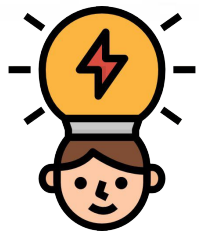


# Web Development with **Javascript**

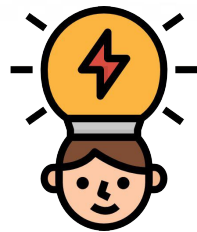
## Module 2

JS Data Types, Methods, Functions, and Events





# Objectives

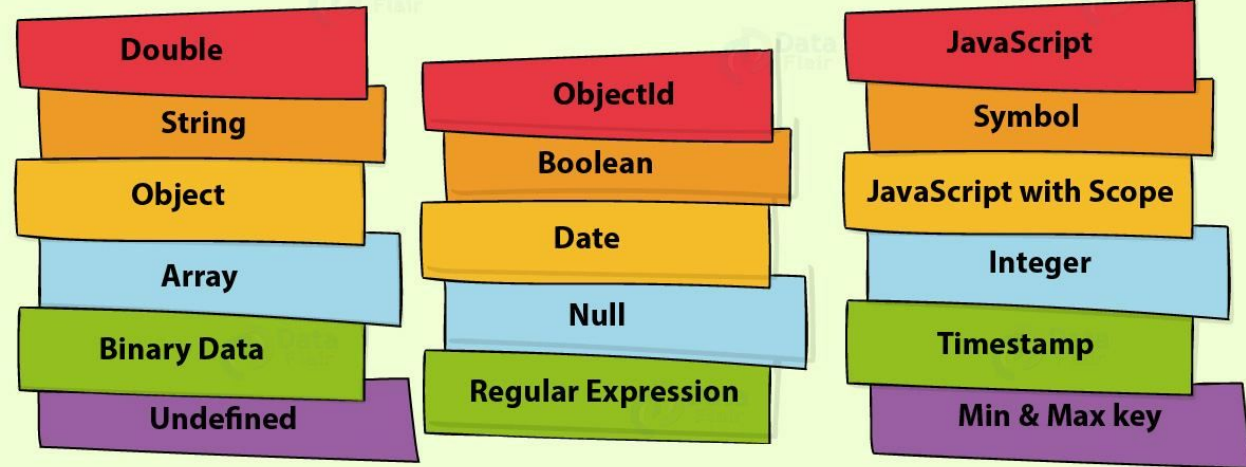


# What we'll do today

- » Understand the different ways to manipulate numbers, strings and arrays in JavaScript
- » Explore the various built-in Methods for numbers, strings, and arrays
- » To learn what functions are and its syntax in JavaScript
- » To learn how to use these HTML Events and why is it useful
- » To learn how functions and HTML Events are related to each other



# Data Types



# JS

## Data Types

# Number

- » The Number constructor contains constants and methods for working with numbers.
- » Values of other types can be converted to numbers using the **Number()** function
- » parseInt()

# Strings

- » Strings are used when storing a sequence of letters, numbers not used for calculation, special characters, and so on.
- » Strings can be used to display and check content like username and passwords.

# Arrays

- » An array is used to store multiple values to a single variable.
- » This is how arrays are used:

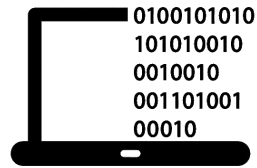
```
var softDrinks = ["Coke", "Pepsi", "Sprite"];
```



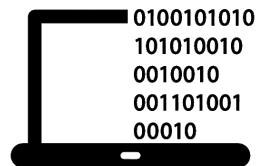


test.html x

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title></title>
8 </head>
9 <body>
10  <ul class="nav">
11    <li class="item1">
12      <div class="title">Menu 1</div>
13    </li>
14    <li class="item2">
15      <div class="title">Menu 2</div>
16    </li>
17    <li class="item3">
18      <div class="title">Menu 3</div>
19    </li>
20  </ul>
21 </body>
22 </html>
```



# Let's start coding!



# JavaScript

Data Types



## Data Type Methods

# Number Methods: Return Value

- » `let x = 2`
- » Returning the value of a Number
  - `x.valueOf()`

# Number Methods: Conversion

## » String to Integer operations

- **var** num= “3”;

num= Number(num);            // String to Number(Int)

// or

num = parseInt(num,<base>); // to any number base

## » String to Float operation

- **var** float = “3.125”;

float = parseFloat(name);    // Transforms String to Float

# String Methods : Conversion

» Transform Data Types into another different Data Type

» Number to String

- **var** str = 15;

