

Java script theory Assignment

1. What is JavaScript? What does it do in web development?

- JavaScript is a language that runs in your browser.
 - It makes web pages interactive and dynamic.
 - It can check forms, show animations, and talk to servers (APIs).
-

2. How is JavaScript different from Python or Java?

- JavaScript runs in browsers. Python and Java usually run on servers.
 - JS doesn't need a specific type for each variable (dynamic). Java does (static).
 - Python is made for simple code. JavaScript is made for events and actions.
 - JS runs without needing to compile. Java needs JVM, and Python needs an interpreter.
-

3. What is the `<script>` tag in HTML? How to link JS to HTML?

- `<script>` is used to add JavaScript to a webpage.
 - Inline example:
`<script>console.log("Hello")</script>`
 - External file:
`<script src="main.js"></script>` (Put it at the bottom before `</body>` or use `defer` in the `<head>`).
-

4. What are variables in JavaScript? How to declare them?

- Variables hold values like numbers or words.
 - **var**: old way, works inside functions.
 - **let**: newer way, works inside blocks ({}).
 - **const**: same as let, but value cannot change.
-

5. What are data types in JavaScript?

- **String**: "Hello"
 - **Number**: 42
 - **Boolean**: true, false
 - **Undefined**: declared but not given a value
 - **Null**: value is empty on purpose
 - **Object**: {name: "Tom"}
 - **Array**: [1, 2, 3]
 - **Symbol**: Symbol("id")
 - **BigInt**: very large number like 12345678901234567890n
-

6. What's the difference between undefined and null?

- **undefined**: variable exists but has no value.
 - **null**: value is empty on purpose.
 - **typeof undefined** → "undefined"
typeof null → "object"
-

7. What are the types of operators in JavaScript?

- **Arithmetic:** + - * / % **
Example: 5 + 2 = 7
 - **Assignment:** = += -= *= /=
Example: x += 5 is x = x + 5
 - **Comparison:** == === != !== < > <= >=
Example: 5 === '5' → false
 - **Logical:** && || !
Example: true && false → false
-

8. What is the difference between == and ===?

- ==: compares values only.
 - ===: compares values *and* types.
 - Example: 5 == '5' → true
5 === '5' → false
-

9. What is control flow? What is an if-else statement?

- Control flow decides what code runs and when.

```
let age = 18;  
if (age >= 18) {  
  console.log("Adult");  
} else {  
  console.log("Minor");  
}
```

10. What is a switch statement? When to use it?

- It checks a variable against many values.
- Use it when you have many if-else cases.

```
let color = "red";  
switch(color) {  
  case "red":  
    console.log("Stop");  
    break;  
  case "green":  
    console.log("Go");  
    break;  
  default:  
    console.log("Wait");  
}
```

11. What are the types of loops in JavaScript?

- **for** loop:

```
for (let i = 0; i < 3; i++) {  
  console.log(i);  
}
```

- **while** loop:

```
let i = 0;
while (i < 3) {
  console.log(i);
  i++;
}
```

- **do-while** loop:

```
let i = 0;
do {
  console.log(i);
  i++;
} while (i < 3);
```

12. while vs. do-while loop

- while: checks the condition first.
- do-while: runs at least once, then checks.

13. What is a function? How to write one?

- A function is code that does a task.

```
function greet() {
  console.log("Hello");
}
greet();
```

14. Function declaration vs. function expression

- **Function Declaration:**

```
function sayHi() { }
```

- It's hoisted (can run before declared).

- **Function Expression:**

```
let sayHi = function() { };
```

- Not hoisted.

15. What are parameters and return values?

- **Parameters:** inputs to a function.
- **Return value:** the output from a function.

```
function add(a, b) {
```

```
  return a + b;
```

```
}
```

```
let result = add(2, 3); // 5
```

16. What is an array? How to declare one?

- Array stores multiple values.

```
let arr = [1, 2, 3];
```

17. Array methods

- `push()`: adds at end → `arr.push(4)`
 - `pop()`: removes from end → `arr.pop()`
 - `shift()`: removes from start → `arr.shift()`
 - `unshift()`: adds at start → `arr.unshift(0)`
-

18. What is an object? How is it different from an array?

- Object stores data with names (keys).
- Array stores values in order.

```
let person = { name: "John", age: 30 };
```

19. How to access or change object values?

- **Dot notation**: `person.name = "Doe"`
 - **Bracket notation**: `person["age"] = 25`
-

20. What are events and event listeners?

- **Events**: actions like clicks or typing.
 - **Event listener**: waits for the event and runs code.
-

21. How does `addEventListener()` work?

```
document.getElementById("btn").addEventListener("click",
function() {
    alert("Clicked!");
});
```

22. What is the DOM? How does JS use it?

- DOM is a tree structure of the web page.
 - JS can use it to change text, images, etc.
-

23. DOM selection methods

- `getElementById("id")`: finds one element by ID.
 - `getElementsByClassName("class")`: finds all elements with that class.
 - `querySelector("selector")`: finds the first matching element using CSS selector.
-

24. What are `setTimeout()` and `setInterval()`?

- `setTimeout()`: runs code once after a delay.
 - `setInterval()`: runs code again and again with delay.
-

25. Example of `setTimeout()`

```
setTimeout(function() {
    console.log("Executed after 2 seconds");
```



```
}, 2000);
```

26. What is error handling? What are try-catch-finally?

- Helps catch and fix errors without stopping the whole program.

```
try {  
  let x = y + 1; // y is not defined  
} catch (err) {  
  console.log("Error:", err.message);  
} finally {  
  console.log("Cleanup");  
}
```

27. Why is error handling important?

- Stops the app from crashing.
 - Makes the app better for users.
 - Helps developers find and fix problems.
-