1️⃣ HTML Structure - Basic Structure

DOCTYPE and Head

html

```
<!DOCTYPE html>
```

- **Purpose**: Tells the browser this page is written in HTML5
- **Reference**: [W3Schools - HTML DOCTYPE]

html

```
<meta charset="UTF-8" />
```

- **Purpose**: Sets character encoding to support all languages (Arabic, English, Chinese...)
- **Without UTF-8**: Arabic characters will appear as weird symbols ����

html

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

- **Purpose**: Makes the page responsive on mobile devices
- **Without it**: Page will appear very small on phones

2️⃣ HTML Table - The Table

Basic Table Structure

html

```
<table border="1" cellpadding="15">
```

Attributes Explained:

- `border="1"` : Border thickness = 1 pixel
- `cellpadding="15"` : Space between cell content and borders = 15 pixels

- **Reference**: W3Schools - HTML Tables

First Row - Main Header

html

~~~
<tr>
  <th colspan="9">Employee Database</th>
</tr>
```

**Detailed Explanation:**
- `<tr>`: Table Row - a row in the table
- `<th>`: Table Header - header cell (bold and centered text)
- `colspan="9"`: Cell spans across 9 columns horizontally

**Visual Representation:**
```
┌──────────────────────────────────────┐
│      Employee Database               │ (spans 9 columns)
└──────────────────────────────────────┘
~~~

Second Row - Column Headers

html

~~~
<tr>
  <td></td>
  <td>name</td>
  <td colspan="3">employee ID</td>
  <td colspan="2">phone</td>
  <td>email</td>
  <td>department</td>
</tr>
```

**Detailed Explanation:**
- Empty cell `<td></td>` at the beginning
- `employee ID` spans 3 columns
- `phone` spans 2 columns
- Total: 1 + 1 + 3 + 2 + 1 + 1 = **9 columns** ✅
~~~

~~~
**Visual:**
```
~~~

	name	employee ID	phone	email	department

Data Rows (5 rows)

html

~~~html
<tr>
  <td rowspan="5">name</td>
  <td>Fatima Ali</td>
  <td colspan="2">E001</td>
  <td colspan="2">01012763893</td>
  <td rowspan="5">email</td>
  <td>sara@gmail.com</td>
  <td>Hr</td>
</tr>
```


**Explaining `rowspan="5"`:**
- Cell spans **vertically** across 5 rows
- The words "name" and "email" appear only once for all 5 employees

**Visual:**
```
~~~

name	Fatima Ali	E001	01012763893	email	sara@gmail..	Hr
	Fatima Ali	E001	01012763893		sara@gmail..	Hr
	Fatima Ali	E001	01012763893		sara@gmail..	Hr
	Fatima Ali	E001	01012763893		sara@gmail..	Hr
	Fatima Ali	E001	01012763893		sara@gmail..	Hr

Empty Row + Total

html

```html
<tr>
  <td colspan="9"></td>
</tr>
<tr>
  <td colspan="2">total employees</td>
  <td colspan="7" style="text-align: right">5</td>
</tr>
```

Explanation:

- Empty row to separate data from total
- `text-align: right` : Aligns text to the right
- Number "5" spans across 7 columns

3️⃣ JavaScript - Detailed Explanation

💾 Variables - Variables

javascript

```javascript
var x = 9
var y = "Abdullah"
var flag = true
var number = 9.8
```

Data Types:

Type	Example	typeof
Number	`9`	"number"
String	`"sara"`	"string"
Boolean	`true`	"boolean"
Undefined	`undefined`	"undefined"

Reference: [W3Schools - JS Variables]

➕ Arithmetic Operations

String Concatenation

javascript

```
console.log("ahmed" + 20)  // "ahmed20"
```

- When adding string + number, the number converts to string

Type Coercion (Automatic Conversion)

javascript

```
console.log(20 + 30 + "10")  // "5010"
```

Step-by-step Explanation:

1. `20 + 30` = `50` (number)
2. `50 + "10"` = `"5010"` (string)

Different Scenarios:

javascript

```
console.log("10" + 20 + 30)  // "102030" ❌
console.log(20 + 30 + "10")  // "5010" ✅
console.log("10" - 5)        // 5 (minus converts to number)
console.log("sara" - 5)      // NaN (Not a Number)
```

Reference: W3Schools - JS Type Conversion

🔢 Boolean and Number Conversion

javascript

```
console.log(true == 1)    // true
console.log(false == 0)   // true
console.log(12 + true)    // 13 (true = 1)
console.log(12 + false)   // 12 (false = 0)
```

Conversion Table:

Value	When Converted to Boolean	When Converted to Number
`0`	`false`	`0`
`1`	`true`	`1`
`""`	`false`	`0`
`"0"`	`true`	`0`
`null`	`false`	`0`
`undefined`	`false`	`NaN`
`NaN`	`false`	`NaN`