```
str= str.toString();      // to base 10 string = "15"
```

// or

```
str = str.toString(2);    // to base 2    string = "1111"
```

```
str = str.toString(8);    // to base 8    string = "17"
```

```
str = str.toString(16);   // to base 16 string = "f"
```

// or

```
var new_string += str    // to base 10 string = "15"
```

# String Methods: Other

## » Finding string length

- **let** firstName = 'Mark'  
    firstName.**length**

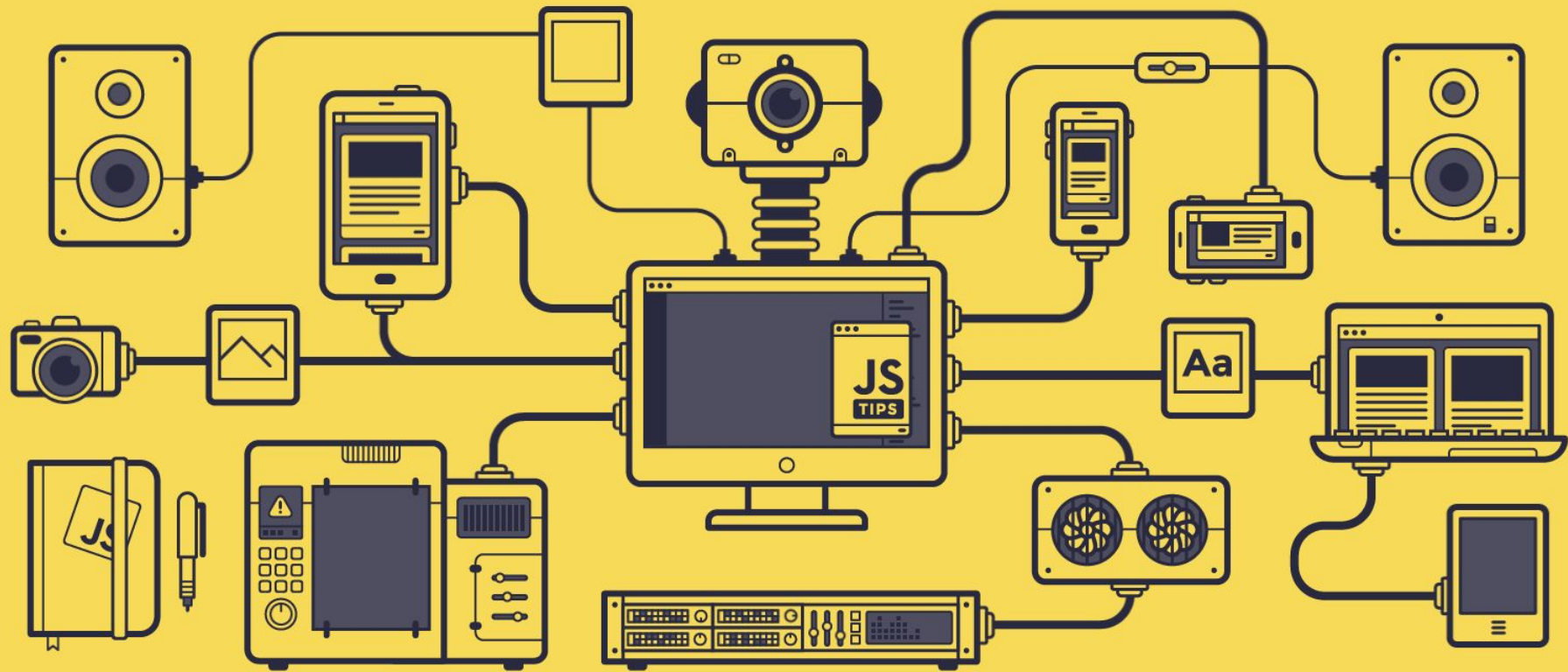
## » Finding a specific character in a string

- **firstName[0]**

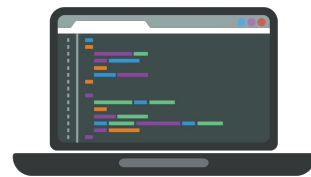
This returns the first letter of the word stored inside the firstName variable.

## » Changing cases

- **let** lastName = 'ZuCkErBerg';  
    **var** new = lastName.**toLowerCase()**;  
    lastName.**toUpperCase()**;



Try on your own



# Activity 1

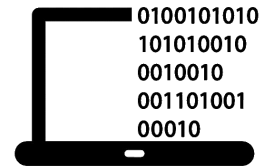
1. Go to your webdev folder.
2. Create files named webjs\_21.html and webjs\_21.js.
3. Open the files in Visual Studio Code.
4. Create a script that converts a sentence from lower case to upper case and vice versa.
5. Create a script that accepts two word inputs, it should detect if the entered words are equal regardless of case sensitivity.



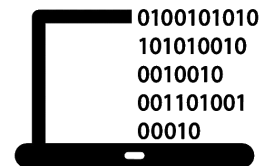


test.html x

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title></title>
8 </head>
9 <body>
10  <ul class="nav">
11    <li class="item1">
12      <div class="title">Menu 1</div>
13    </li>
14    <li class="item2">
15      <div class="title">Menu 2</div>
16    </li>
17    <li class="item3">
18      <div class="title">Menu 3</div>
19    </li>
20  </ul>
21 </body>
22 </html>
```



