

Date : Day - 24  
Expt.No. :

Positive Task (Sheet)

Page.No:

i) Define an error in Javascript.

Ans: An error in Javascript is a problem that occurs while running or parsing your code, which stops the program or makes it behave incorrectly.

Type of Javascript errors

i) Syntax error:

Happens when Javascript cannot understand your code structure

e.g: missing ) or { } or ; or balanced

ii) Runtime error:

Happens while the code is executing.

e.g: Calling a function that doesn't exist.

iii) Logical error:

Code runs, but output is wrong due to incorrect logic.

e.g: Wrong formula or condition

In short: Errors are issues that break or affect the execution of Javascript code.

Q) List types of Javascript errors

Ans i) Syntax error ii) Runtime error

iii) Logical error.

3)

D) Difference between runtime and logical error.

Ans:

Feature Meaning

Runtime error

Error happens while the program is running

logical error

Code runs successfully but result is wrong

Steps

Yes, usually stops execution

No, program continues

Cause

Invalid operations  
(undefined variables,  
wrong function call)

Wrong logic / wrong condition / wrong formula

Easy to detect

Error (silence error message)

Handlers (no error message)

Example

"console.log(x)", where x is not defined

if ( $age > 18$ ) instead of  $>= 18$

4) Explain try / catch / finally ?

Ans: Try / catch / finally in JavaScript it's used for handling errors, so your program doesn't crash suddenly.

Date :  
Expt.No. :

Page.No:

### i) try :

The code that may cause an error in ~~Wellthen einrich~~  
try.

e.g: `try {  
 // risky code  
}`

### ii) catch :

If an error occurs in try, JavaScript jumps to  
catch card gives the error object.

e.g: `catch (error) {  
 // handle error  
}`

### iii) Finally :

This block always runs, whether errors happen or not.  
used for cleanup tasks (closing file, stopping  
listeners, etc.)

e.g: `finally {  
 // always executes  
}`

## try / catch / finally :-

Full example:

```
try {  
    let x = 10;  
    console.log(x/y); // y is not defined -> error  
}  
  
catch (error) {  
    console.log("Error occurred:", error.message);  
}  
  
finally {  
    console.log("This will run always");  
}
```

Output:

- Error occurred: y is not defined
- This will run always

5] What is the use of "throw"?  
In Javascript, throw is used to manually create (raise) an error in your program.

Throw used for:

- \* To stop execution when something is wrong
- \* To show a custom error message
- \* To send the error to catch block for handling.

Date :  
Expt.No. :

Page.No.:

Eg:

let age = 15;

try {

if (age < 18) {

throw "You are not eligible!";

}

Console.log ("Access granted");

catch (error) {

Console.log ("Error:", error);

}

Output

Error: You are not eligible!

In short:

Throw is used to generate your own error and handle it using try / catch.