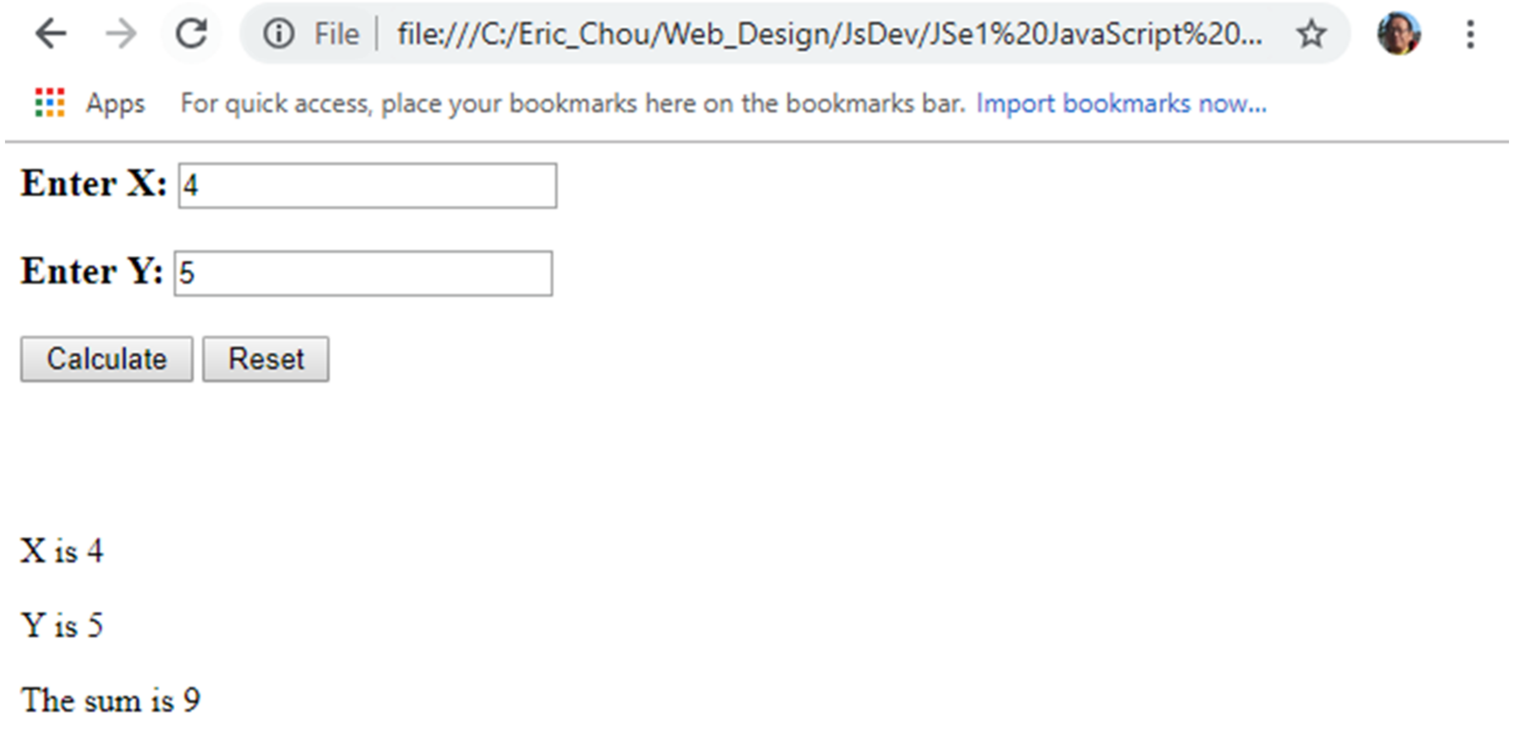
Project 4: Unicode

Demo Program:
unicode.html

Project 6: Basic I/O

Demo Program: io3.html

- 2 input forms for variable x, y
- 3 text paragraphs for outputs.
- 2 buttons for Calculate and Reset
- 4 handler for getX(), getY(), Calculate() and reset().


File
file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20...
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Calculate
Reset

Enter X: 4

Enter Y: 5

X is 4

Y is 5

The sum is 9

Project 6: Basic I/O

Project 7: Variable Data Types

Demo Program: uainfo1.html, uainfo2.html, uainfo3.html

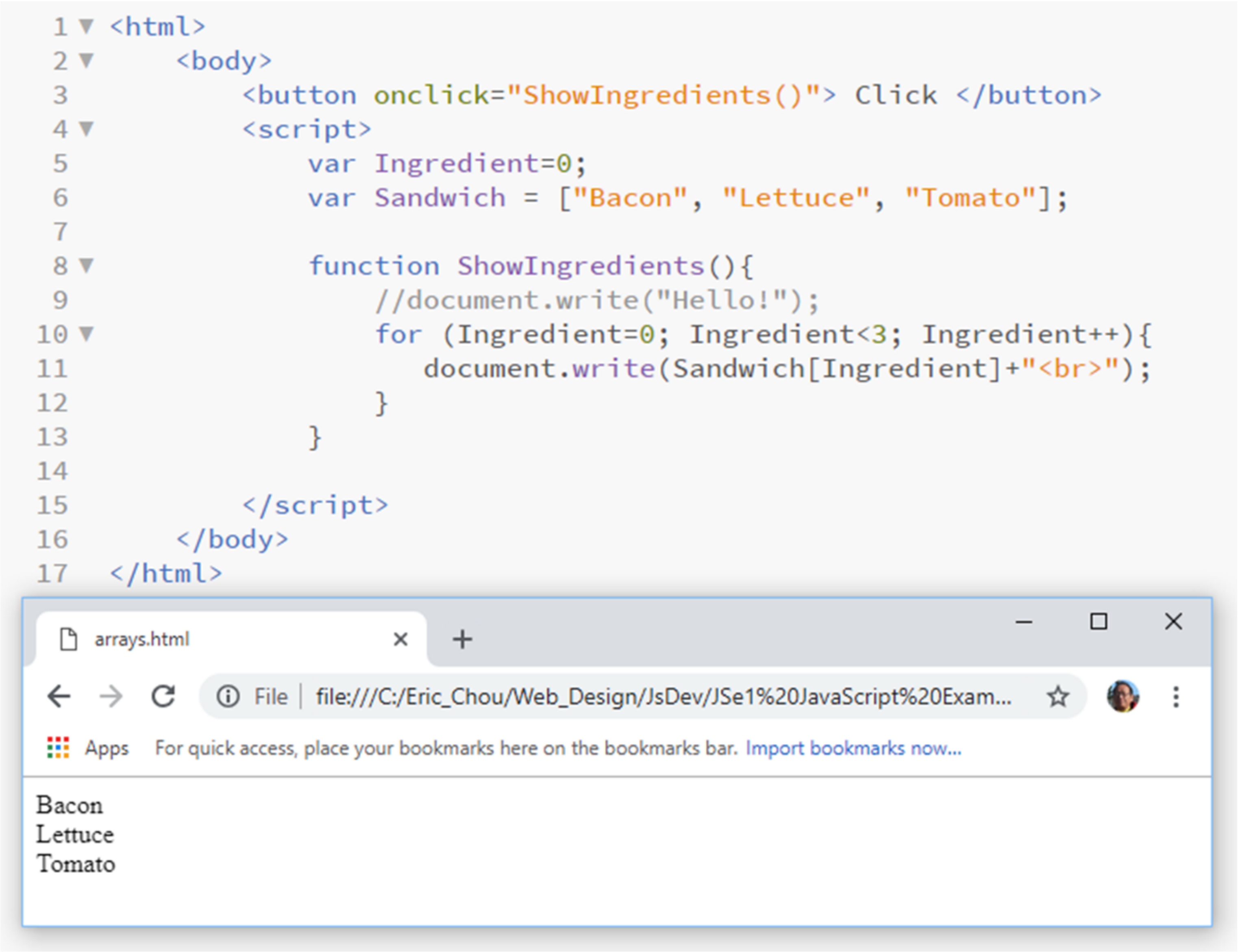
- There are number type, string type and boolean type in JavaScript language.
- In uainfo1.html: we demonstrate a case involves number.
- In uainfo2.html: we demonstrate a case involves a character (string).
- In uainfo3.html: we demonstrate a case involves a string.

Project 8: Array

Demo Program: arrays.html

- Array data representation is exactly the same as list in Python language.
- The for-loop variable can be pre-declared variable or var type number.

Project 8: Array



```
<html>
    <body>
        <button onclick="ShowIngredients()"> Click </button>
        <script>
            var Ingredient=0;
            var Sandwich = ["Bacon", "Lettuce", "Tomato"];

            function ShowIngredients(){
                //document.write("Hello!");
                for (Ingredient=0; Ingredient<3; Ingredient++){
                   document.write(Sandwich[Ingredient]+"<br>");
                }
            }

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

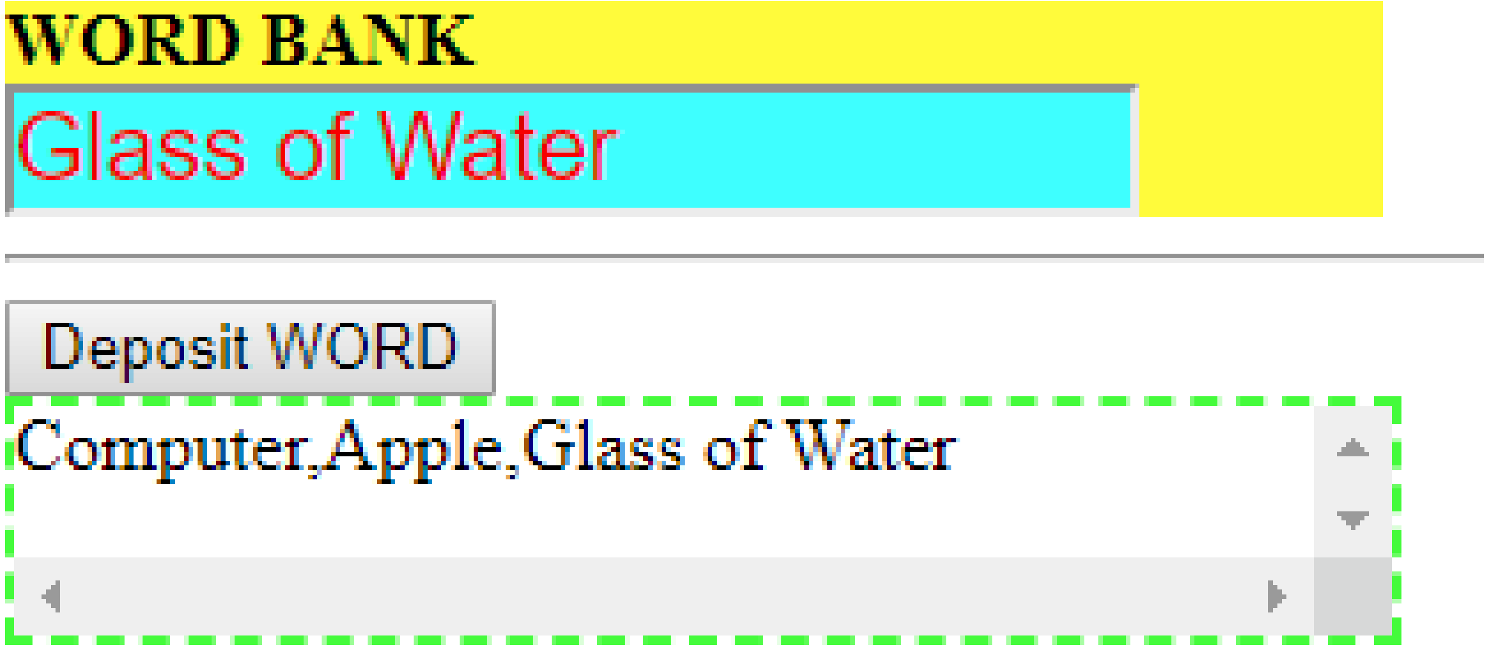
arrays.html
File
file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20Exam...
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

```
Bacon
Lettuce
Tomato
```

Project 9: Array as List

Demo Program: arrayList.html

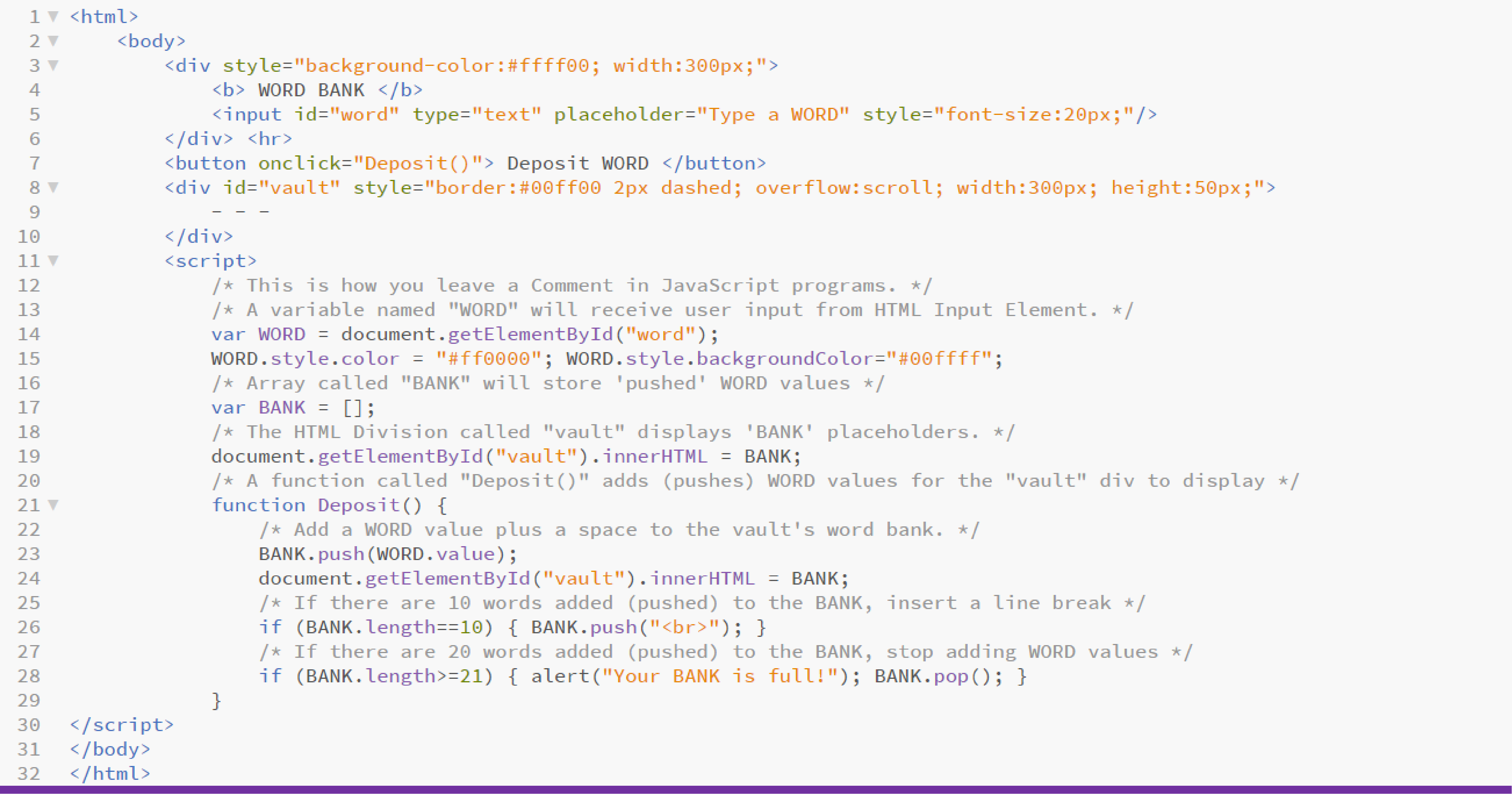
- Array in JavaScript language has the arraylist feature. So, it has the random access features and it also has the serial (iterator) access features.
- push() and pop() function (like stacks) are used from JavaScript language as the data addition and removal functions.



WORD BANK

Glass of Water

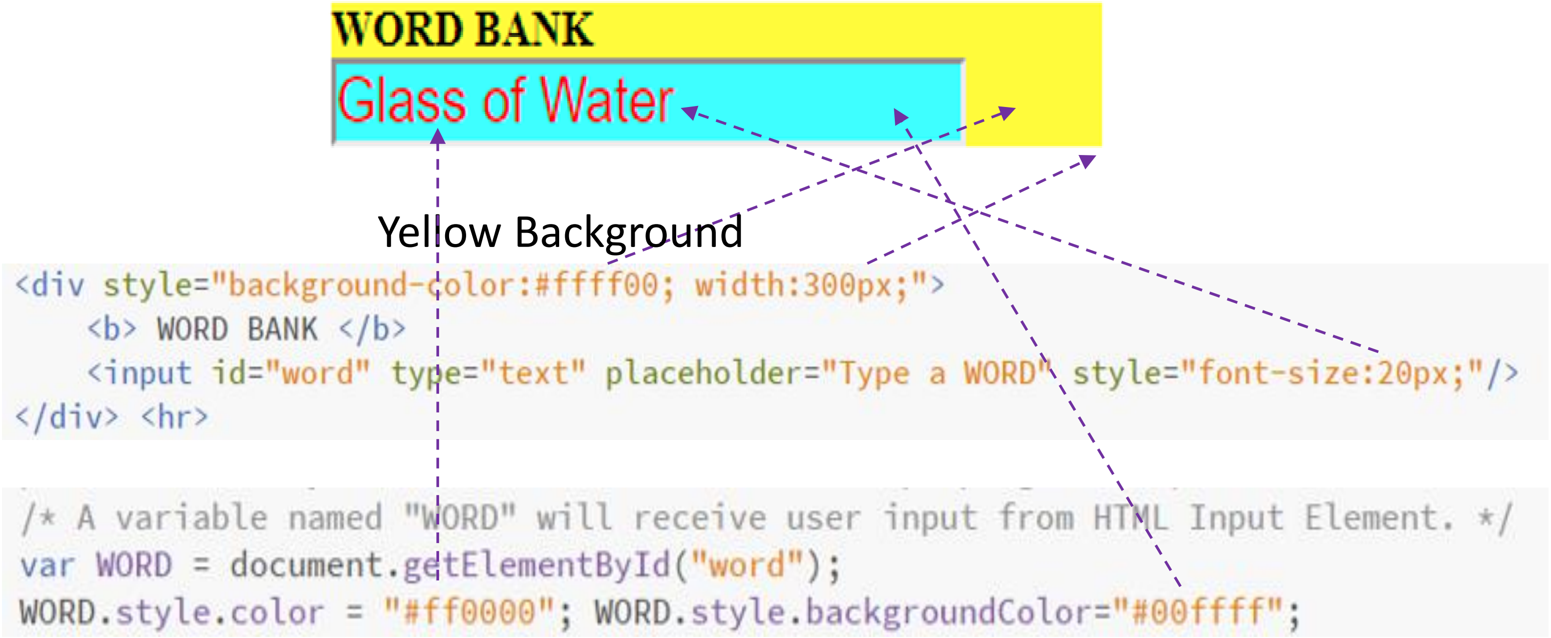
Computer,Apple,Glass of Water