# Let's start coding!



# Array: Indexing

» Given:

```
var softDrinks = ["Coke", "Pepsi", "Sprite"];
```

**softDrinks[0]** returns **"Coke"**

// This will return the very first data in the array, which is Coke.

# Array: Methods (Pop & Push)

## » Popping:

- Removes the last element in the array

```
var favorites = ["Dog", "Cat", "Cake"];  
favorites.pop();
```

## » Pushing:

- Adds an item on the last index of the array

```
var favorites = ["Dog", "Cat", "Cake"];  
favorites.push("Ice Cream")
```

# Array: Methods (Replace)

- » This adds the element “Games” to the second of the array replacing “Cat”

```
var favorites = ["Dog", "Cat", "Cake"];  
favorites[1] = "Games";
```

# Array: Methods (Slice & Sort)

- » This makes a new array with the elements from favorites starting from “Cat”

```
var favorites = ["Dog", "Cat", "Cake"];  
var most_favorite = favorites.slice(1);
```

- » This method sorts the array in alphabetical order

```
var favorites = ["Dog", "Cat", "Cake"];  
favorites.sort();
```

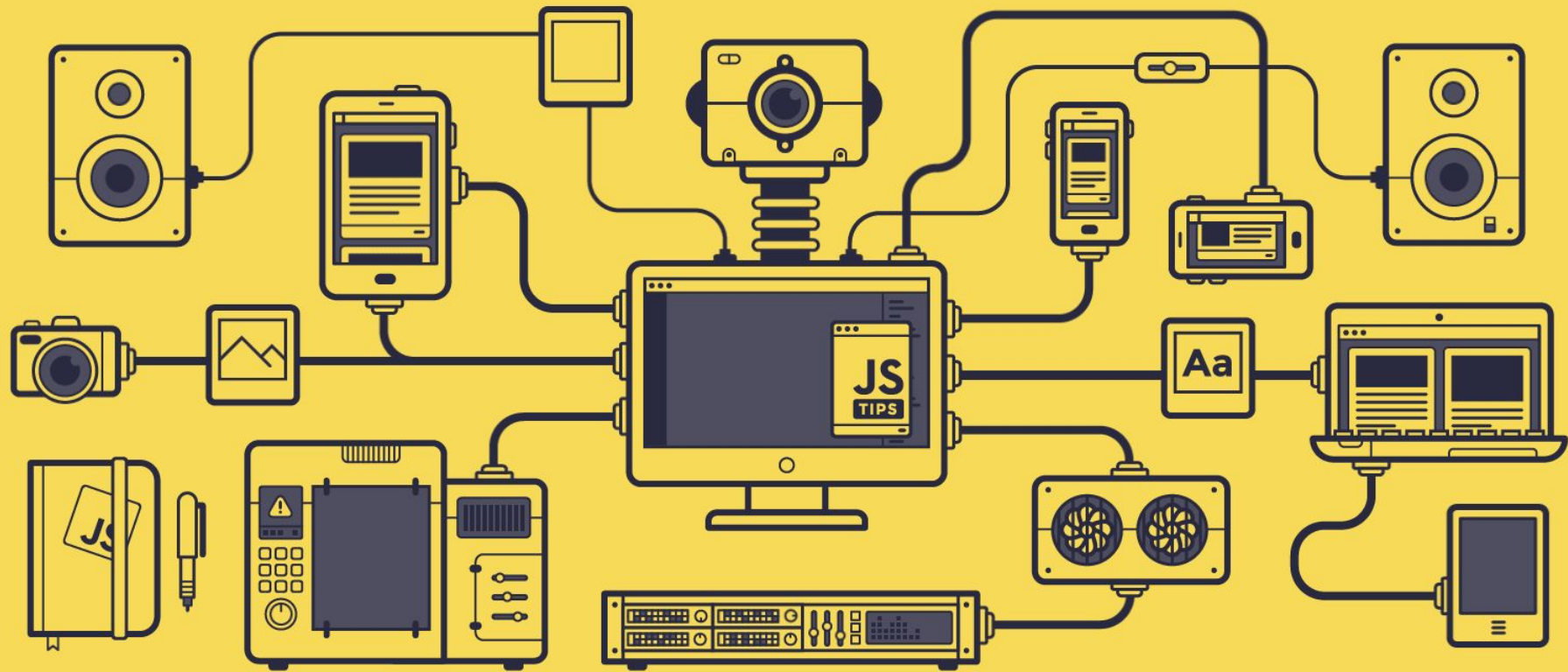
# Array: Iteration

## » For Loops:

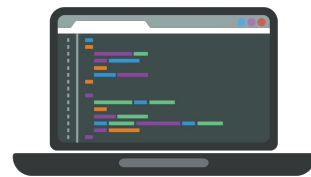
```
var Loona_members = ["Heejin", "Hyunjin", "Haseul", "Yeojin", "Vivi", "Kim Lip", "Jinsoul", "Choerry",  
