Error Handling

Types of errors in JavaScript:

JavaScript supports the following list of errors,these are divided into

syntax error
runtime error
logical error

Syntax Error: Is called as parsing errors,occurs at compile time fortraditional programming languages at interpret time for JavaScript. Following example causes a syntax error because it is missing a closing parenthesis.

ex:

```
<body>
      <script type="text/javascript">
            document.write(; <-------syntax error
      </script>
</body>
```

Runtime Errors: These are called exceptions and these errors occurred at execution time. The following example causes a run time error because here syntax iscorrect but at run time it is trying to call a non existed method.

Eg:

```
<body>
      <script type="text/javascript">
            document.writeli("hello welcome");
      </script>
</body>
```

Logical errors: These can be most difficult error to find.These errorsoccurred if you make a mistake in the business logic.These errors unableto handle.

```
      <script>
//finding area of circle
          var r=10;
              var area=3.14*r*2;
          alert(area);
</script>
```

<u>Error handling in JavaScript</u>:

to work with error/exceptions use **try, catch, finally, throw** keywords.

try, catch and **finally** are blocks.

throw statement.

Try catch statement: This statement allows you to test a block of code for errors. The try block contains the code to be run & the catch block contains the code to be executed if an error occurs.

Syntax:

```
      try {
          code to run[break;]
       }
       catch(refvar) {
          code to run if an exception occurs[break;]
       }
         finally {
          Statements
```

```
    }
```

Error Object Properties

Property	Description
name	Sets or returns an error name
message	Sets or returns an error message (a string)

Syn: excep.name
excep.message
excep.description

List of Errors:

Error Name	Description
EvalError	An error has occurred in the eval() function (old ver)□
RangeError	A number "out of range" has occurred
ReferenceError	An illegal reference has occurred
SyntaxError	A syntax error has occurred □
TypeError	A type error has occurred
URIError	An error in encodeURI("uri") has occurred

Note:

> writing only try is not allowed.
> writing only catch is not allowed.
> writing only finally is not allowed.
> we can write any no.of try-catch blocks, and anywhere(within the script tag or external js).
> try block is successfully executed then catch block not executed.
>if try block is faired/thrown an error then only catch block is executed.
>Order of writing is 1st try block and then catch block.Meansdon't change order writing (otherwise it becomes error);
>One try is allowing only one catch block.

ex:

```
<head>
    <script type='text/javascript'>
        alet("welcome to exceptions");
        alert("Thank you");
    </script>
</head>
```

No o/p

In the above example we need to apply the try catch block

ex:

```
<head>
    <script type='text/javascript'>
        try{
            alrt("welcome to exceptions");
        }
        catch(e){
            alert(e.description);
        }
    </script>
</head>
```

above script get executed successfully

eval(): It is a global function stands for evaluate.It evaluates anumerical values.

syntax: eval(expression)

ex:

```
<head>
    <script type='text/javascript'>
        var x=prompt("Enter value to evaluate");
        alert(eval(x));
        alert("Next");
    </script>
</head>
```

In the above script if you enter the numerical value script get executedsuccessfully otherwise script unable to run. That time we should implement try catch block.

eg:

```
<head>
    <script type='text/javascript'>
        try{
            var x=prompt("Enter value to evaluate");
            alert(eval(x)) ;
        }
        catch(e) {
            alert("sorry alpha-invalid:"+e.description)
        }
        alert("next")
    </script>
</head>
```

.

finally block: This block gets executed regardless of an exceptionoccurring. this block sep used for writing some common code for try & catch block.

finally block should be follower of either try block or catch block.

syntax:

```
<script>
    try{
        code to run[break;]
    }catch(e){
        code to run if an exception occurs
        [break;]
    }finally{
        code that is always executed regardless of an exceptionoccurring
    }
</script>
```

eg:

```
<head>
    <script type='text/javascript'>
        try{
            var x=prompt("Enter value to evaluable");
            alert(eval(x));
        }catch(e){
            alert("sorry Alpha-Invalid:"+e.description);
        }finally{
            alert("This Block Always get executed");                    }
            alert("next")
        }
    </script>
</head>
```

Throw statement: This statement allows to you create an exception. If you use this statement together with try catch statement, you can control program flow and generate accurate error message. The exception can be String, integer, boolean or an object.

Syn:

throw excepobj;
throw "text";

Throw Exception

```
<body>
    <script type='text/javascript'>
        var x=prompt("Enter Any number")
        try{
            if(x>10) {
                throw "Err1";
            }
            else if(x<=10){
                throw "Err2";
            }
            else if(isNaN(x)){
                throw "Err3";
            }
        }catch(err){
            if(err=="Err1"){
                document.write("Error:The value is too height");
            }
            if(err=="Err3"){
                document.write("Error:The value is not a number");
            }
        }
    </script>
</body>
```