```
<html>
    <body>
        <div style="background-color:#ffff00; width:300px;">
            <b> WORD BANK </b>
            <input id="word" type="text" placeholder="Type a WORD" style="font-size:20px;"/>
        </div> <hr>
        <button onclick="Deposit()"> Deposit WORD </button>
        <div id="vault" style="border:#00ff00 2px dashed; overflow:scroll; width:300px; height:50px;">
            - - -
        </div>
        <script>
            /* This is how you leave a Comment in JavaScript programs. */
            /* A variable named "WORD" will receive user input from HTML Input Element. */
            var WORD = document.getElementById("word");
            WORD.style.color = "#ff0000"; WORD.style.backgroundColor="#00ffff";
            /* Array called "BANK" will store 'pushed' WORD values */
            var BANK = [];
            /* The HTML Division called "vault" displays 'BANK' placeholders. */
            document.getElementById("vault").innerHTML = BANK;
            /* A function called "Deposit()" adds (pushes) WORD values for the "vault" div to display */
            function Deposit() {
                /* Add a WORD value plus a space to the vault's word bank. */
                BANK.push(WORD.value);
                document.getElementById("vault").innerHTML = BANK;
                /* If there are 10 words added (pushed) to the BANK, insert a line break */
                if (BANK.length==10) { BANK.push("<br>"); }
                /* If there are 20 words added (pushed) to the BANK, stop adding WORD values */
                if (BANK.length>=21) { alert("Your BANK is full!"); BANK.pop(); }
            }
</script>
</body>
</html>
```

Project 9: Array as List

Demo Program: arrayList.html



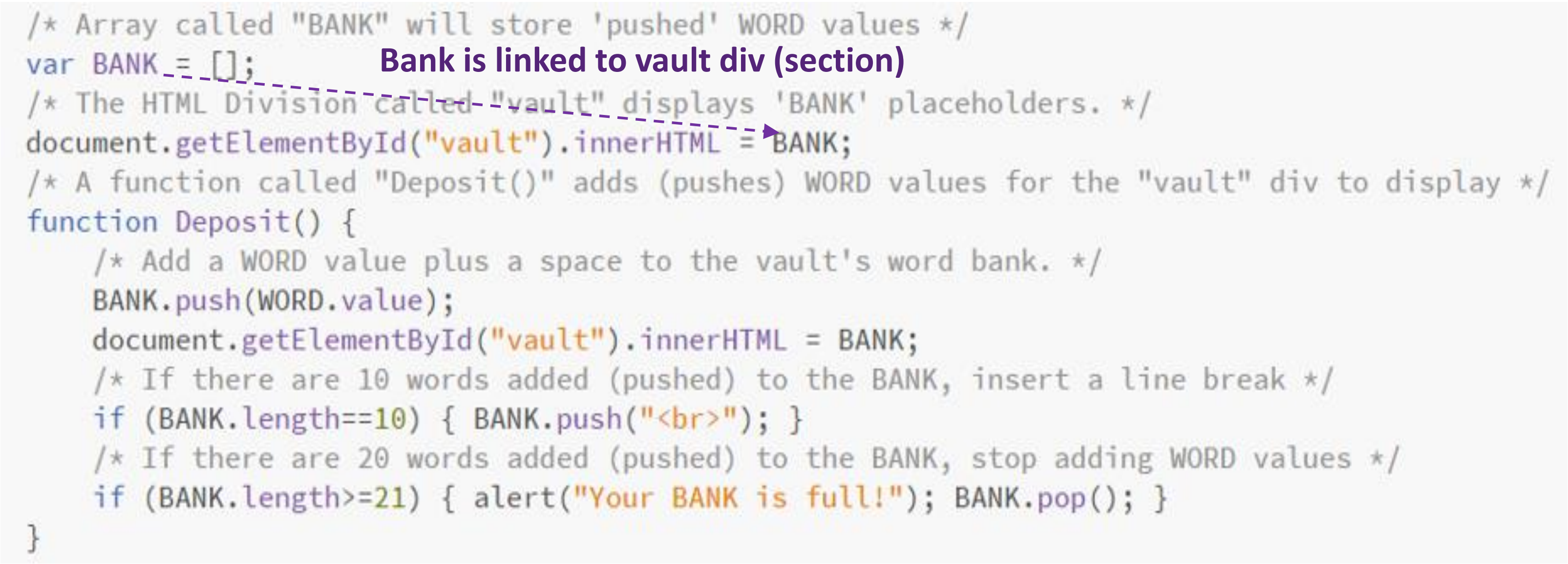
Yellow Background

```
<div style="background-color:#ffff00; width:300px;">
    <b> WORD BANK </b>
    <input id="word" type="text" placeholder="Type a WORD" style="font-size:20px;"/>
</div> <hr>
```

```
/* A variable named "WORD" will receive user input from HTML Input Element. */
var WORD = document.getElementById("word");
WORD.style.color = "#ff0000"; WORD.style.backgroundColor="#00ffff";
```

Project 9: Array as List

Demo Program: arrayList.html



Bank is linked to vault div (section)

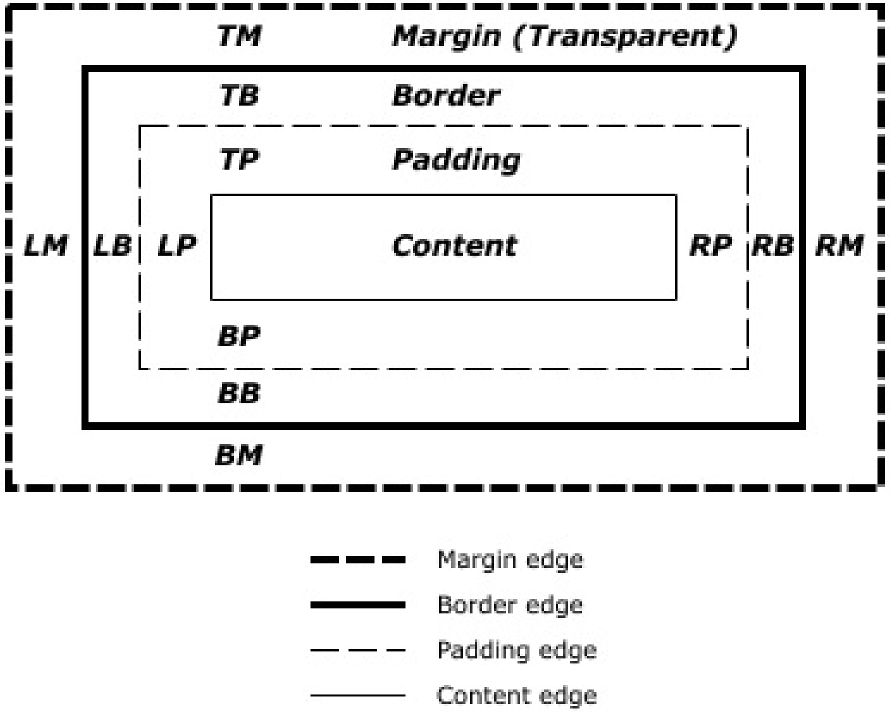
```
/* Array called "BANK" will store 'pushed' WORD values */
var BANK = [];
/* The HTML Division called "vault" displays 'BANK' placeholders. */
document.getElementById("vault").innerHTML = BANK;
/* A function called "Deposit()" adds (pushes) WORD values for the "vault" div to display */
function Deposit() {
    /* Add a WORD value plus a space to the vault's word bank. */
    BANK.push(WORD.value);
    document.getElementById("vault").innerHTML = BANK;
    /* If there are 10 words added (pushed) to the BANK, insert a line break */
    if (BANK.length==10) { BANK.push("<br>"); }
    /* If there are 20 words added (pushed) to the BANK, stop adding WORD values */
    if (BANK.length>=21) { alert("Your BANK is full!"); BANK.pop(); }
}
```

Graphics and Animation

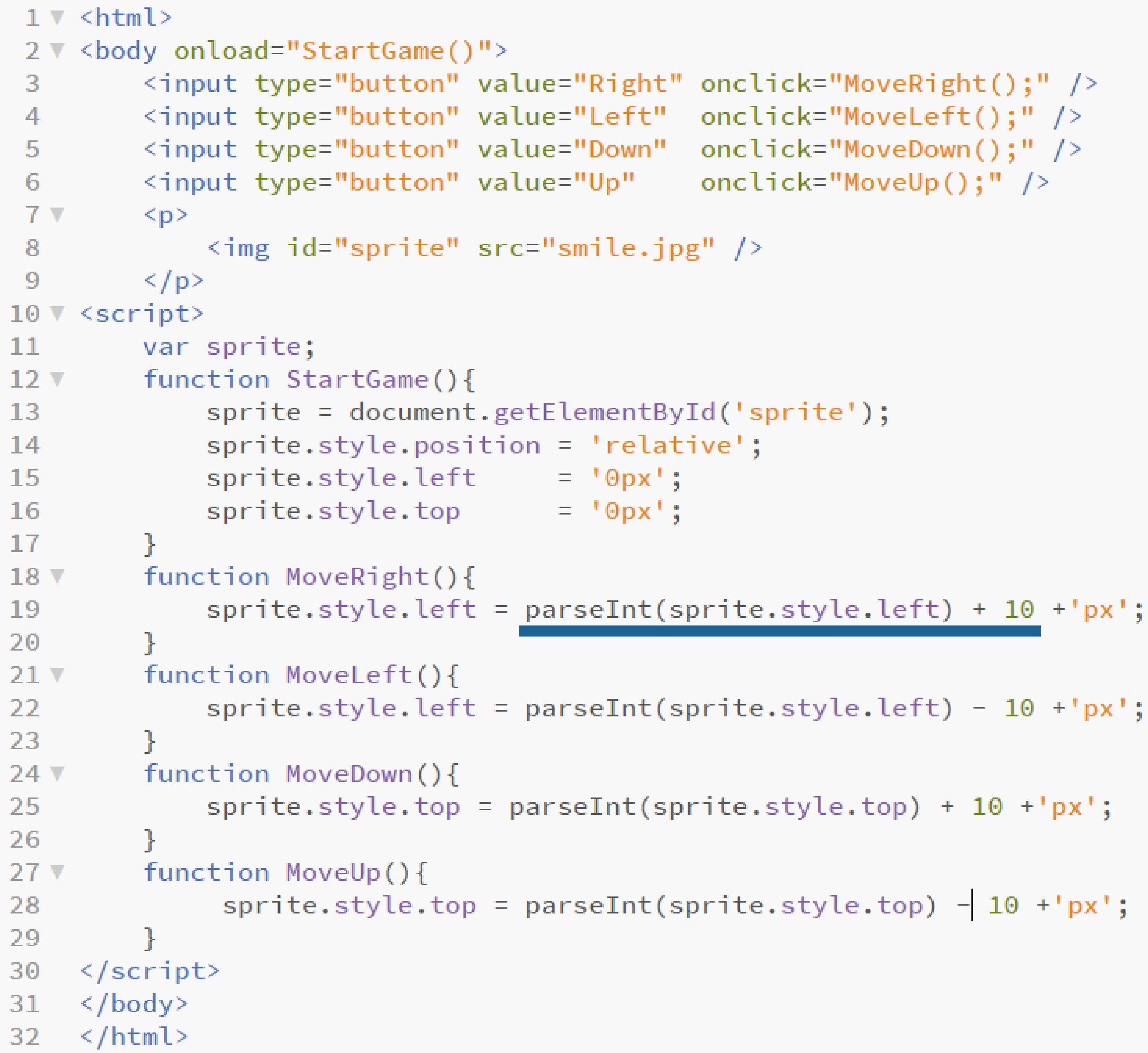
SECTION 1

Project 10: Moving a DOM Object

Demo Program: smile.html



- A DOM object can be moved by updating its bounding box location (CSSOM)
- Updates made to the top/left coordinates of the content object will lead to the movement of the image block.



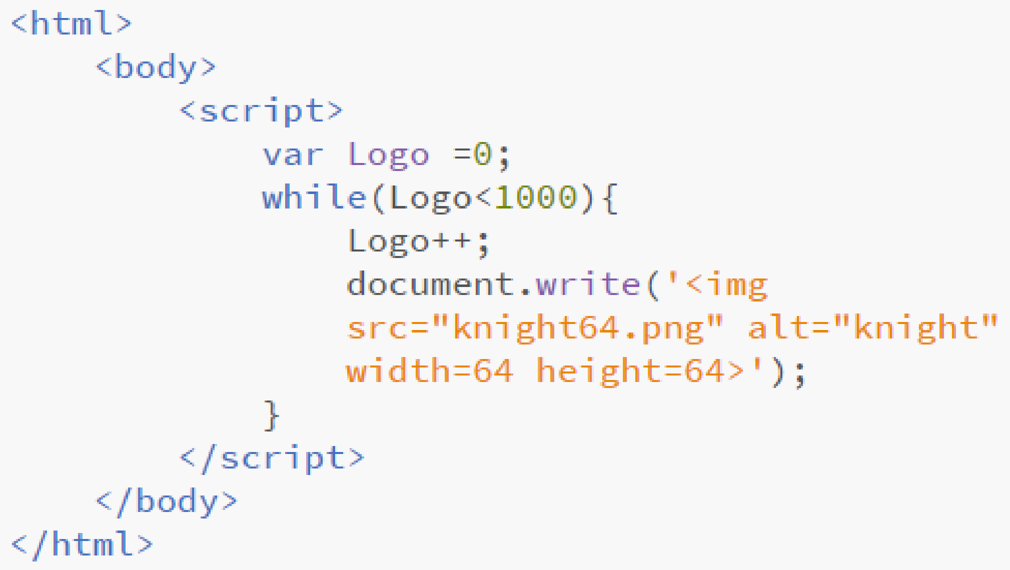
```html
<html>
<body onload="StartGame()">
    <input type="button" value="Right" onclick="MoveRight();" />
    <input type="button" value="Left"  onclick="MoveLeft();" />
    <input type="button" value="Down"  onclick="MoveDown();" />
    <input type="button" value="Up"    onclick="MoveUp();" />
    <p>
        <img id="sprite" src="smile.jpg" />
    </p>
<script>
    var sprite;
    function StartGame(){
        sprite = document.getElementById('sprite');
        sprite.style.position = 'relative';
        sprite.style.left     = '0px';
        sprite.style.top      = '0px';
    }
    function MoveRight(){
        sprite.style.left = parseInt(sprite.style.left) + 10 +'px';
    }
    function MoveLeft(){
        sprite.style.left = parseInt(sprite.style.left) - 10 +'px';
    }
    function MoveDown(){
        sprite.style.top = parseInt(sprite.style.top) + 10 +'px';
    }
    function MoveUp(){
         sprite.style.top = parseInt(sprite.style.top) - 10 +'px';
    }
</script>
</body>
</html>
```

Project 11: Motion

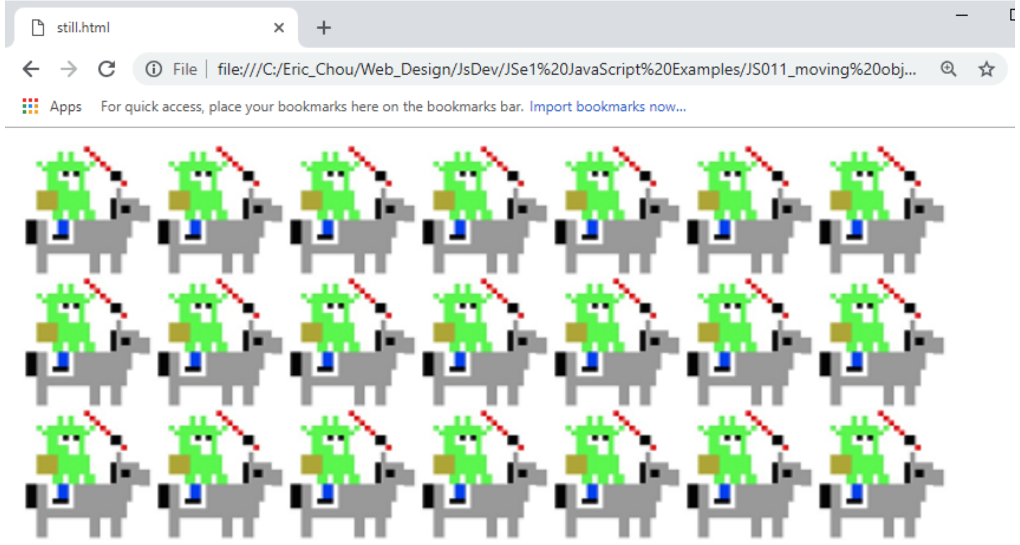
Demo Program: still.html

- An image can be place in “flow” layout format using a for-loop.
- document.write('<img src="knight64.png" alt="knight" width=64 height=64>‘);