"Yves", "Chuu", "Go Won", "Olivia Hye"];  
  
for(var i = 0; i < Loona_members.length; i++){  
    console.log(Loona_members[i]);  
}
```

## » While Loops:

```
i = 0;  
  
while(i < Loona_members.length){  
    console.log(Loona_members[i]);  
    i++;  
}
```



Try on your own



# Daily Output 1

1. Go to your webdev folder.
2. Create files named favorites.html and favorites.js.
3. Open the files in Visual Studio Code.
4. Create a Favorites website with the following features:
  - Declare an array named “favorites” with at least 3 elements of your favorite things
  - It can change the specified item to a new value
  - It can add or remove a specified item in the list
  - Inputs will come from alert boxes





# Concepts to remember





# JavaScript

# JS

## Functions, Events, Event Listeners

# Functions

- » A block of code that executes a specific task
- » Only runs whenever it is called

```
function myFunction(p1, p2) {  
    return p1 * p2;  
    // The function returns the product of p1 and p2  
}
```

# Functions

## » Syntax

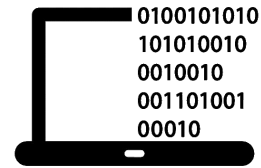
- Name
- Parameters - values needed by the function
- block of code to be executed.
- return value

```
function functionName(parameter1, parameter2, ....) {  
    // block of code  
}
```

