

---

<Semester II>

# <//Programming for Interactive Interfaces//>

Week 7 Part 1

---

# console.log() and JavaScript Comments

- ❑ The `console.log()` method in JavaScript logs messages or data to the console.
- ❑ The `console.log()` method is useful for debugging or testing purposes.
- ❑ You will not see the output of `console.log()` on your website, but rather in the developer tools.
- ❑ Syntax: `console.log("");`
- ❑ Any message either number, string, array object, etc. can be the parameter
- ❑ Comments help explain code (they are not executed and hence do not have any logic implementation). They can be used to temporarily disable parts of your code.
- ❑ A single-line comment in JavaScript is denoted by two forward slashes (`//`)
- ❑ A multiline comment begins with `/*` and ends with `*/`
- ❑ Using comments to prevent execution of code is suitable for code testing

# HTML DOM Methods

- ❑ The HTML DOM can be accessed with JavaScript (and with other programming languages).
- ❑ In the DOM, all HTML elements are defined as objects.
- ❑ The API is the properties and methods of each object.
- ❑ A property is a value that you can get or set (like changing the content of an HTML element).
- ❑ A method is a type of functions that allows you to take some action on your webpage (like add or deleting an HTML element).
- ❑ For example, you can use DOM methods to access HTML elements using id, attribute, tag name, class name, etc., add events to the document or HTML elements, add new HTML elements to the DOM, etc.

# document.getElementById()

- ❑ The `getElementById()` method returns the elements that have given an ID which is passed to the function. This function is a widely used HTML DOM method in web designing to change the value of any particular element or get a particular element.
- ❑ An Element object describing the DOM element object matching the specified ID, or null if no matching element was found in the document.
- ❑ Each ID needs to be unique. If there are multiple elements with the same ID, only the first one will be returned.
- ❑ Syntax: `document.getElementById( element_ID );`
- ❑ The ID is a case-sensitive string which is unique within the document; only one element should have any given ID

# Forms in HTML: <label> tag

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The HTML <form> element is used to create an HTML form for user input:

The HTML <input> element is the most used form element.

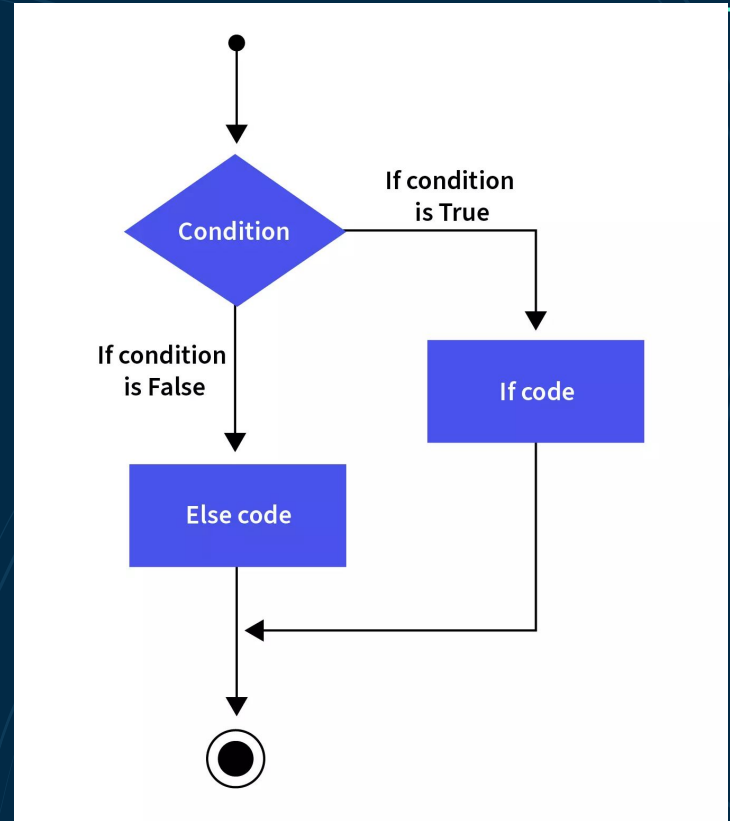
The <label> tag defines a label for many form elements. The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element. The <label> element also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox. The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

# Using document.getElementById() to get User Input



# Conditionals in JavaScript

- ❏ In computer programming, the `if...else` statement is a conditional statement that executes a block of code only when a specific condition is met.
- ❏ The `if...else` statement executes a statement if a specified condition is true. If the condition is false, another statement in the optional `else` clause will be executed.



**JS**

# IF STATEMENTS

**JS**

```
if (coder) {  
    knowledge++  
}
```



# Const Variables



# Arithmetic Operators



# Loops in JavaScript

Loops perform an operation (i.e., a chunk of work) a number of times, usually once for every item in an array or list, or to simply repeat an operation until a certain condition is met. Loops are handy, if you want to run the same code over and over again, each time with a different value.

## The For Loop

The for statement creates a loop with 3 optional expressions:

```
for (expression 1; expression 2; expression 3) {  
    // code block to be executed }
```

Expression 1 is executed (one time) before the execution of the code block.