⚖️ Comparison Operators

== vs ===

javascript

```
// == Loose Equality (compares value only)
console.log(8 == "8")     // true
console.log(1 == true)    // true

// === Strict Equality (compares value and type)
console.log(8 === "8")    // false
console.log(1 === true)   // false
```

Advanced Examples:

javascript

```
console.log(null == undefined)   // true (special case)
console.log(null === undefined)  // false (different types)
console.log([] == [])            // false (different objects)
console.log(NaN === NaN)         // false (NaN ≠ anything, even itself!)
```

Reference: W3Schools - JS Comparisons

🔄 Type Conversion Functions

javascript

```
Number(null)      // 0
Number(undefined) // NaN
Number(true)      // 1
Number("123")     // 123
Number("sara")    // NaN

String(123)       // "123"
String(true)      // "true"

Boolean(0)        // false
Boolean("")       // false
Boolean(89)       // true
Boolean("0")      // true
```

🤔 Special Cases

javascript

```
// null in operations
console.log(null + null)      // 0
console.log(null + 0)         // 0
console.log(20 + null)        // 20

// undefined in operations
console.log(undefined + undefined) // NaN
console.log(undefined + 5)         // NaN

// NaN in operations
console.log("sara" - 40)      // NaN
console.log("50" - 40)        // 10 (minus converts)
console.log(1 - NaN)          // NaN
console.log(40 - true)        // 39

// Check for NaN
console.log(Number.isNaN("sara" - 7))  // true ✅
console.log(Number.isNaN(123))         // false
```

🎯 Control Flow

If-Else Statement

javascript

````
var time = 12
if(time < 9) {
   console.log("Good Morning")
}
else if(time < 14) {
    console.log("Good Day")    // ✅ Will print this
}
else {
    console.log("Good Evening")
}
````

Reference: [W3Schools - JS If Else]

Ternary Operator

javascript

~~~
var color = "red"
color == "red" ? console.log("red") : console.log("another color")
```


**Syntax:**
```
condition ? expressionIfTrue : expressionIfFalse
~~~

Examples:

javascript

````
var age = 20
var status = age >= 18 ? "adult" : "minor"  // "adult"

var score = 85
var grade = score >= 90 ? "A" : score >= 80 ? "B" : "C"  // "B"
````

Switch Statement

javascript

```
var day = 3
switch(day) {
    case 1:
        console.log("saturday")
        break;
    case 2:
        console.log("sunday")
        break;
    case 3:
        console.log("monday")  // ✅ Will print this and stop
    case 4:
        console.log("tuesday") // ⚠️ Without break, will also print this!
        break;
    default:
        console.log("invalid day")
        break;
}
```

⚠️ **Important Note:**

- Without `break` in case 3, execution continues to case 4 (Fall-through)

Reference: W3Schools - JS Switch

🔁 Loops

For Loop

javascript

```
for(var i = 0; i <= 3; i++) {
    console.log(i)  // 0, 1, 2, 3
}
```

Different Forms:

javascript

```javascript
// 1. Complete form
for(var i = 0; i <= 3; i++) {
    console.log(i)
}

// 2. Initialize i outside loop
var i = 0
for( ; i <= 3; ) {
    console.log(i)
    i++
}

// 3. Infinite Loop
for( ; ; ) {
    console.log("infinite loop")
    break;  // Must stop it!
}
```

Reference: W3Schools - JS For Loop

Continue & Break

javascript

```javascript
// Continue: skip current iteration
for(var i = 0; i <= 5; i++) {
    if(i == 3) continue;  // Will skip 3
    console.log(i)        // 0, 1, 2, 4, 5
}

// Break: stop loop completely
for(var i = 0; i <= 5; i++) {
    if(i == 3) break;     // Will stop at 3
    console.log(i)        // 0, 1, 2
}
```

💬 User Interaction

Prompt - Input Data

javascript

```
var name = prompt("Enter Your Name");
// Shows a dialog box for user input
```

Scenarios:

- User types "Ahmed" → `name = "Ahmed"`
- User clicks Cancel → `name = null`
- User clicks OK without typing → `name = ""`

Convert to Number:

javascript

```
var input = prompt("Enter a number")
var number = Number(input)
console.log(number + 10)
```

Reference: W3Schools - JS Popup Boxes

Confirm - Confirmation Message

javascript

```
var x = confirm("Are you sure?")
console.log(x)
```

Results:

- User clicks OK → `x = true`
- User clicks Cancel → `x = false`

📊 Summary of Possible Scenarios

Scenario 1: Opening the Page

1. HTML loads
2. Table displays with 5 employees

3. `script.js` loads (if exists)

Scenario 2: User Interaction

javascript

```
var name = prompt("Enter Your Name")
if(name) {
    alert("Welcome " + name)
} else {
    alert("No name entered")
}
```

Scenario 3: Data Validation

javascript

```
var age = prompt("Enter your age")
var ageNum = Number(age)

if(Number.isNaN(ageNum)) {
    alert("Please enter a valid number")
} else if(ageNum < 18) {
    alert("You are a minor")
} else {
    alert("You are an adult")
}
```

BY. Abdullah Ali