```
                                    // each of these will place the image
                                    // once
```



```
<html>
    <body>
        <script>
            var Logo =0;
            while(Logo<1000){
                Logo++;
                document.write('<img
                src="knight64.png" alt="knight"
                width=64 height=64>');
            }
        </script>
    </body>
</html>
```



still.html

file:///C:/Eric_Chou/Web_Design/JsDev/JSe1%20JavaScript%20Examples/JS011_moving%20obj...

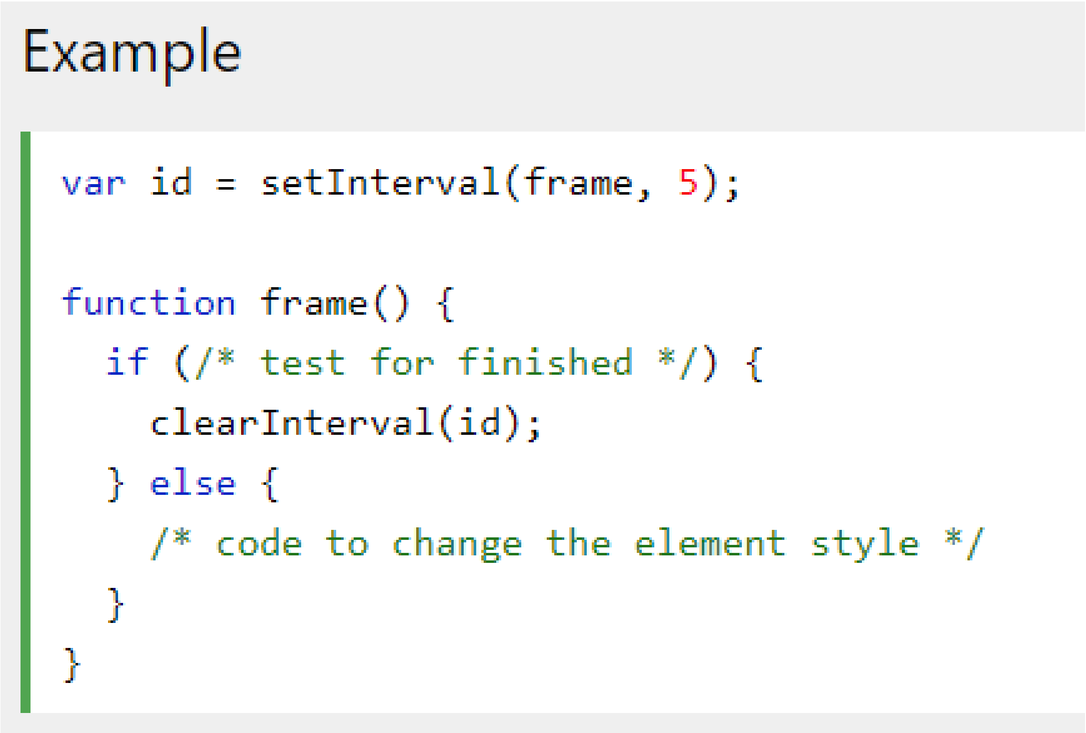
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Project 11: Motion

Demo Program: animation.html

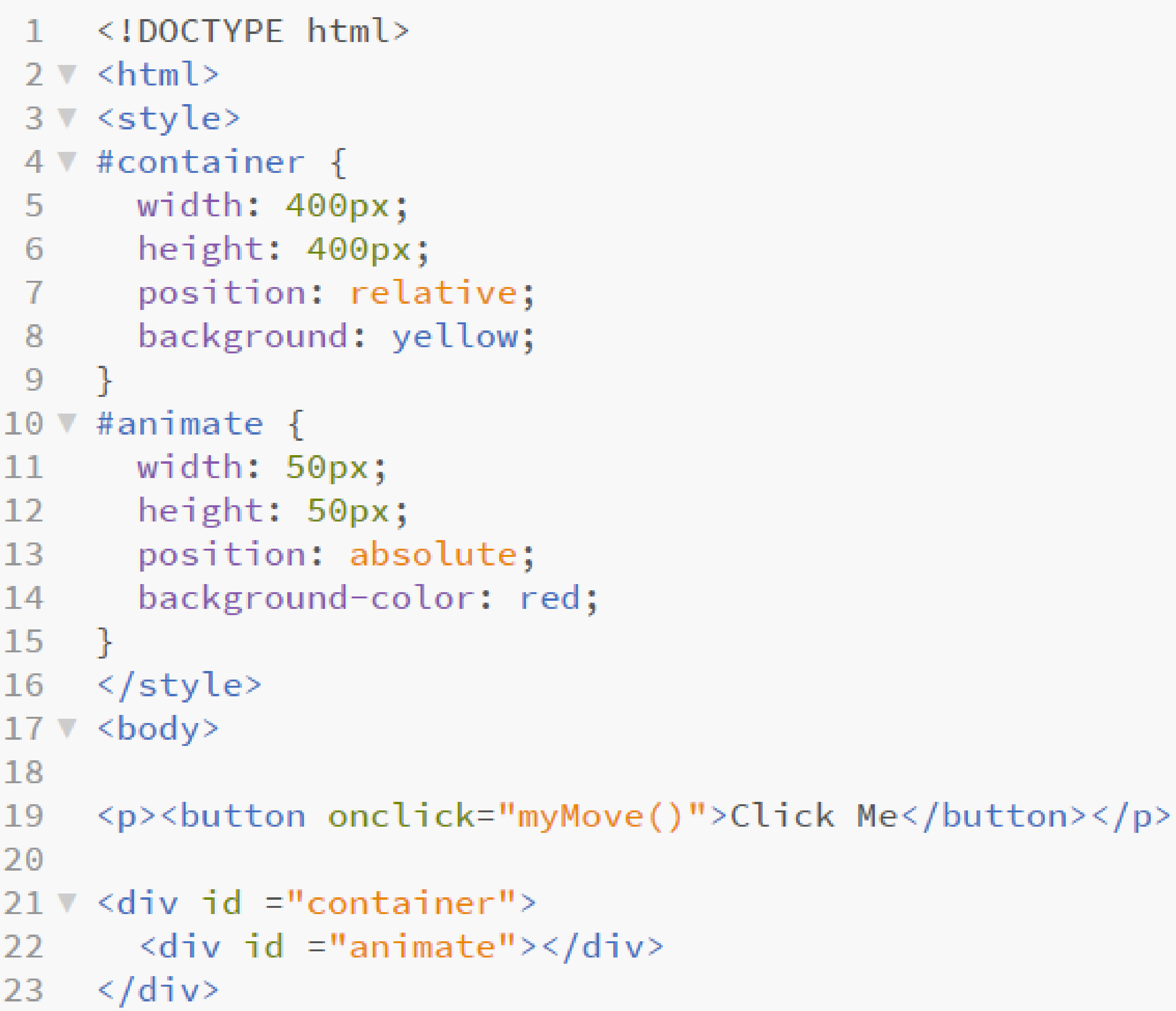
Animation Code

- JavaScript animations are done by programming gradual changes in an element's style.
- The changes are called by a **timer**. When the timer interval is small, the animation looks continuous.

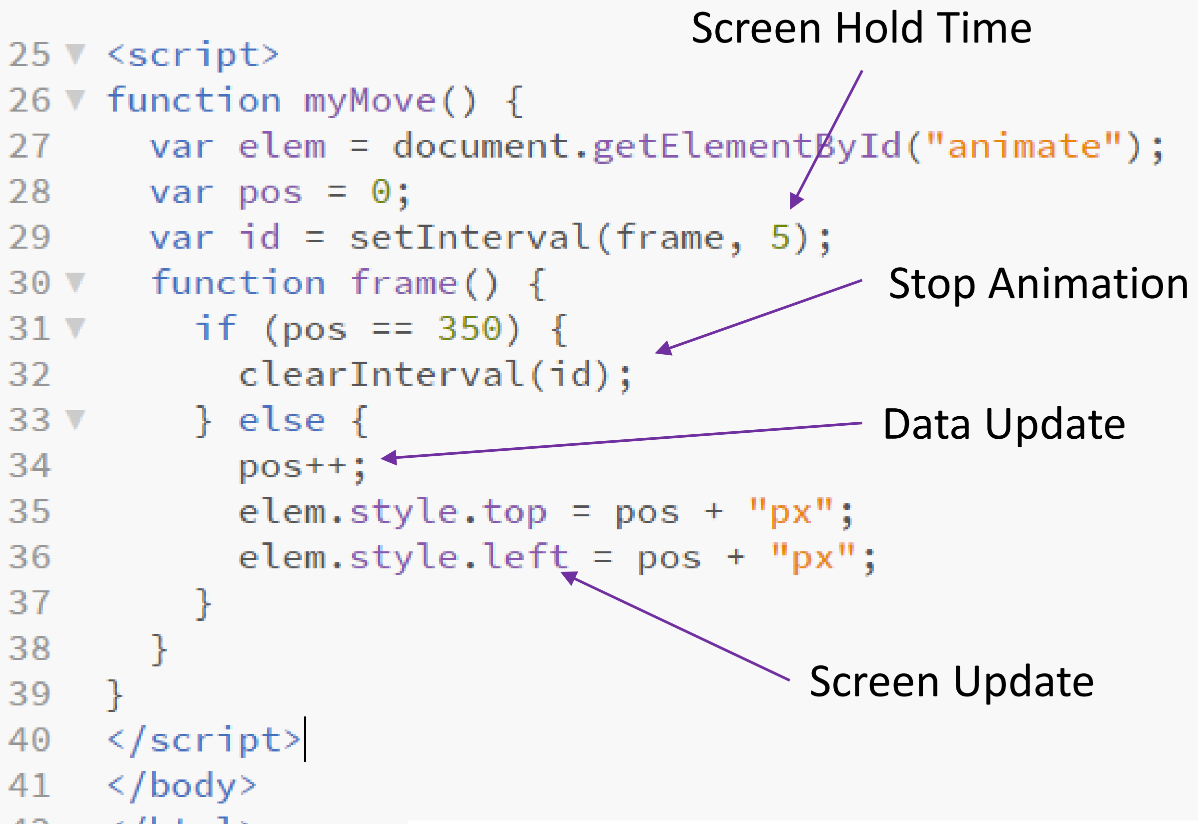


Example

```
var id = setInterval(frame, 5);

function frame() {
  if (/* test for finished */) {
    clearInterval(id);
  } else {
    /* code to change the element style */
  }
}
```



```
<!DOCTYPE html>
<html>
<style>
#container {
  width: 400px;
  height: 400px;
  position: relative;
  background: yellow;
}
#animate {
  width: 50px;
  height: 50px;
  position: absolute;
  background-color: red;
}
</style>
<body>

<p><button onclick="myMove()">Click Me</button></p>

<div id ="container">
  <div id ="animate"></div>
</div>
```



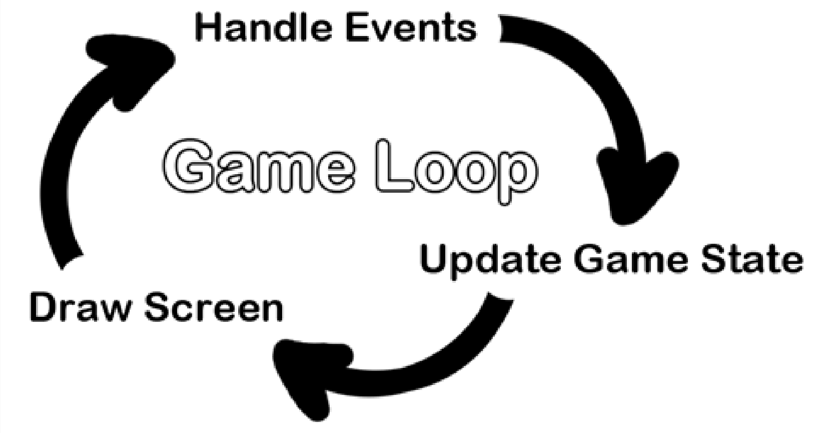
```
<script>
function myMove() {
  var elem = document.getElementById("animate");
  var pos = 0;
  var id = setInterval(frame, 5);
  function frame() {
    if (pos == 350) {
      clearInterval(id);
    } else {
      pos++;
      elem.style.top = pos + "px";
      elem.style.left = pos + "px";
    }
  }
}
</script>
</body>
</html>
```

Screen Hold Time

Stop Animation

Data Update

Screen Update

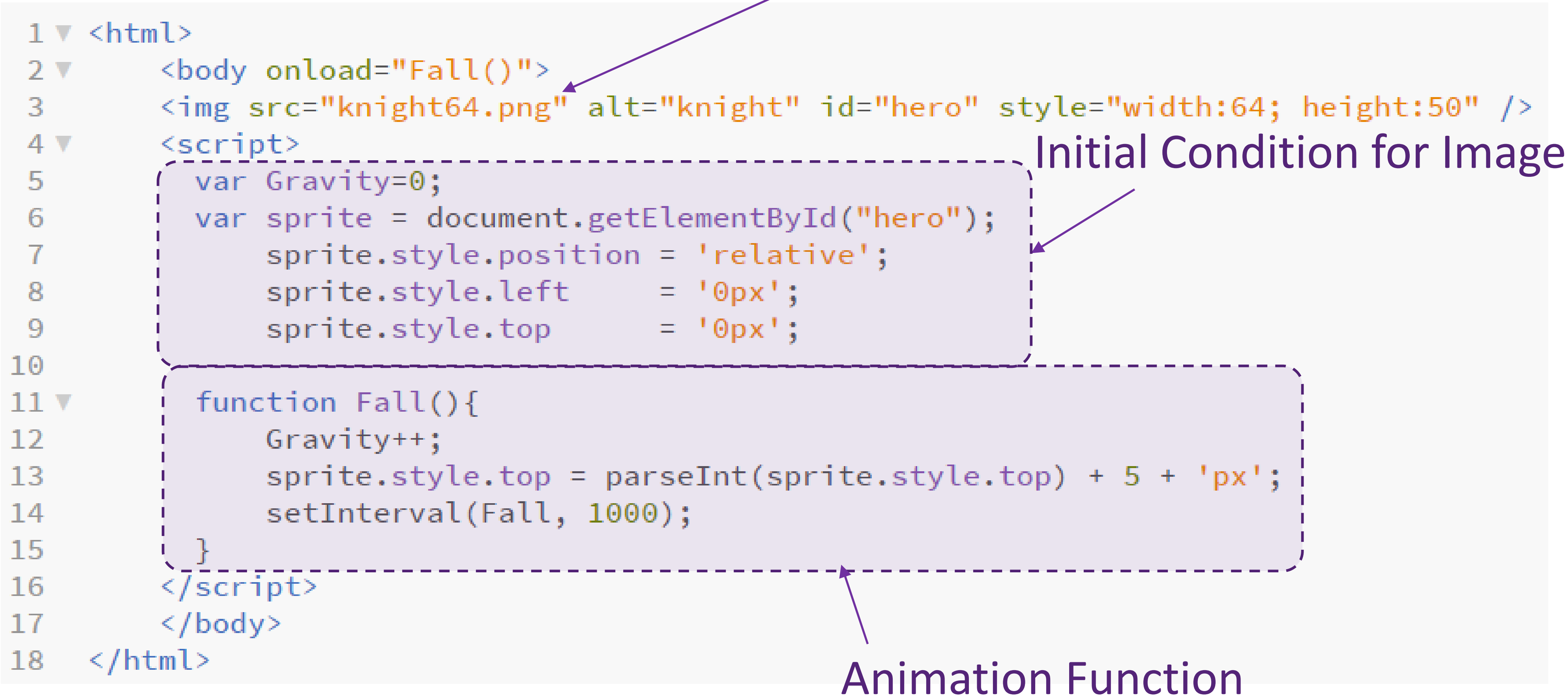


Project 11: Motion

Demo Program: move.html

Animation Code

- Simplified program from animation.html
- Present a complete html example for animation



Still Image

Initial Condition for Image

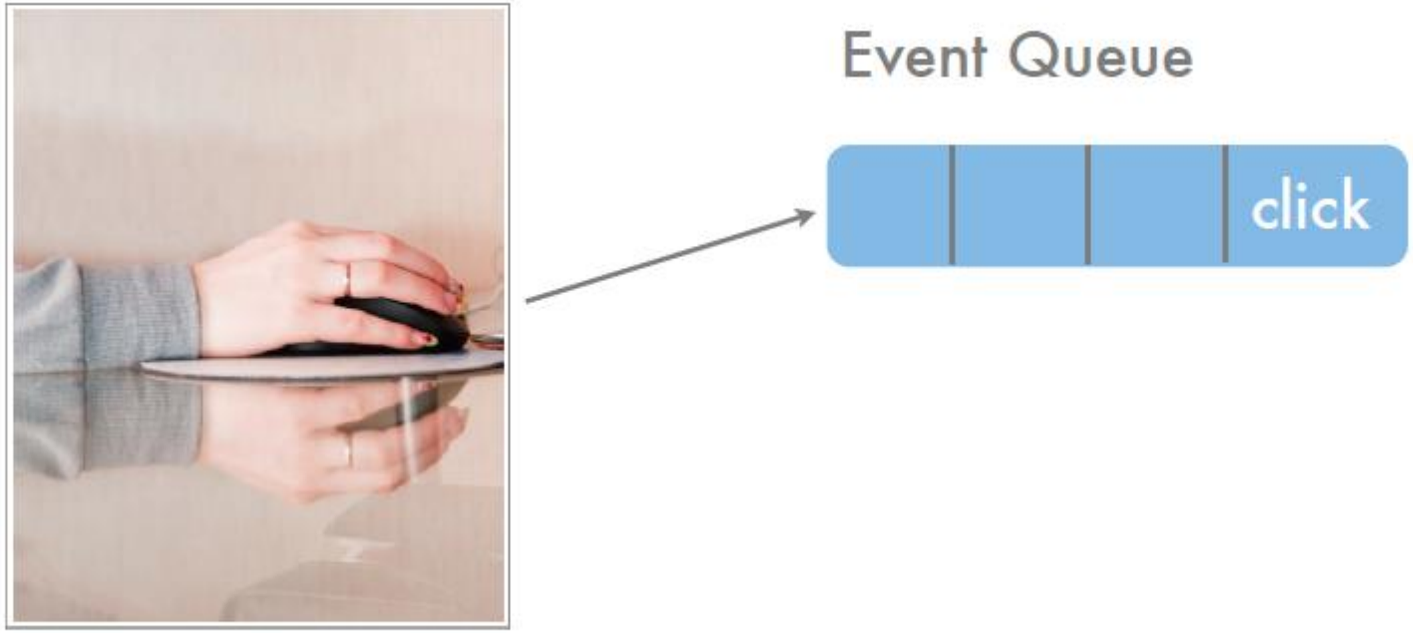
Animation Function

