

Laboratorul 7

JavaScript - partea a II-a

1. Instrucțiunile “try...catch” și “throw”

1.1. Instrucțiunea “try...catch”

- Este utilizată pentru a detecta eventualele erori dintr-o secvență de cod
- Blocul “try” conține secvența de cod
- Blocul “catch” conține codul ce se execută în cazul apariției unor erori

```
try
{
    //Secvență de cod
}
catch(err)
{
    //Tratarea erorilor apărute în blocul “try”
}
```

Exemplu:

```
<html>
<head>
<script type="text/javascript">
var msg_err="";
function mesaj()
{
try
{
    alerta("Secventa de cod util!");
}
catch(err)
{
    msg_err ="Eroare!\n\n";
    msg_err += "Descriere: " + err.description + ".";
    alert(msg_err);
}
}
</script>
</head>

<body>
<input type="button" value="Apasa-ma!" onclick="mesaj()" />
</body>
</html>
```

1.2. Instrucțiunea “throw”

- Permite crearea excepțiilor
- Excepția poate fi de tip *string*, *integer*, *boolean* sau un obiect.
- Utilizarea sa împreună cu instrucțiunea “try...catch” permite controlul execuției programului precum și generarea de mesaje de eroare precise

Exemplu:

```
<html>
<body>
<script type="text/javascript">
var x=prompt("Introduceți un număr între 0 și 10:", "");
try
{
  if(x>10)
  {
    throw "Eroare1";
  }
  else if(x<0)
  {
    throw 2;
  }
  else if(isNaN(x))
  {
    throw true;
  }
}
catch(er)
{
  if(er=="Eroare1")
  {
    alert("Eroare! Valoarea este prea mare!");
  }
  if(er==2)
  {
    alert("Eroare! Valoarea este prea mică!");
  }
  if(er==true)
  {
    alert("Eroare! Introduceți un număr!");
  }
}
</script>
</body>
</html>
```

2. Programarea orientată obiect în JavaScript

2.1. Introducere

- JavaScript este un limbaj de programare orientat obiect
- Există obiecte JavaScript predefinite dar pot fi create și obiecte aparținând unor tipuri definite de utilizator

Exemplu utilizare membri

```
<html>
<body>
<script type="text/javascript">
var text="Hello World!";
document.write(text.length);
</script>
</body>
</html>
```

Exemplu utilizare metode

```
<html>
<body>
<script type="text/javascript">
var sir1="Hello world!";
var sir2="22";
document.write(sir1.toUpperCase()+"<br />");
document.write(parseInt(sir2));
</script>
</body>
</html>
```

2.2. Proprietăți și funcții JavaScript de tip global

- pot fi folosite cu toate obiectele JavaScript predefinite

Proprietăți JavaScript de tip global

Proprietatea	Descriere
Infinity	Valoare numerică ce reprezintă plus/minus infinit (+Infinity / -Infinity)
NaN	Valoare ce nu reprezintă un număr (NaN)
Undefined	Indică faptul că o variabilă nu este definită

Exemplu "undefined"

```
<html>
<body>
<script type="text/javascript">
var v1="";
var v2;
if (v1 == undefined)
{
document.write("Variabila v1 este nedefinită!");
}
if (v2 == undefined)
{
document.write("Variabila v2 este nedefinită!");
}
</script>
</body>
</html>
```

Funcții JavaScript de tip global

Proprietatea	Descriere
eval()	Evaluează un șir de caractere și-l execută ca și cum ar fi o bucată de cod
isFinite()	Determină dacă o valoare este finită
isNaN()	Determină dacă o valoare reprezintă sau nu un număr
Number()	Face conversia valorii unui obiect la un număr
parseFloat()	Face conversia unui șir de caractere la un număr de tip „float”
parseInt()	Face conversia unui șir de caractere la un întreg
String()	Face conversia valorii unui obiect la un șir de caractere

2.3. Obiectul “String”

Creare

```
var sir = new String("Continut sir");  
  
var sir = "Continut sir";
```

Membrii obiectului de tip String

Membru	Descriere
constructor	Returnează numele funcției cu care a fost creat prototipul obiectului
Length	Lungimea șirului
prototype	Permite adăugarea de membri și metode unui obiect

Metodele obiectului de tip String

Metoda	Descriere
charAt()	Caracterul de pe poziția indicată
concat()	Concatenează două sau mai multe șiruri
split()	Desparte un șir în mai multe subșiruri
substr()	Extrage un subșir dintr-un șir
toLowerCase()	Realizează conversia caracterelor unui șir în caractere minuscule
toUpperCase()	Realizează conversia caracterelor unui șir în caractere majuscule

Metode de încapsulare HTML a obiectelor de tip String

Aceste metode returnează șirul de caractere încadrat de etichetele HTML corespunzătoare

Metoda	Descriere
anchor()	Crează un element de tip ancoră ce conține șirul
bold()	Crează un element HTML de tip ce conține șirul
italics()	Crează un element HTML de tip <i> ce conține șirul
link()	Crează un element HTML de tip legătură ce conține șirul
strike()	Crează un element HTML de tip <strike> ce conține șirul
sub()	Crează un element HTML de tip <sub> ce conține șirul
sup()	Crează un element HTML de tip <sup> ce conține șirul

2.4. Obiectul "Array"

Membrii obiectului de tip Array

Membru	Descriere
constructor	Returnează numele funcției cu care a fost creat prototipul obiectului
Length	Numărul de elemente din obiectul Array
prototype	Permite adăugarea de membri și metode unui obiect