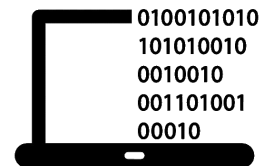


test.html x

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title></title>
8 </head>
9 <body>
10  <ul class="nav">
11    <li class="item1">
12      <div class="title">Menu 1</div>
13    </li>
14    <li class="item2">
15      <div class="title">Menu 2</div>
16    </li>
17    <li class="item3">
18      <div class="title">Menu 3</div>
19    </li>
20  </ul>
21 </body>
22 </html>
```



# Let's start coding!



NAME



PARAMETERS



BODY



```
function addNumbers(a, b) {  
  return a + b;  
}
```

# JS

## Functions

# Functions

## » Example

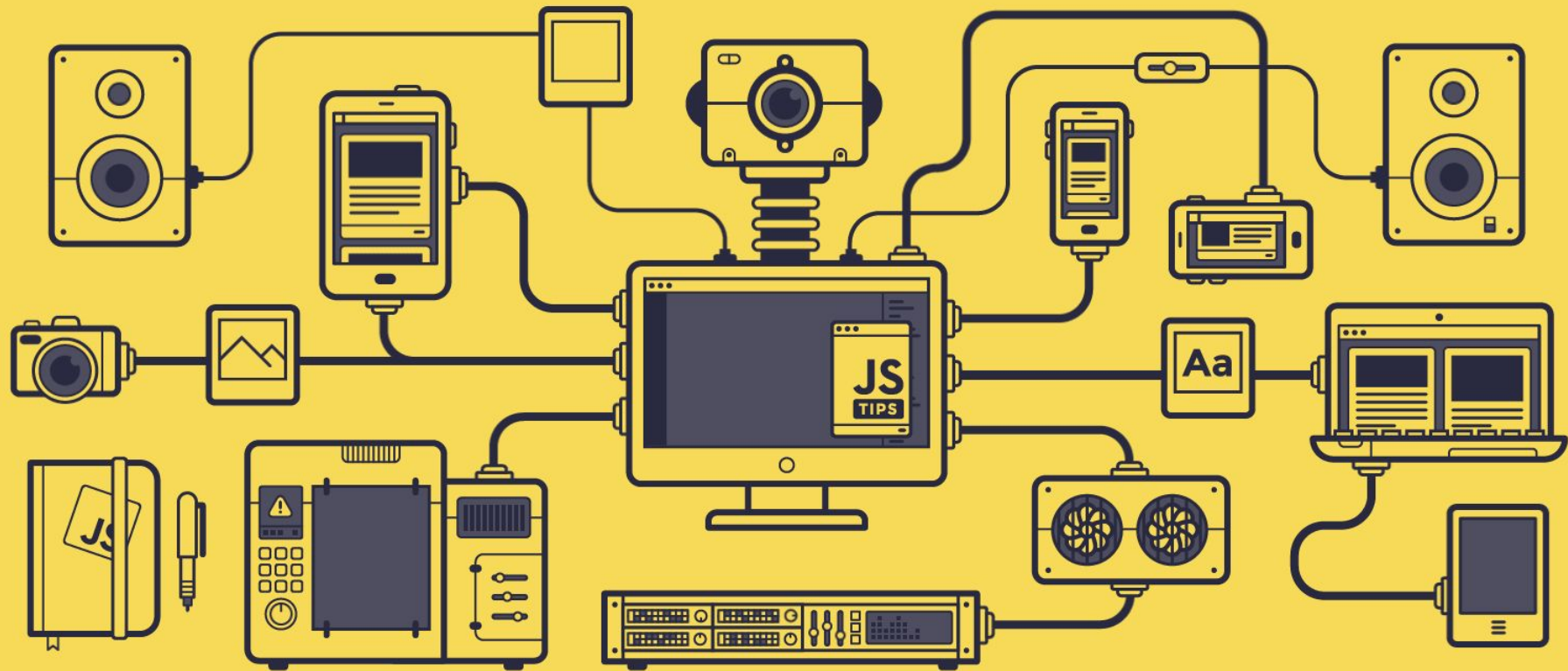
```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}  
  
document.getElementById("demo").innerHTML  
= toCelsius(77);
```

# Functions

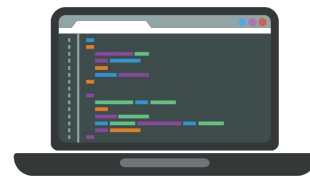
## » Example