```
<html>
    <body onload="Fall()">
    <img src="knight64.png" alt="knight" id="hero" style="width:64; height:50" />
    <script>
      var Gravity=0;
      var sprite = document.getElementById("hero");
          sprite.style.position = 'relative';
          sprite.style.left     = '0px';
          sprite.style.top      = '0px';

      function Fall(){
          Gravity++;
          sprite.style.top = parseInt(sprite.style.top) + 5 + 'px';
          setInterval(Fall, 1000);
      }
    </script>
    </body>
</html>
```

Event-Driven Programming

SECTION 1

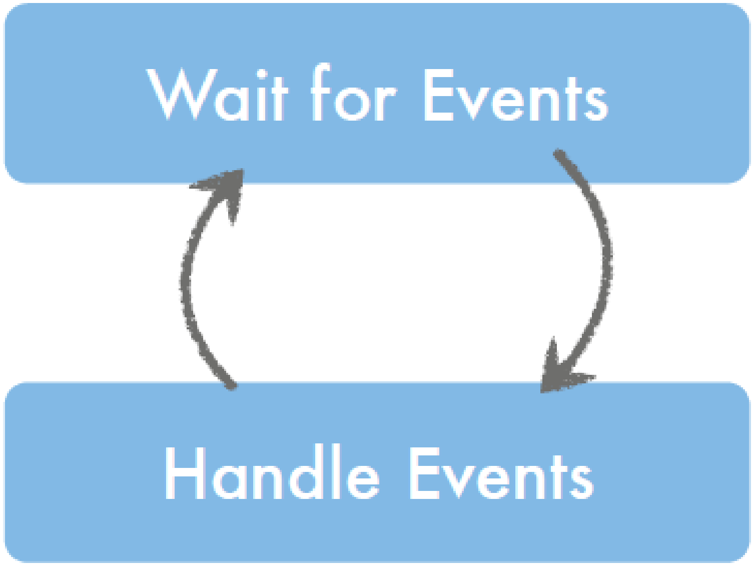
Browser Events Loop



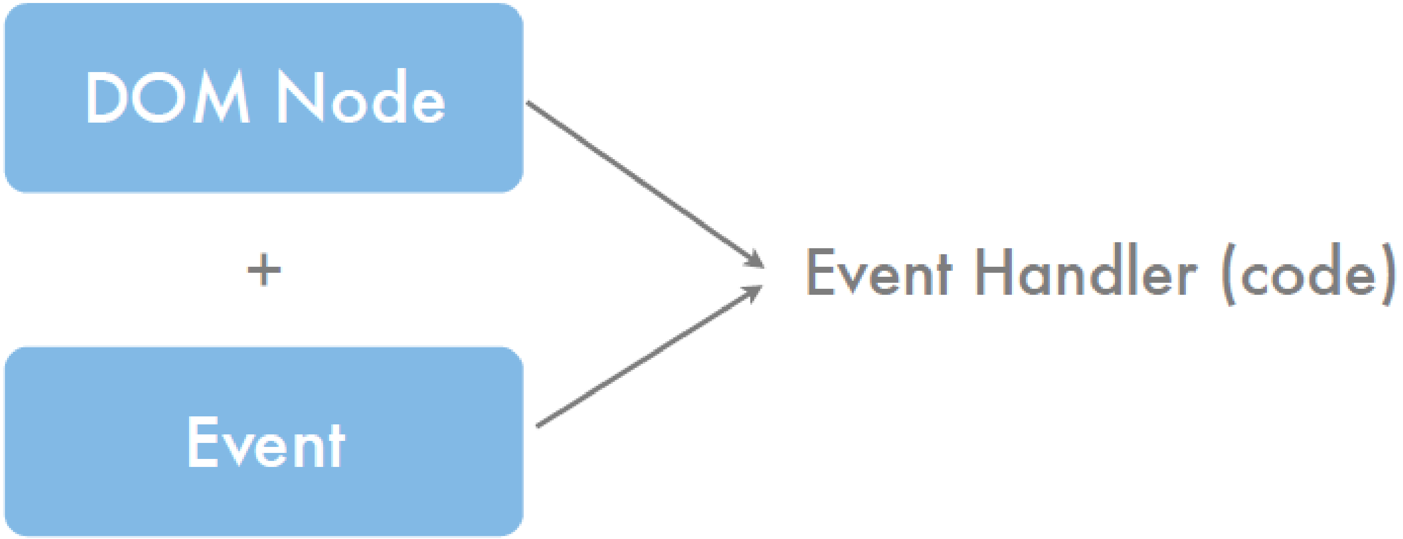
Event Queue

click

Event Loop



Event Handling



DOM Node

+

Event

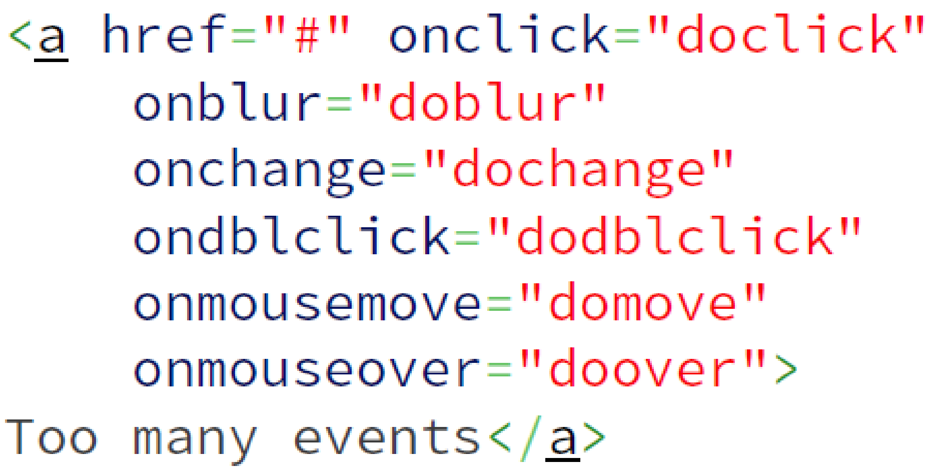
Event Handler (code)

Code Outline

- From HTML:
    - `<a on...=”handleEvent()”>`

Code Outline

- But this can get messy



```
<a href="#" onclick="doclick"
    onblur="doblur"
    onchange="dochange"
    ondblclick="dodblclick"
    onmousemove="domove"
    onmouseover="doover">
Too many events</a>
```

Code Outline

- From JS
  - Get a DOM node
  - Bind event to code

Getting DOM Nodes

- getElementById(...)
- getElementsByTagName(...)
- querySelector(...) - IE8 and up

Browser Events

- All browsers use:

  ```
  node.onevent = ...
  ```

- IE uses:

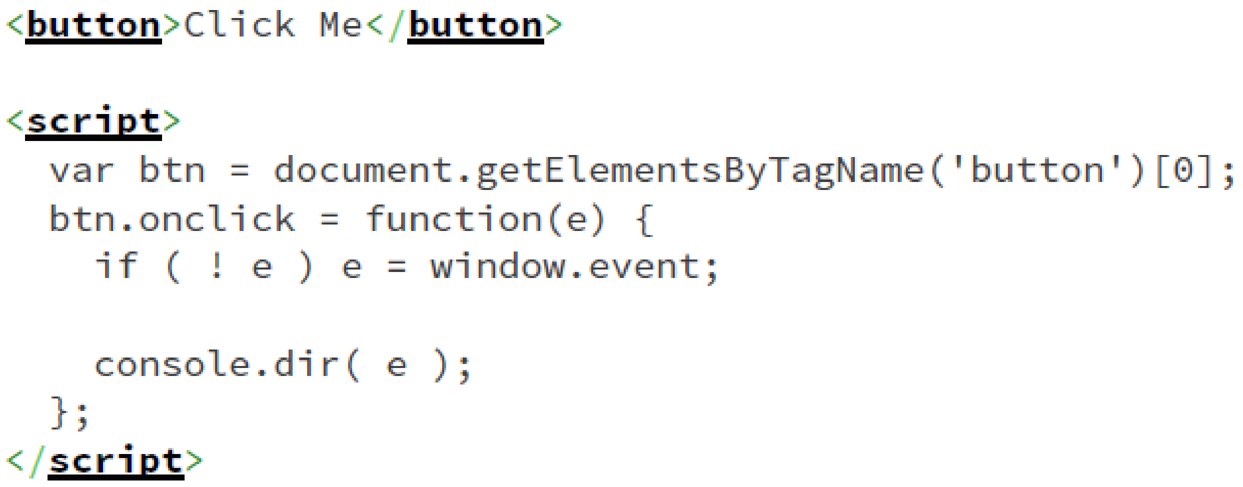
  ```
  node.attachEvent(...)
  ```

- Other browsers use

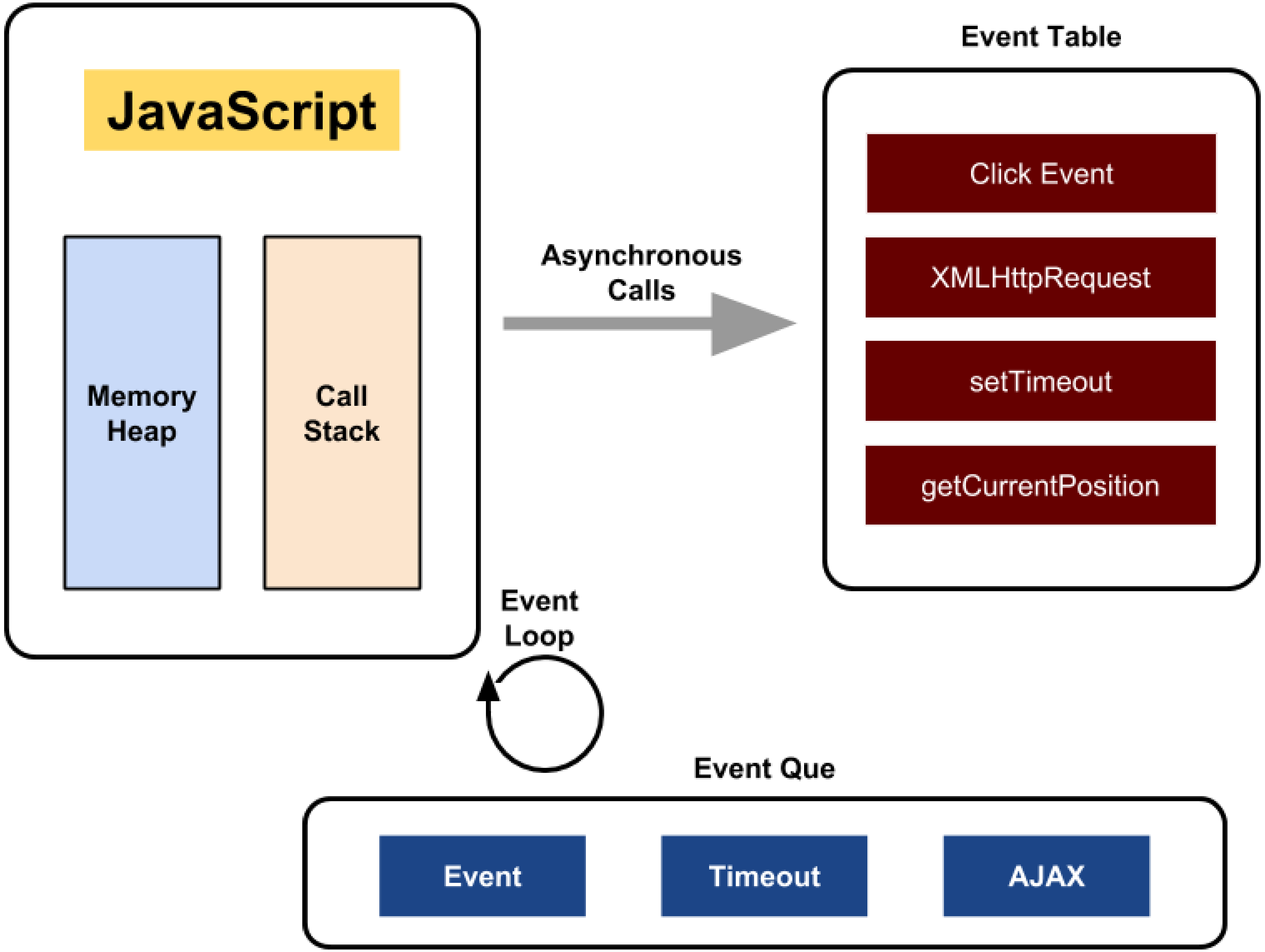
  ```
  node.addEventListener(...)
  ```

Using the Event Object

- Event object includes info on the event
- Print it to console for inspection



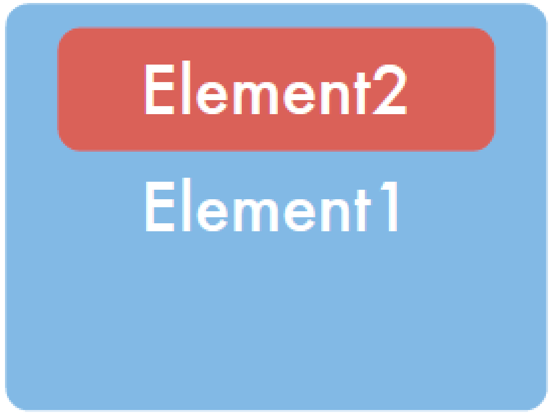
```
<button>Click Me</button>