Metodele obiectului de tip Array

Metoda	Descriere
concat()	Concatenează două sau mai multe obiecte de tip Array
pop()	Returnează ultimul element din vector și-l șterge din vector
push()	Adaugă elemente noi la sfârșitul vectorului și returnează noua lungime
reverse()	Inversează ordinea elementelor în vector
shift()	Returnează primul element din vector și-l șterge din vector
sort()	Sortează elementele vectorului
unshift()	Adaugă elemente noi la începutul vectorului și returnează noua lungime

```
<html>
<body>

<script type="text/javascript">
var discipline = new Array();
discipline[0] = "PATR";
discipline[1] = "AI";
discipline[2] = "CSC";

var numere=new Array(23, 15, 3);

var logic=[true,false,true,true];

for (i=0;i < discipline.length;i++)
{
document.write(discipline[i] + "<br />");
}

for (i=0;i < numere.length;i++)
{
document.write(numere[i] + "<br />");
}

for (i=0;i < logic.length;i++)
{
document.write(logic[i] + "<br />");
}

</script>

</body>
</html>
```

2.5. Obiectul "Date"

Creare

```
var d = new Date();  
var d = new Date(milliseconds);  
var d = new Date(dateString);  
var d = new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

Membrii obiectului de tip Date

Membru	Descriere
constructor	Returnează numele funcției cu care a fost creat prototipul obiectului
prototype	Permite adăugarea de membri și metode unui obiect

Metodele obiectului de tip Date

Metoda	Descriere
getDate()	Ziua din lună (1-31)
getDay()	Ziua din săptămână (0-6)
getFullYear()	Returnează anul (4 cifre)
getHours()	Returnează ora (0-23)
getMilliseconds()	Returnează milisecundele (0-999)
getMinutes()	Returnează minutele (0-59)
getMonth()	Returnează luna (0-11)
getSeconds()	Returnează secunde (0-59)
toDate()	Realizează conversia părții de dată la un obiect de tip String
toString()	Realizează conversia la un obiect de tip String
getTime()	Realizează conversia părții de timp la un obiect de tip String

Exemplu

```
<html>  
<body>  
  
<script type="text/javascript">  
  
var data=new Date();  
var zi_sapt=new Array(7);  
  
zi_sapt[0]="Sunday";  
zi_sapt[1]="Monday";  
zi_sapt[2]="Tuesday";  
zi_sapt[3]="Wednesday";  
zi_sapt[4]="Thursday";  
zi_sapt[5]="Friday";  
zi_sapt[6]="Saturday";  
  
document.write("Today is " + zi_sapt[data.getDay()]);  
  
</script>  
  
</body>  
</html>
```

2.6. Obiectul "Boolean"

- Este utilizat pentru realizarea conversiei unei valori oarecare într-o valoare de tip Boolean (**true** sau **false**).

Creare

```
//Obiecte create cu valoarea initiala false
var flag = new Boolean();
var flag = new Boolean(0);
var flag = new Boolean(null);
var flag = new Boolean("");
var flag = new Boolean(false);
var flag = new Boolean(NaN);

//Obiecte create cu valoarea initiala true
var flag = new Boolean(true);
var flag = new Boolean("true");
var flag = new Boolean("false");
```

Membrii obiectului de tip Boolean

Membru	Descriere
constructor	Returnează numele funcției cu care a fost creat prototipul obiectului
prototype	Permite adăugarea de membri și metode unui obiect

Metodele obiectului de tip Boolean

Metoda	Descriere
toString()	Realizează conversia unei valori de tip Boolean într-un șir și returnează rezultatul

2.7. Obiectul "Math"

- Permite realizarea de calcule matematice
- Math nu este un constructor, ci un obiect al cărui membri și metode se pot accesa.

```
var x = Math.PI;           // Returnează PI
var y = Math.sqrt(16);    // Returnează radicalul din 16
```

Membrii obiectului de tip Math

Membru	Descriere
E	Valoarea constanta lui Euler (2.718)
LN2	Valoarea logaritm natural din 2 (0.693)
LN10	Valoarea logaritm natural din 10 (2.302)
LOG2E	Valoarea logaritm in baza 2 din E (1.442)
LOG10E	Valoarea logaritm in baza 10 din E(0.434)
PI	Valoarea PI (approx. 3.14159)
SQRT1_2	Radical din 1/2 (approx. 0.707)
SQRT2	Radical din 2 (approx. 1.414)

Exemplu:

```
<html>
<body>

<script type="text/javascript">

//generare numar aleator intre 0 si 1
document.write(Math.random() + "<br />");

//generare numar aleator intreg intre 0 si 10
document.write(Math.floor(Math.random()*11));

</script>

</body>
</html>
```

2.8. Crearea obiectelor definite de utilizator

a. Crearea unei instante a unui obiect

```
<html>
<body>
<script type="text/javascript">

individ=new Object();
individ.nume="Ionescu";
individ.prenume="Ion";
individ.virsta=50;
individ.ocupatie="somer";
document.write(individ.prenume + " are " + individ.virsta + " de ani.");

</script>
</body>
</html>
```

b. Crearea unui tipar pentru un obiect

```
<html>
<body>
<script type="text/javascript">
function individ( nume, prenume, virsta, ocupatie)
{
this.nume=nume;
this.prenume=prenume;
this.virsta=virsta;
this.ocupatie=ocupatie;
}

tata = new individ("Ionescu", "Ion", 35, "somer");
mama = new individ("Ionescu", "Ioana", 33, "somera");

document.write(tata.prenume + " are " + tata.virsta + " de ani. ");
document.write(mama.prenume + " are " + mama.virsta + " de ani.");

</script>
</body>
</html>
```