

Eg: <body>

<h2> demo on syntax error </h2> <script>

document.write("apple")

document.write("Banana")

</script>

<script>

document.write("Samsung");

document.write("Sony"); → without double quotes

</script>

Due to this } It is Shown error.
before write age, Syntax Error,
will be, all }
Code will be }
infected. (will not execute)

</body>

→ there are 5 literals in JavaScript.

Note:- 5) Literals:

String " " or ' '

Template ` ` (back tick)

array []

object { }

regular expression // (forward slash)

String Literal

- " " and ' '
- used to represent string/text.
- Since JS1.
- 1 line of text. (It is OK to text in html)

Template literal.

- (` `)
- used to ~~get~~ string/text.
- Since JS6.
- n lines of text (It can be like text area in html).

{ Yes * we can put HTML tags inside <script> tag but
we have to follow some rule & regulation.
otherwise it gives syntax ERROR }

- not supports interpolation
- supports interpolation.

e.g. <script>

```
document.write("HTML");
document.write('CSS');
document.write(`JavaScript`);
```

multiple line
are not allowed
in string
literal

console.log("simla
green
apples");

OR

```
console.log(` simla
green
apples `);
```

Allowed backtick
in multiple line
(Template Literal.)

</script>

e.g. <script>

<h2> Hello world </h2> → Error

// It will give error.

⇒ <h1> Hello </h1> → It will execute but not display.

document.write("<h1> Hello </h1>")

// executed.

not allowed.
give error.

document.write("<form action = \"login.php\">")

due to
quotation
syntax
error.

document.write("<form action = 'login.php'>")

allowed.

homogeneous type of quotations are not allowed. give error.

X allowed { document.write(`" text " id= "una" >
`);

✓ allowed { document.write(`<input type="password" id="pw" >

`);

allowed { document.write(`<label> Password </label>
`);

allowed { document.write(`<button> Log In </button>`);

document.write(`<form>`); } allowed - ✓

<script>
</body>
</html>

~~NOTE~~ combination of quotations of single / double / back tick.

| | | |
|---------|---------|----------|
| a → " x | l → " l | ' → " l |
| ' → " l | ' → " x | ' → " l |
| " → " l | " → " x | " → " x. |

Working with Variables :-

⇒ Variable is a reference name of a memory block.

⇒ Variables are created or stored in RAM (stack area).

⇒ Variables are used to store / to hold a value for reuse purpose & automatically substitute values in steps.

a - var
[100] value

→ reduce no. of line in code while reusing.

How to declare a variable?

⇒ We can define vars in JS three ways, those are:-
 ⇒ by using "Var"

only use these three Variable names:
 long, double float, int nothing allowed in JS it gives error.

Synt: var varname; ← declaration

or
 var varname = value; ← Initialization

⇒ by using "let" (since JS6)

both are ECMAs.

Synt: let varname;

or
 let varname = value; ← Initialization

⇒ by using "Const" (since JS6)

Synt: const varname = value; ← Initialization.

Where do we declare variables?

⇒ We can declare variables in open script tag, with in function or within block.

inside the
script i.e.
global
variable

(1) <script>
vars
</script>

(2) inside
the
fun.
i.e local
variable.

(3) block
scope.
vars

Rules for variable naming:-

- 1.) Name should start with an alphabet (a to z or A to Z), underscore (-), dollar (\$).
- 2.) After first character we can use digit (0 to 9).
- 3.) Variables are case sensitive.
for-ex:- a and A are different variables.
- 4.) Space is not allowed, means name should be single word.
- 5.) Special chars (symbols) are not allowed in name, expect _ and \$.
- 6.) Variable name length is not restricted (unlimited). But preferred small variable name for future.

Eg:-

<!-- Example on JS variables -->

<!-- In this program var & let will discuss -->

<script>

var a; // Var declaration:

document.write(a) # output undefined.

a = 100; // assigning.

Var b = 111; // var. init.

document.write(a)

document.write(b)

let x ; // var declaraⁿ.

document.write(x); // undefined output

x = 200; // assign.

let y = 300; // var init.

$x = 220 ;$ // modification

$b = 101 ;$

document. writeln(x)

document. writeln(b)

</script>

</body>

</html>