```
function addAllNumbers(start, last) {  
    var i, sum;  
    for(i = start, sum = 0; i <= last; i++){  
        sum += i;  
    }  
    return sum;  
}  
  
var summation = addAllNumbers(1,100);
```





Try on your own



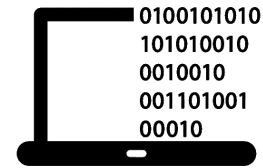
# Activity 2

1. Go to your webdev folder.
2. Create files named webjs\_22.html.
3. Open the files in Visual Studio Code.
4. Create a script with the following JavaScript functions:
  - reverses a given number
  - accepts a string as a parameter and find the longest word within the string

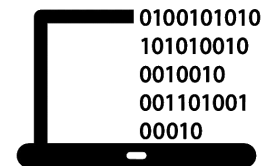


test.html x

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title></title>
8 </head>
9 <body>
10  <ul class="nav">
11    <li class="item1">
12      <div class="title">Menu 1</div>
13    </li>
14    <li class="item2">
15      <div class="title">Menu 2</div>
16    </li>
17    <li class="item3">
18      <div class="title">Menu 3</div>
19    </li>
20  </ul>
21 </body>
22 </html>
```



# Let's start coding!



# A Simple Explanation of Event Delegation



# JS

## Events

# HTML Events

- » two types:
  - a browser does (e.g. web page has finished loading)
  - a user does (e.g. clicking buttons)
- » Uses JavaScript code to work

# HTML Events

» Syntax

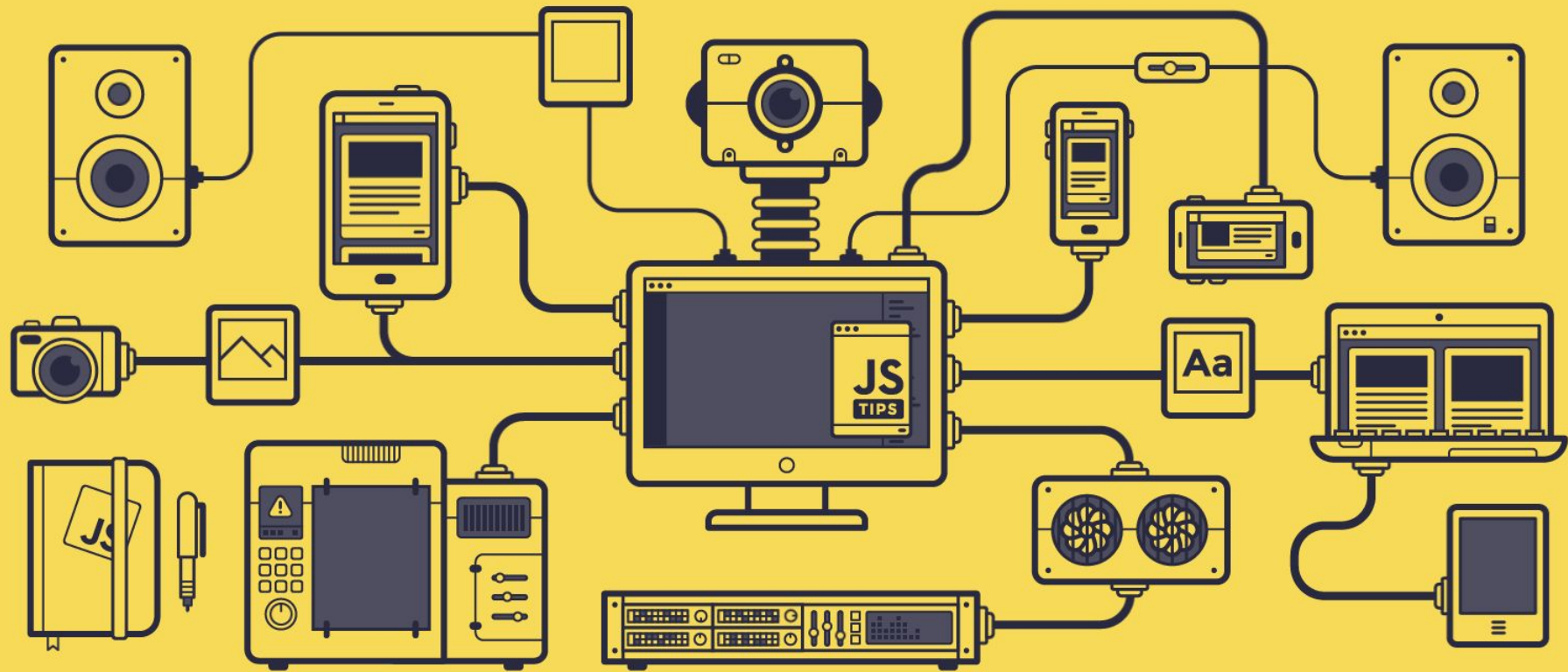
```
<element event="some JavaScript">
```