<script>
  var btn = document.getElementsByTagName('button')[0];
  btn.onclick = function(e) {
    if ( ! e ) e = window.event;

    console.dir( e );
  };
</script>
```


JavaScript
Memory Heap
Call Stack
Asynchronous Calls
Event Table
Click Event
XMLHttpRequest
setTimeout
getCurrentPosition
Event Loop
Event Que
Event
Timeout
AJAX

Usages

- Default event handlers
- Dynamic event handlers

Double Handlers

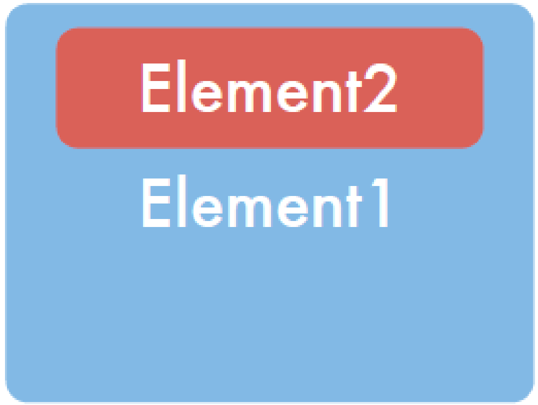


Element2

Element1

```
element1.onclick =
doSomething;

element2.onclick =
doSomething;
```

Double Handlers



```
function doSomething(e) {
      if ( ! e ) e = window.event;

      // this refers to
      // the current element

      // for inner event:
      // this = element2

      // for outer event:
      // this = element1
}
```

Event Types

Interface Events	Mouse Events	Form Events
load, unload	click, dblclick	submit
resize, scroll,	mousedown, mouseup, mousemove	reset
focus, blur	mouseover, mouseout	

Default Action

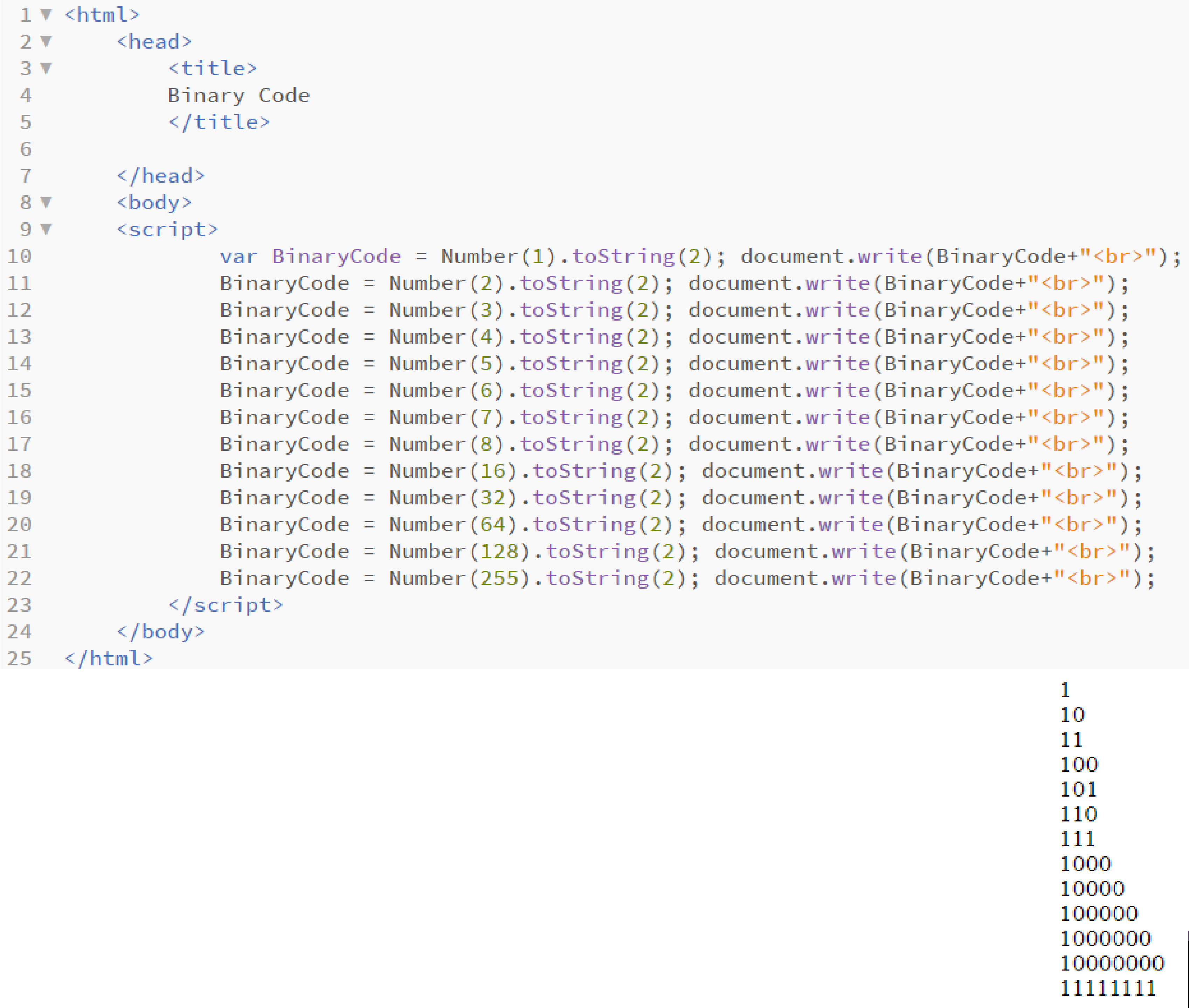
- Some events also have a “default” action
- For example: A link will take you to another page by default

Default Action

- Possible to prevent
- `return false` from handler
- Demo

Project 12: Number and ASCII

Demo Program: ascii.html



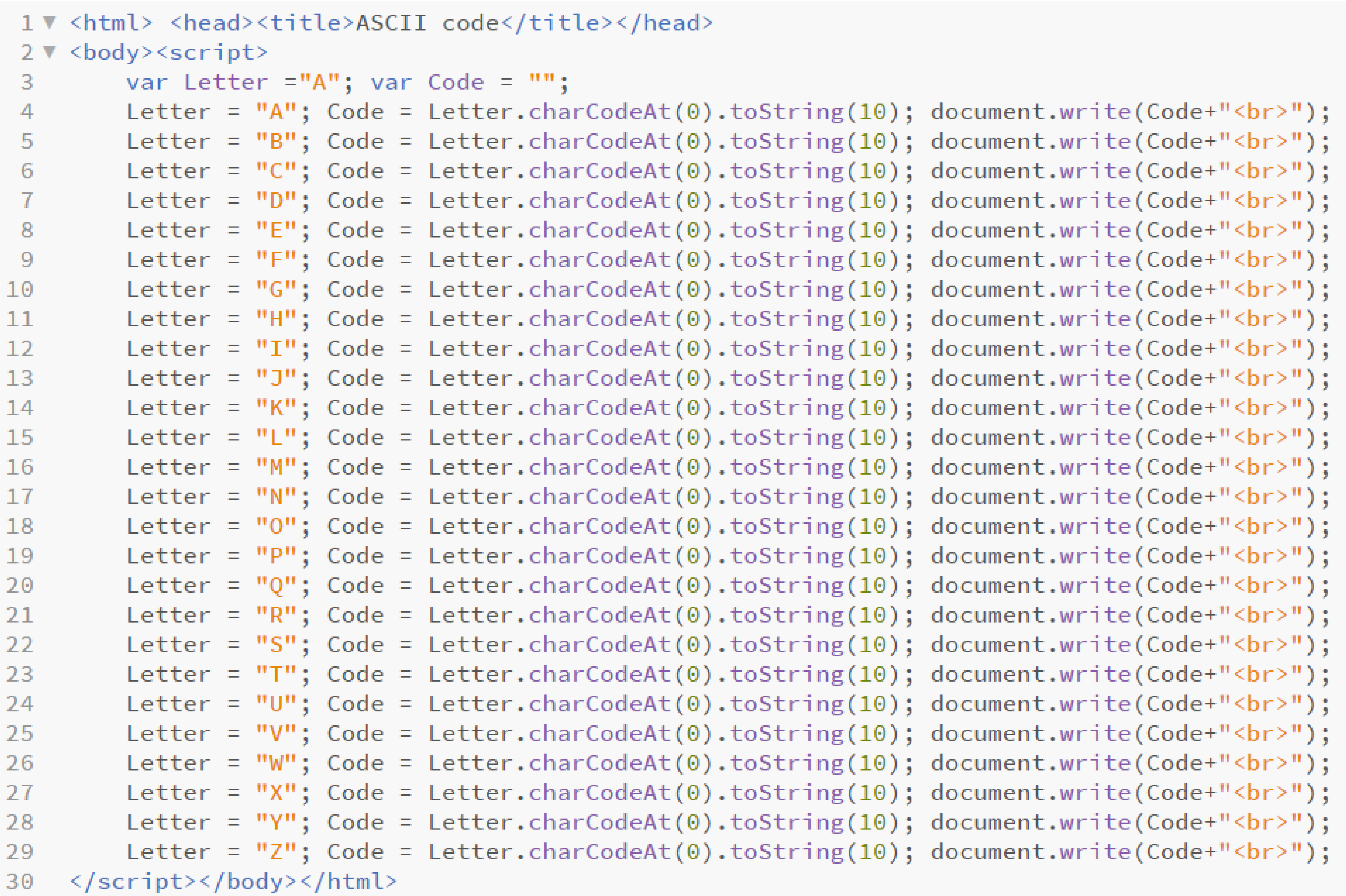
```
<html>
    <head>
        <title>
        Binary Code
        </title>

    </head>
    <body>
    <script>
            var BinaryCode = Number(1).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(2).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(3).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(4).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(5).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(6).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(7).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(8).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(16).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(32).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(64).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(128).toString(2); document.write(BinaryCode+"<br>");
            BinaryCode = Number(255).toString(2); document.write(BinaryCode+"<br>");
        </script>
    </body>
</html>
```

```
1
10
11
100
101
110
111
1000
10000
100000
1000000
10000000
11111111
```

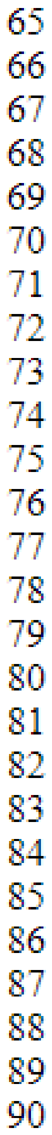
Project 12: Number and ASCII

Demo Program: ascii1.html

- Convert a symbol to its ASCII code
- toString(10) -> convert to 10-base String
- toString(2) – convert to binary String.
- These functions are important when we deal with the keyboard events.



```html
<html> <head><title>ASCII code</title></head>
<body><script>
    var Letter ="A"; var Code = "";
    Letter = "A"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "B"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "C"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "D"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "E"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "F"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "G"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "H"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "I"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "J"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "K"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "L"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "M"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "N"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "O"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "P"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "Q"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "R"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "S"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "T"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "U"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "V"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "W"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "X"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "Y"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
    Letter = "Z"; Code = Letter.charCodeAt(0).toString(10); document.write(Code+"<br>");
</script></body></html>
```



```
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
```

Keyboard and Mouse Events

- Mouse Events
  - onMouseDown
  - onMouseMove
  - onMouseOut
  - onMouseOver
  - onMouseUp
  - onClick
  - onDblClick
  - onDragDrop

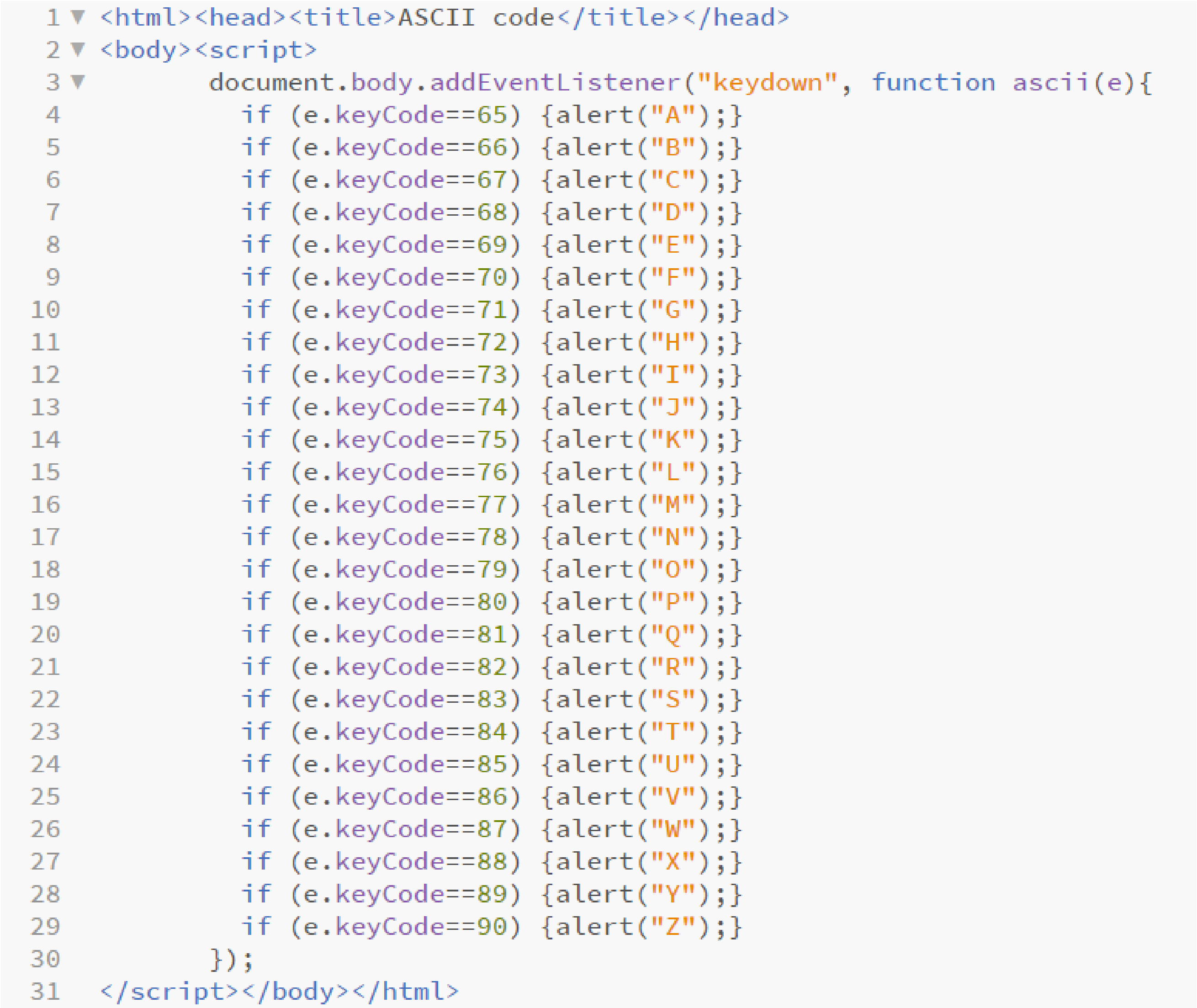
- Keyboard Events
  - onKeyDown
  - onKeyPress
  - onKeyUp

Project 13: Keyboard Event

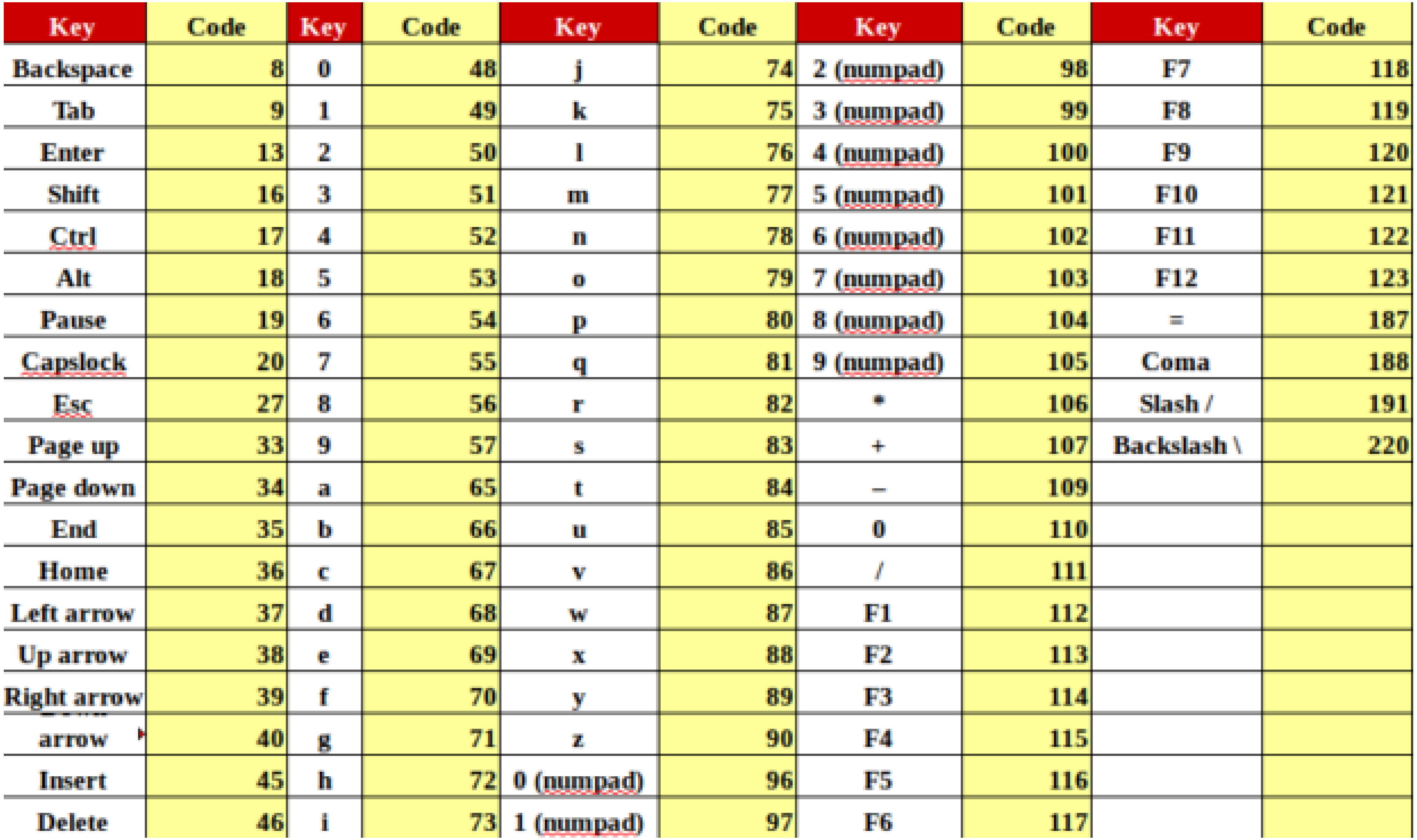
Demo Program: keyboardevent.html

- Using **addEventListener** to create dynamic binding between the event and handler.
- <body onclick="handler()"></body> This type of binding is static binding.
- document.body.addEventListener("onclick",

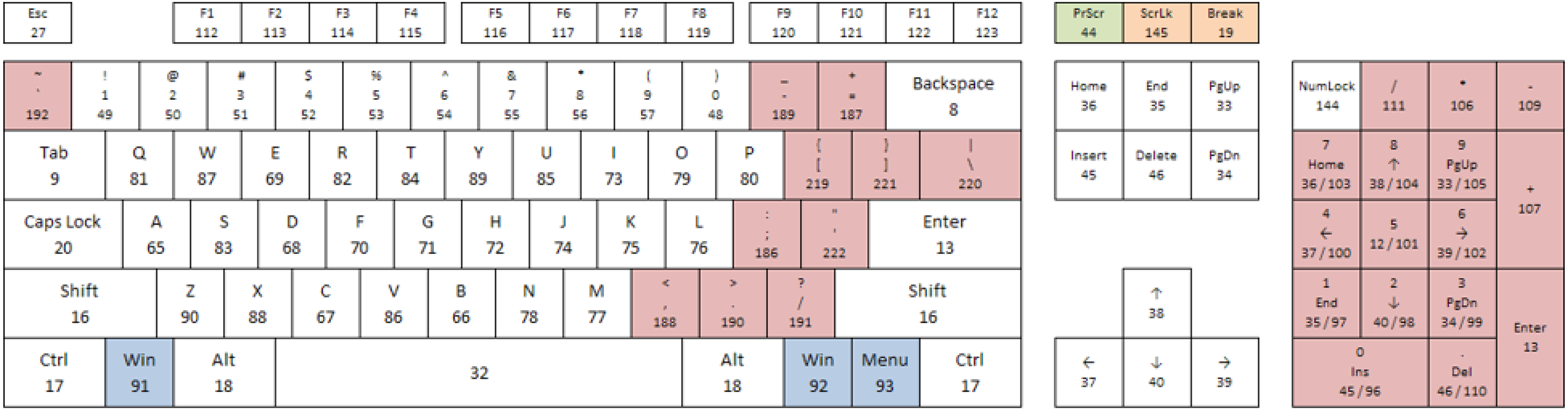
```
        function handler(event_object){ … }
 );  // this is dynamic binding
```