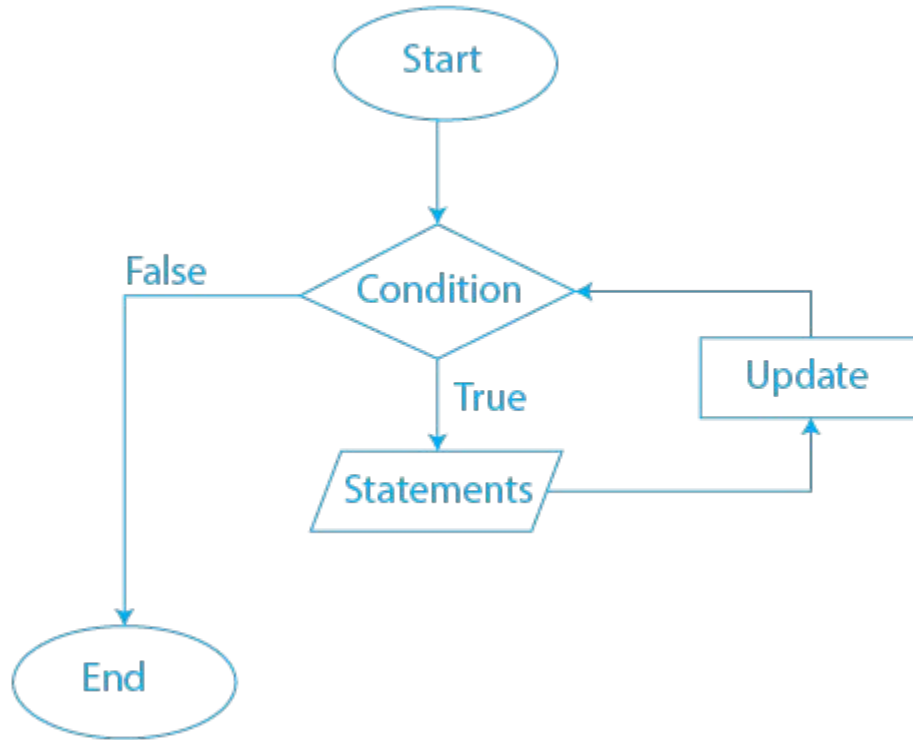
Expression 2 defines the condition for executing the code block.

Expression 3 is executed (every time) after the code block has been executed.

Eg: 

```
for(let i=0; i<10; i++) {  
    console.log(i); }
```

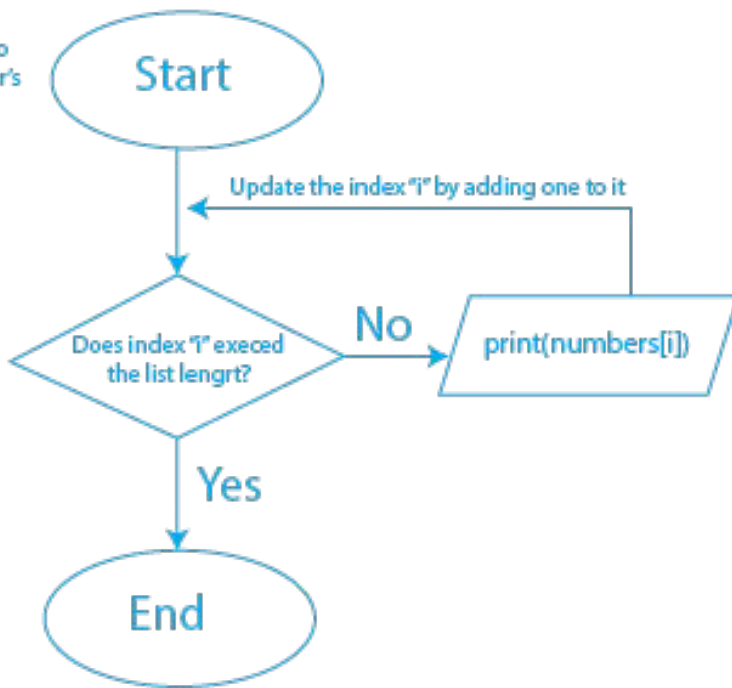
# for Loop



# for Loop

number = [1,2,3,4,5]

index "i" refers to  
the first number's  
index in the list



# for Loop JavaScript

JS

## FOR LOOPS

JS

```
for ( Do something 10 times ) { ... }
```

1 2 3 4 5 6 7 8 9 10

# Reference



1. [console.log\(\) method - W3Schools](#)
  2. [JavaScript comments - W3Schools](#)
  3. [JavaScript HTML DOM Methods - W3Schools](#)
  4. [Document: getElementById\(\) method - Web APIs | MDN](#)
  5. [JavaScript if else else if - W3Schools](#)
  6. [JavaScript const keyword - W3Schools](#)
  7. [JavaScript Operators - W3Schools](#)
  8. [JavaScript Assignment - W3Schools](#)
  9. [Loops and iteration - JavaScript | MDN](#)
  10. [JavaScript for Loop - FreeCodeCamp](#)
-

# Fonts & colors used

This presentation has been made using the following fonts:

## **Blinker**

(<https://fonts.google.com/specimen/Blinker>)

## **Inconsolata**

(<https://fonts.google.com/specimen/Inconsolata>)

#ffffff

#022a46

#72ffdd

#72fff

#0d3a58