

# Exception handling

# Catching Errors

- Have you seen a dialog box telling us there is a runtime error and asking "Do you wish to debug?".
- Catching errors will deal with handling such types of error, so that user don't see this unpleasant message.
- By using the `try...catch` statement (available in IE5+, Mozilla 1.0, and Netscape 6)

# try...Catch Statement

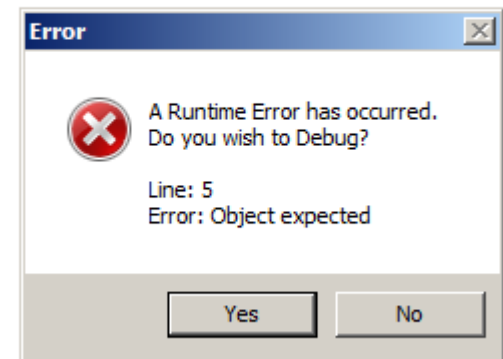
```
try {  
    //Run some code here  
} catch(err) {  
    //Handle errors here  
}
```

# Example

- The example below contains a script that is supposed to display the message "Welcome!" when you click on a button. But `alert()` is misspelled as `allert()`. A JavaScript error occurs.

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("Welcome!");
}
</script>
</head>
```

View message



```
<body>
<input type="button" value="View message"
onclick="message()" />
</body>
```

```
</html>
```

Let us see how to handle this error in a graceful manner so that user understands this message.

```
<html><head>
<script type="text/javascript">
var txt=""
function message() {
try {
    adddlert("Welcome guest!");
}
catch(err)
{
txt+="Error description: " + err.description ;
document.write("You see this page because some
problem occurred : " +txt);
}}
</script>
</head>
<body><input type="button" value="View message"
onclick="message()" />
</body></html>
```

# The Throw Statement

- The throw statement allows you to create an exception. If you use this statement together with the try...catch statement, you can control program flow and generate accurate error messages.
- **Syntax**
  - **`throw(exception)`**
- The exception can be a String, integer, Boolean or any object.

```
html><body>
<script type="text/javascript">
var x=prompt("Enter a number between 0 and
10:", "");
try{
if(x>10)
throw "Err1";
else if(x<0)
throw "Err2";
}
catch(er) {
if(er=="Err1")
alert("Error! The value is too high");
if(er == "Err2")
alert("Error! The value is too low");
}
</script>
</body></html>
```



# The onError Event

- To use the onError event, you must create a function to handle the errors.
- Then you call the function with the onError event handler.
- The event handler is called with three arguments: msg (error message), url (the url of the page that caused the error) and line (the line where the error occurred).

# onerror Syntax

```
onerror=handleErr
```

```
function handleErr(msg,url,l)
```

```
{
```

```
//Handle the error here return true or false
```

```
}
```

```
<html><head>
<script type="text/javascript">
onerror=handleErr;
var txt="";
function handleErr(msg,url,l) {
txt="There was an error on this page.\n\n";
txt+="Error is : " + msg + "\n";
txt+="URL is : " + url + "\n";
txt+="Line number is: " + l + "\n\n";
txt+="Click OK to continue.\n\n";
alert(txt);
return true;
}
function message() {
adddlert("Welcome guest!");
}
</script>
```

```
</head><body>  
<input type="button" value="View message"  
onclick="message()" />  
</body></html>
```

# Cookie

- A small piece of textual information sent by the server to the client, stored on the client's machine, and returned by the client's machine with each request made to the server.
- Web server sends cookies by sending the Set-Cookie response header.
- Cookies maintain information between more than one browser session.

# Components of a cookie

- `name= value`
- `comment= cmt`
- `domain= domainName`
- `expires = date`
- `path= path`
- Set-Cookie: uid=732423sdfs73242; expires=Fri, 31-Dec-2010 23:59:59 GMT; path=/; domain=.example.net

# Example-cookie

- This example displays a textbox which is empty first time it is viewed.
- After the name has been entered in the textbox and button is clicked, the cookie is set.
- Next time when the file is opened, the textbox has the name already set from the cookies.

```
<html><head>
```

```
<script>
```

```
function getCookie(name) {  
  if (document.cookie.length>0)  {  
    ind=document.cookie.indexOf(name) ;  
    if (ind!=-1)      {  
      ind=ind+name.length+1 ;  
      ind1=document.cookie.indexOf(";",ind) ;  
      if (ind1==-1)  
        ind1=document.cookie.length;  
      return  
        unescape (document.cookie.substring(ind,ind  
        1)) ;  
    }  
  }  
  return ""  
}
```



```
function setCookie(value) {  
var dt=new Date();  
dt.setDate(dt.getDate()+1);  
document.cookie="uname=" +escape(value)+" ;  
    expires="+dt.toGMTString();  
}  
  
function checkCookie() {  
username=getCookie('uname');  
if (username!=null && username!="") {  
    document.forms[0].uname.value=username;  
}  
}  
</script></head>  
<body onLoad="checkCookie()">  
<form>Name:<input type="text" name="uname">  
<input type="button" value="set"  
    onClick="setCookie(document.forms[0].uname.v  
    alue)">  
</form></body></html>
```