E.g.

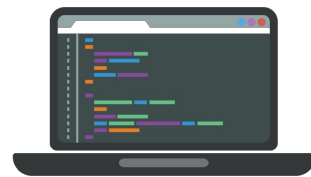
```
<button  
onclick="document.getElementById('demo').innerHTMLHT  
ML = Date()">The time is?</button>
```

# HTML Events: Common

Event	Event Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page



Try on your own





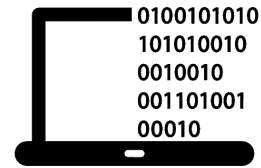
# Activity 3

1. Go to your webdev folder.
2. Create file named webjs\_23.html.
3. Open the files in Visual Studio Code.
4. Create a website with the following characteristics:
  - Whenever you hover to the red box at the center of the page, it will say “Mouse Over”, else “Mouse Out”, and an image will also appear.
  - When a button is click, a big text will appear that says “SURPRISE!”

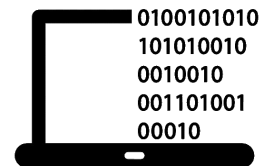


test.html x

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title></title>
8 </head>
9 <body>
10  <ul class="nav">
11    <li class="item1">
12      <div class="title">Menu 1</div>
13    </li>
14    <li class="item2">
15      <div class="title">Menu 2</div>
16    </li>
17    <li class="item3">
18      <div class="title">Menu 3</div>
19    </li>
20  </ul>
21 </body>
22 </html>
```



# Let's start coding!



# HTML DOM EventListener

## » addEventListener()

- attaches an event handler to the specified element
- can add multiple event handlers to a single element
- can add multiple event handlers of the same type to a single element