```
<html><head><title>ASCII code</title></head>
<body><script>
       document.body.addEventListener("keydown", function ascii(e){
         if (e.keyCode==65) {alert("A");}
         if (e.keyCode==66) {alert("B");}
         if (e.keyCode==67) {alert("C");}
         if (e.keyCode==68) {alert("D");}
         if (e.keyCode==69) {alert("E");}
         if (e.keyCode==70) {alert("F");}
         if (e.keyCode==71) {alert("G");}
         if (e.keyCode==72) {alert("H");}
         if (e.keyCode==73) {alert("I");}
         if (e.keyCode==74) {alert("J");}
         if (e.keyCode==75) {alert("K");}
         if (e.keyCode==76) {alert("L");}
         if (e.keyCode==77) {alert("M");}
         if (e.keyCode==78) {alert("N");}
         if (e.keyCode==79) {alert("O");}
         if (e.keyCode==80) {alert("P");}
         if (e.keyCode==81) {alert("Q");}
         if (e.keyCode==82) {alert("R");}
         if (e.keyCode==83) {alert("S");}
         if (e.keyCode==84) {alert("T");}
         if (e.keyCode==85) {alert("U");}
         if (e.keyCode==86) {alert("V");}
         if (e.keyCode==87) {alert("W");}
         if (e.keyCode==88) {alert("X");}
         if (e.keyCode==89) {alert("Y");}
         if (e.keyCode==90) {alert("Z");}
       });
</script></body></html>
```

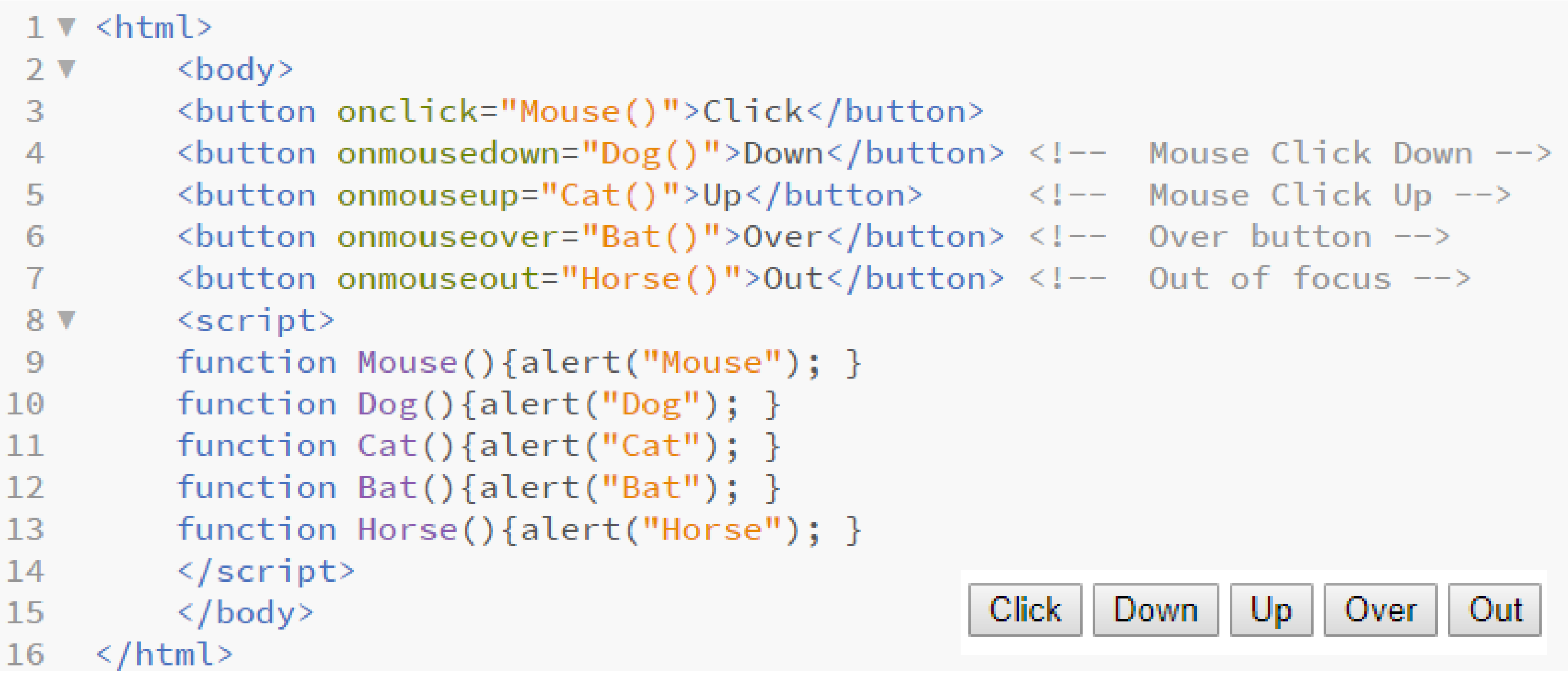


Key	Code	Key	Code	Key	Code	Key	Code	Key	Code
Backspace	8	0	48	j	74	2 (numpad)	98	F7	118
Tab	9	1	49	k	75	3 (numpad)	99	F8	119
Enter	13	2	50	l	76	4 (numpad)	100	F9	120
Shift	16	3	51	m	77	5 (numpad)	101	F10	121
Ctrl	17	4	52	n	78	6 (numpad)	102	F11	122
Alt	18	5	53	o	79	7 (numpad)	103	F12	123
Pause	19	6	54	p	80	8 (numpad)	104	=	187
Capslock	20	7	55	q	81	9 (numpad)	105	Coma	188
Esc	27	8	56	r	82	*	106	Slash /	191
Page up	33	9	57	s	83	+	107	Backslash \	220
Page down	34	a	65	t	84	–	109		
End	35	b	66	u	85	0	110		
Home	36	c	67	v	86	/	111		
Left arrow	37	d	68	w	87	F1	112		
Up arrow	38	e	69	x	88	F2	113		
Right arrow	39	f	70	y	89	F3	114		
Down arrow	40	g	71	z	90	F4	115		
Insert	45	h	72	0 (numpad)	96	F5	116		
Delete	46	i	73	1 (numpad)	97	F6	117		

Esc 27
F1 112
F2 113
F3 114
F4 115
F5 116
F6 117
F7 118
F8 119
F9 120
F10 121
F11 122
F12 123
PrScr 44
ScrLk 145
Break 19
~ ` 192
! 1 49
@ 2 50
# 3 51
$ 4 52
% 5 53
^ 6 54
& 7 55
* 8 56
(9 57
) 0 48
_ - 189
+ = 187
Backspace 8
Home 36
End 35
PgUp 33
NumLock 144
/ 111
* 106
- 109
Tab 9
Q 81
W 87
E 69
R 82
T 84
Y 89
U 85
I 73
O 79
P 80
{ [219
}] 221
| \ 220
Insert 45
Delete 46
PgDn 34
7 Home 36 / 103
8 ↑ 38 / 104
9 PgUp 33 / 105
+ 107
Caps Lock 20
A 65
S 83
D 68
F 70
G 71
H 72
J 74
K 75
L 76
: ; 186
" ' 222
Enter 13
4 ← 37 / 100
5 12 / 101
6 → 39 / 102
Shift 16
Z 90
X 88
C 67
V 86
B 66
N 78
M 77
< , 188
> . 190
? / 191
Shift 16
↑ 38
1 End 35 / 97
2 ↓ 40 / 98
3 PgDn 34 / 99
Enter 13
Ctrl 17
Win 91
Alt 18
32
Alt 18
Win 92
Menu 93
Ctrl 17
← 37
↓ 40
→ 39
0 Ins 45 / 96
. Del 46 / 110

Project 14: Mouse Event

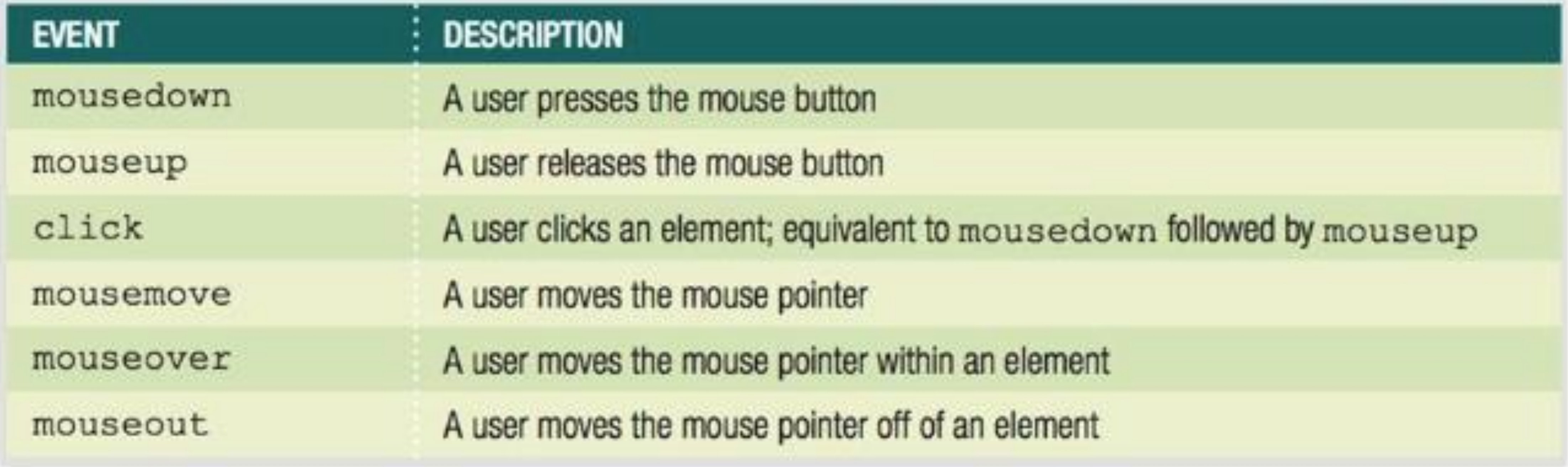
Demo Program: mouse.html



```
<html>
    <body>
    <button onclick="Mouse()">Click</button>
    <button onmousedown="Dog()">Down</button> <!--  Mouse Click Down -->
    <button onmouseup="Cat()">Up</button>     <!--  Mouse Click Up -->
    <button onmouseover="Bat()">Over</button> <!--  Over button -->
    <button onmouseout="Horse()">Out</button> <!--  Out of focus -->
    <script>
    function Mouse(){alert("Mouse"); }
    function Dog(){alert("Dog"); }
    function Cat(){alert("Cat"); }
    function Bat(){alert("Bat"); }
    function Horse(){alert("Horse"); }
    </script>
    </body>
</html>
```

Creating a Drag-and Drop Application with Mouse Events

- Mouse events
  - events based on actions of mouse or touchpad



EVENT	DESCRIPTION
`mousedown`	A user presses the mouse button
`mouseup`	A user releases the mouse button
`click`	A user clicks an element; equivalent to `mousedown` followed by `mouseup`
`mousemove`	A user moves the mouse pointer
`mouseover`	A user moves the mouse pointer within an element
`mouseout`	A user moves the mouse pointer off of an element

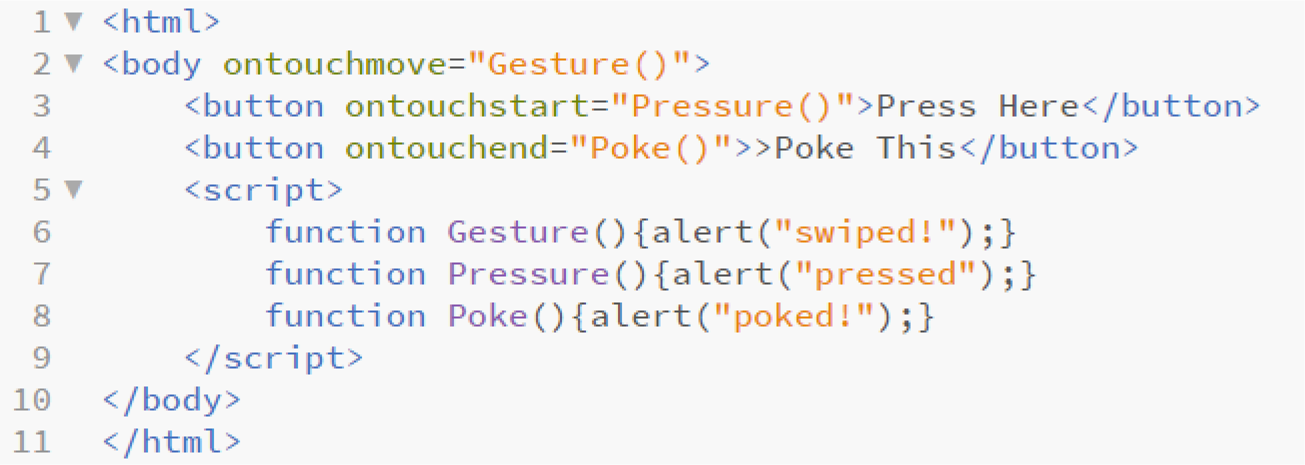
Project 15: Touch Event

Demo Program: touch.html

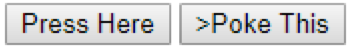
ontouchmove: Swipe

ontouchstart: Press

ontouchend: Poke

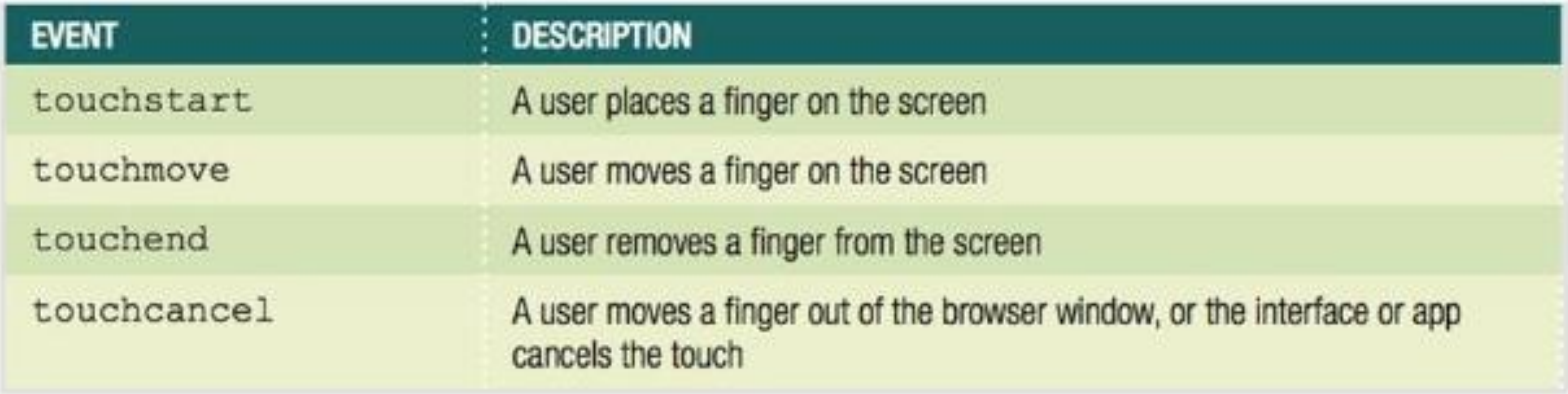


```
<html>
<body ontouchmove="Gesture()">
    <button ontouchstart="Pressure()">Press Here</button>
    <button ontouchend="Poke()">>Poke This</button>
    <script>
        function Gesture(){alert("swiped!");}
        function Pressure(){alert("pressed");}
        function Poke(){alert("poked!");}
    </script>
</body>
</html>
```

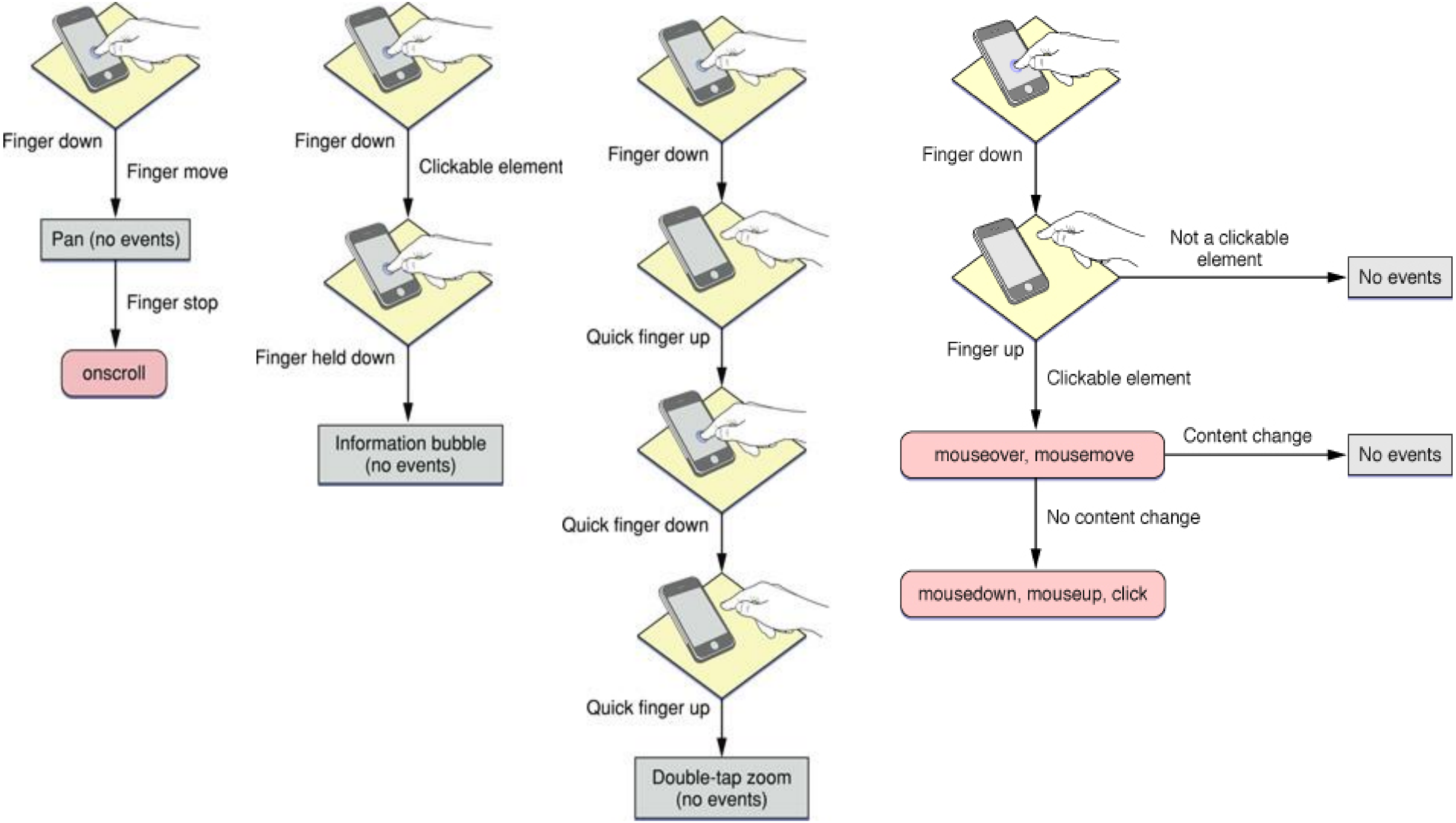


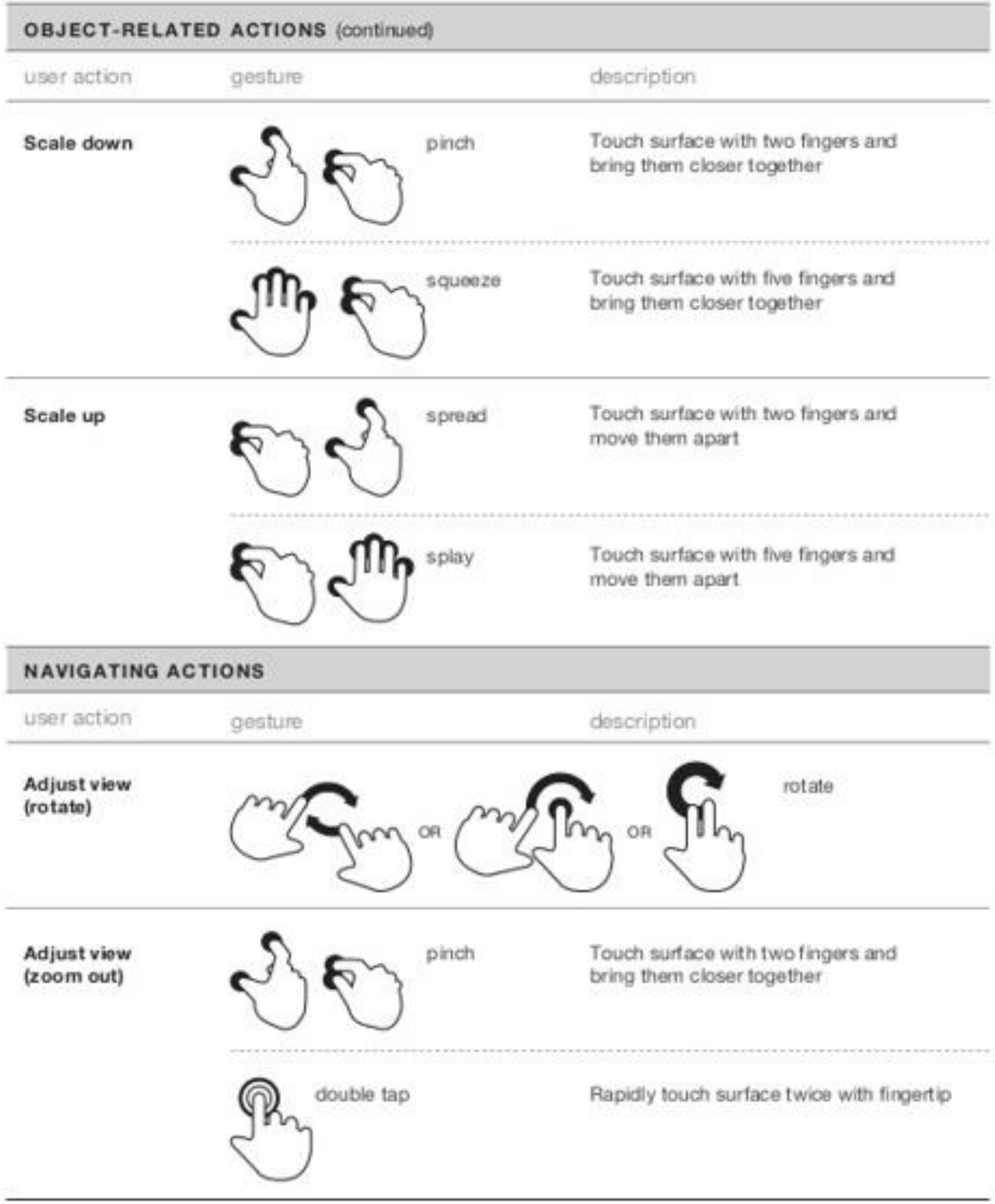
Implementing Touch Events

- Respond to user's finger touches on a touchscreen



EVENT	DESCRIPTION
`touchstart`	A user places a finger on the screen
`touchmove`	A user moves a finger on the screen
`touchend`	A user removes a finger from the screen
`touchcancel`	A user moves a finger out of the browser window, or the interface or app cancels the touch


Finger down
Finger move
Pan (no events)
Finger stop
onscroll
Finger down
Clickable element
Finger held down
Information bubble (no events)
Finger down
Quick finger up
Quick finger down
Quick finger up
Double-tap zoom (no events)
Finger down
Not a clickable element
No events
Finger up
Clickable element
mouseover, mousemove
Content change
No events
No content change
mousedown, mouseup, click

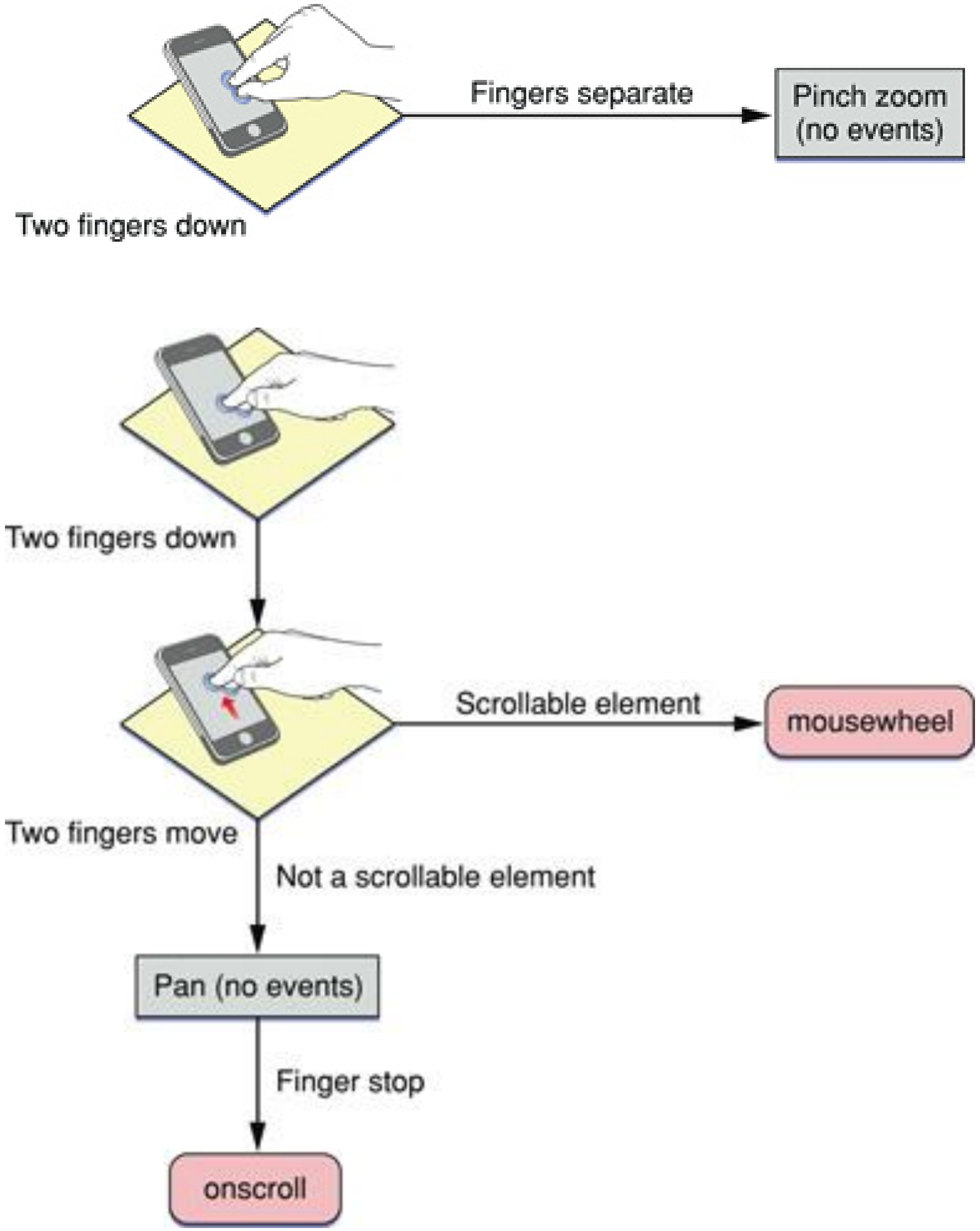


OBJECT-RELATED ACTIONS (continued)

user action	gesture	description
Scale down	pinch	Touch surface with two fingers and bring them closer together
	squeeze	Touch surface with five fingers and bring them closer together
Scale up	spread	Touch surface with two fingers and move them apart
	splay	Touch surface with five fingers and move them apart

NAVIGATING ACTIONS

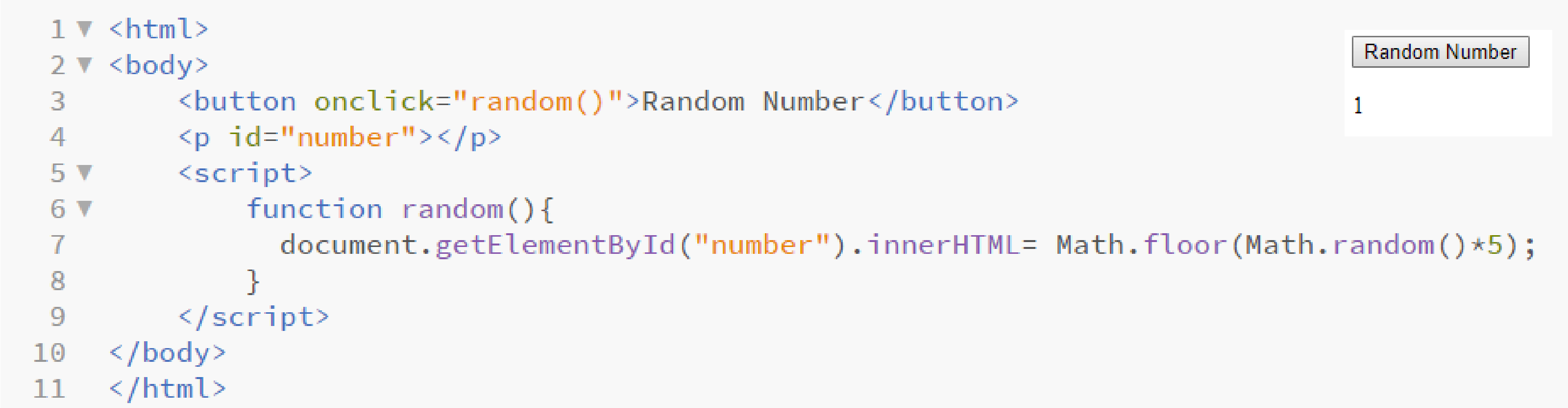
user action	gesture	description
Adjust view (rotate)	OR OR rotate	
Adjust view (zoom out)	pinch	Touch surface with two fingers and bring them closer together
	double tap	Rapidly touch surface twice with fingertip



Project 16: Random Number Generator

Demo Program: random.html

- Click to generate a new random number.

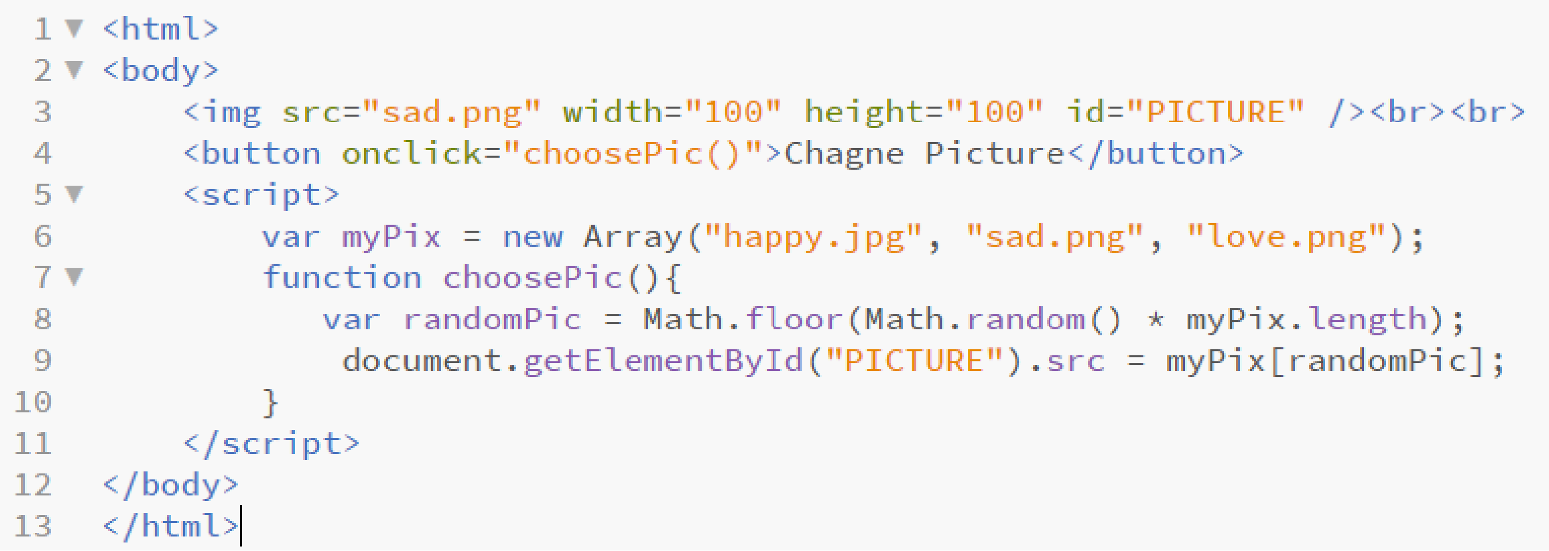