# HTML DOM EventListener

## » addEventListener()

### ○ Syntax

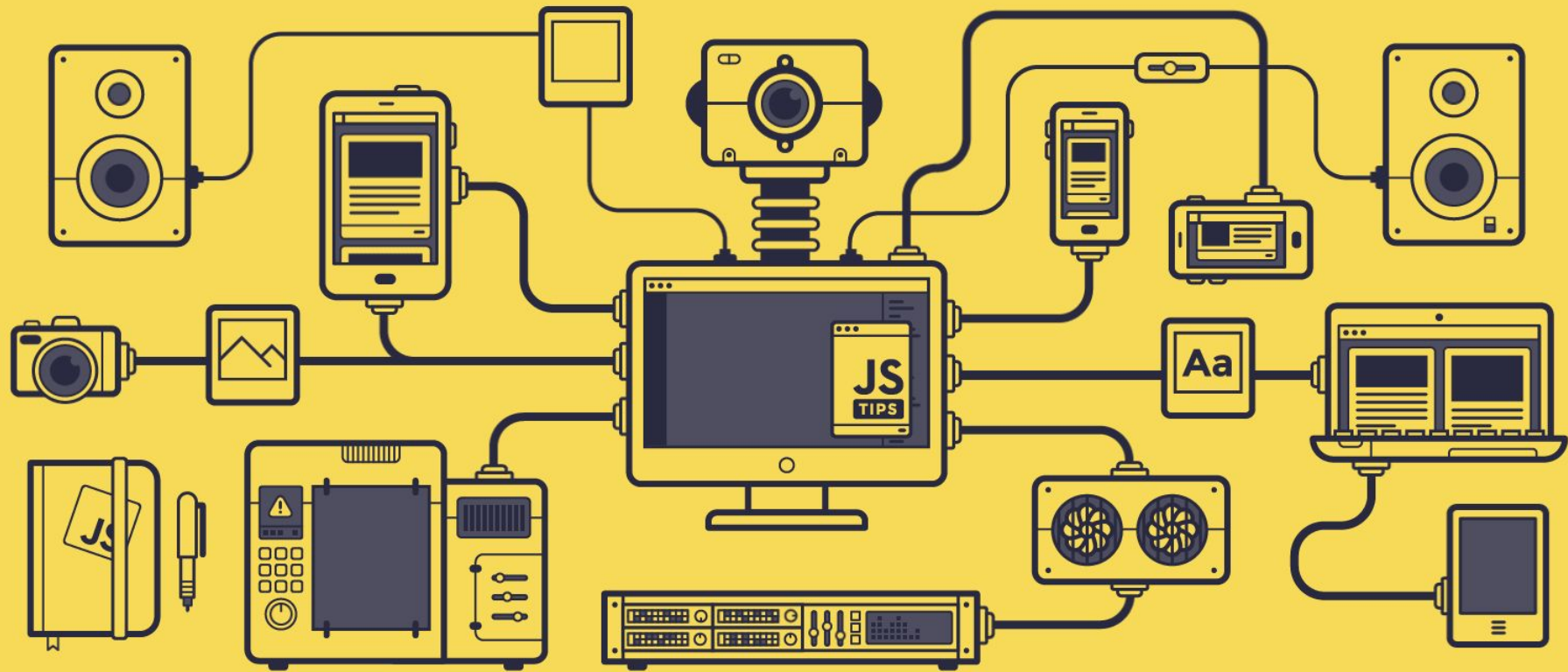
*`element.addEventListener(event, function, useCapture);`*

E.g.

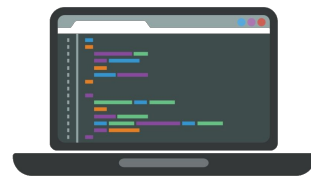
```
document.getElementById("myBtn").addEventListener("click", displayDate);
```

# Common HTML DOM Events

- » **mouse events:** click, dblclick, mousemove, mouseover, mousewheel, mouseout, mousedown, mouseup, contextmenu
- » **touch events:** touchend, touchstart, touchmove, touchcancel
- » **keyboard events:** keydown, keypress, keyup
- » **form events:** focus, blur, change, submit
- » **window events:** scroll, resize, hashchange, load, unload



Try on your own



# Activity 4

1. Go to your webdev folder.
2. Go back to webjs\_23.html.
3. Open the files in Visual Studio Code.
4. Edit the ff that you've made:
  - add EventListeners to the red box “Mouse over” and “Mouse out”
  - add EventListeners to the button “Click”

# Daily Output 2

1. Go to your webdev folder.
2. Create files named calculator.html.
3. Open the files in Visual Studio Code.
4. Create a simple calculator website. The calculator can accept two input numbers. The operators it can do are the following:
  - add
  - subtract
  - multiple
  - divide





**You finished today!**

Good job! That was a lot, but you  
managed to finish it! See you again  
next meeting