```
<html>
<body>
    <button onclick="random()">Random Number</button>
    <p id="number"></p>
    <script>
        function random(){
          document.getElementById("number").innerHTML= Math.floor(Math.random()*5);
        }
    </script>
</body>
</html>
```

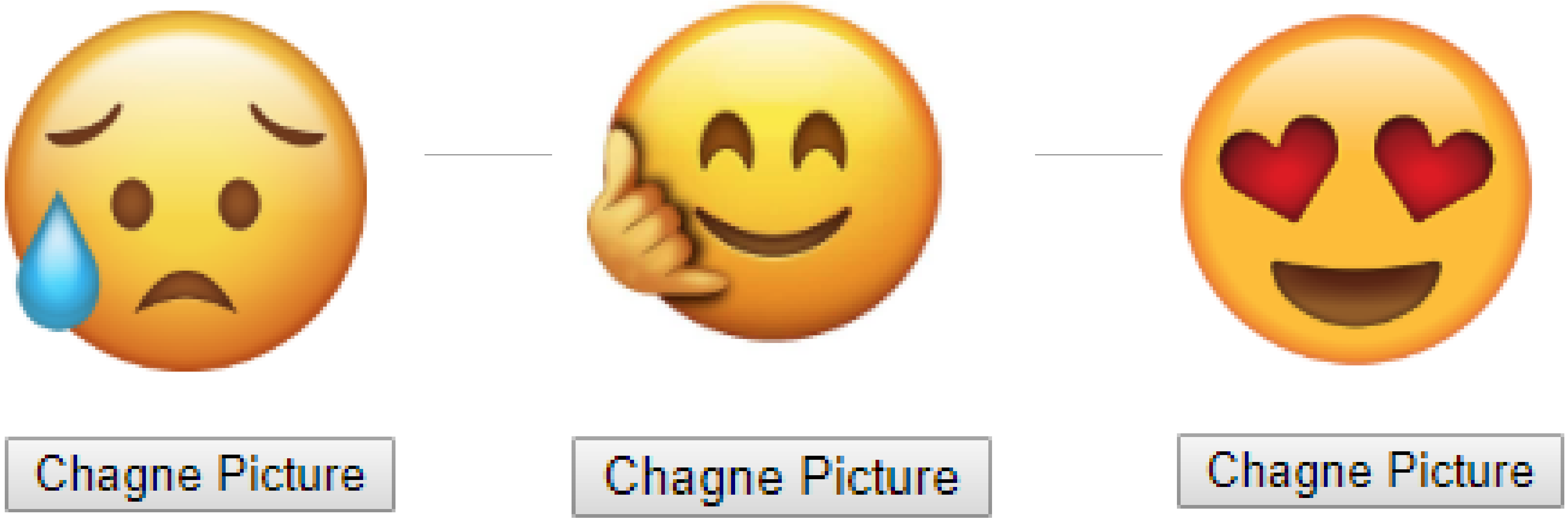
```
1
```

Project 16: Random Number Generator

Demo Program: randomimage.html



```
<html>
<body>
    <img src="sad.png" width="100" height="100" id="PICTURE" /><br><br>
    <button onclick="choosePic()">Chagne Picture</button>
    <script>
        var myPix = new Array("happy.jpg", "sad.png", "love.png");
        function choosePic(){
           var randomPic = Math.floor(Math.random() * myPix.length);
            document.getElementById("PICTURE").src = myPix[randomPic];
        }
    </script>
</body>
</html>
```



Project 16: Random Number Generator

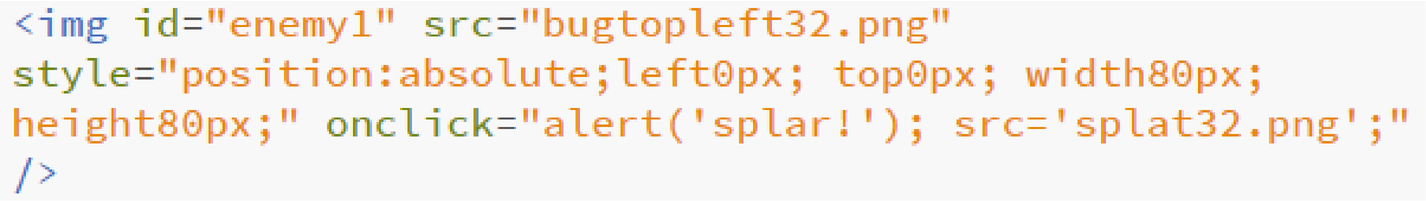
Demo Program: randomimage.html

CLICK TO GENERATE A NEW RANDOM NUMBER AND THEN PICK A NEW IMAGE.

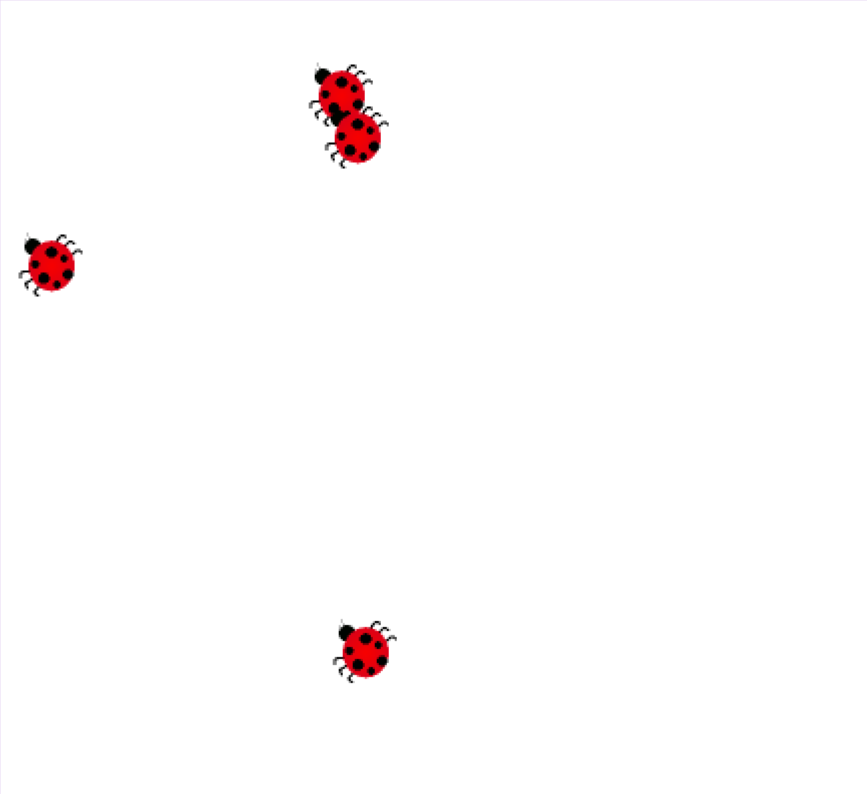
Project 17: Bug and Splat

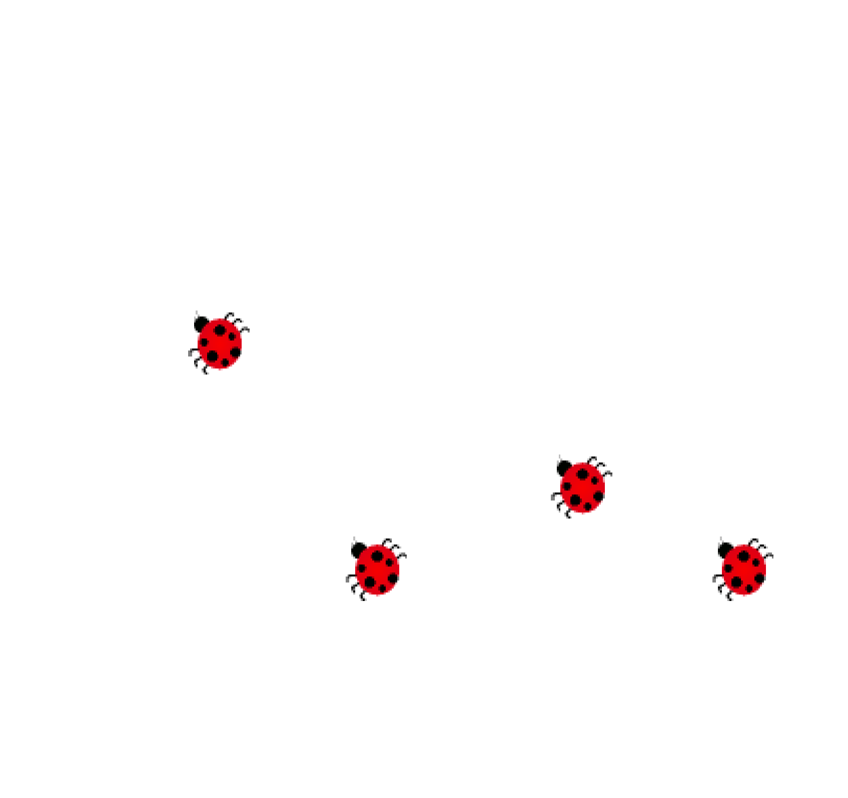
Demo Program: bugsplat.html

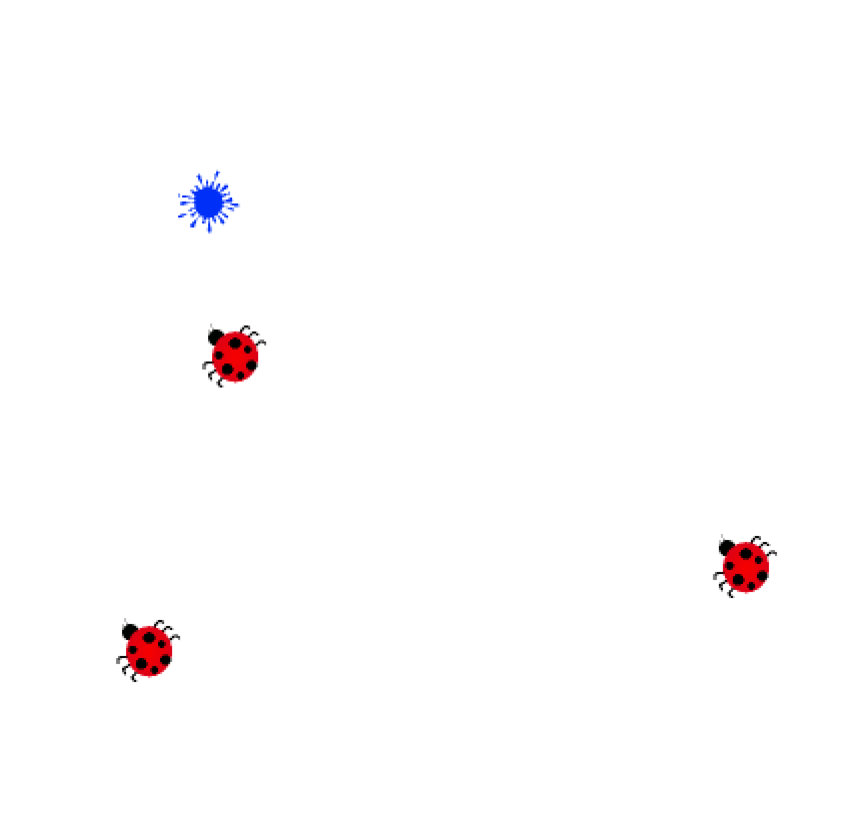
- Put 4 image items on to the document. Then randomly generate new top-level location for each image.
- The play can use mouse to click on the bug. When a bug is clicked upon, the image will change to “Splat”.



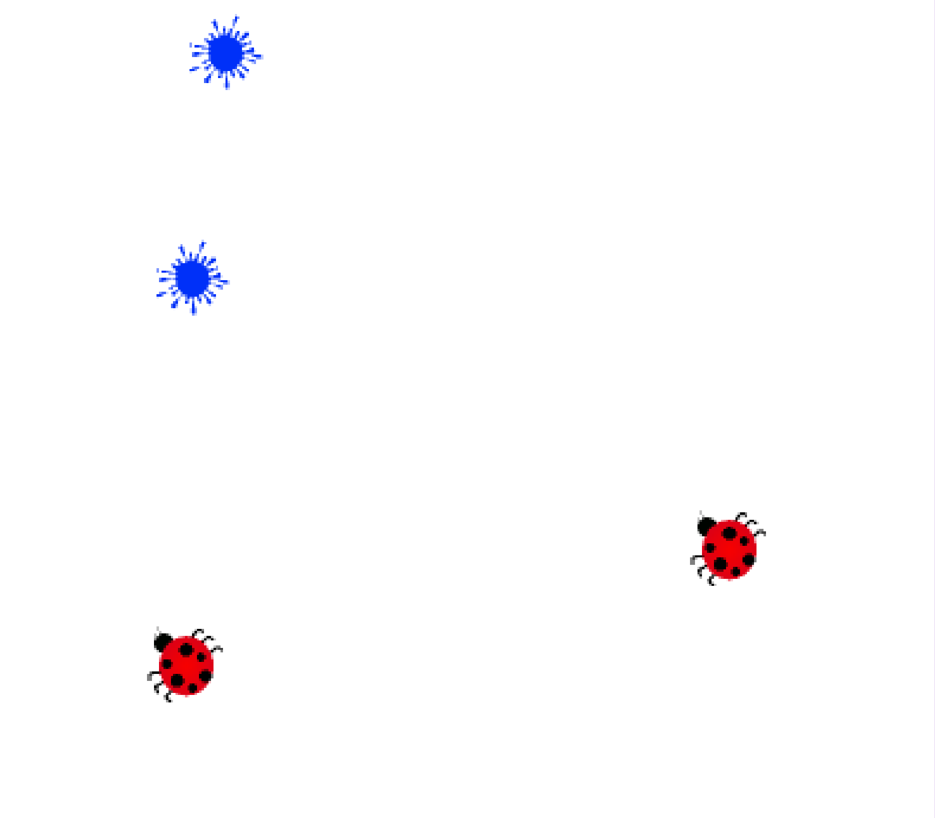
```
<img id="enemy1" src="bugtopleft32.png"
style="position:absolute;left0px; top0px; width80px;
height80px;" onclick="alert('splar!'); src='splat32.png';"
/>
```







One Hit



Two Hit