**ლექცია №7. ინტერნეტ დაპროგრამება. პირობითი ოპერატორები.**

პირობითი ოპერატორი if

მას აქვს სახე

**if**  (პირობა) {ოპერატორი ან ოპერატორები}

**if** საგასაღებო სიტყვაა და მისი მითითება აუცილებელია. ოპერატორი მუშაობს შემდეგნაირად :

თუ მრგვალ ფრჩხილებში მოთავსებული პირობა (მისი მნიშვნელობა ლოგიკური ტიპისაა) ჭეშმარიტია მაშინ შესრულდება ფიგურულ ფრჩხილებში ჩაწერილი ოპერა­ტორი ან ოპერატორები, შემდეგ კი მართვა გადაეცემა ფიგურული ფრჩხილების შემ­დეგ ჩაწერილ ოპერატორს. თუ პირობა მცდარია მაშინ ფიგურულ ფრჩხილებში ჩაწე­რილი ოპერატორები იგნორირდება და მართვა გადაეცემა ფიგურული ფრჩხილების შემდეგ ჩაწერილ ოპერატორს. თუ ფიგურულ ფრჩხილებში ერთი ოპერატორია, მაშინ ფიგურული ფრჩხილები შეიძლება არ დავწეროთ, მაგალითად.

ლისტინგი 1.

<html>

 <head>

 <title> if -ის მაგალითი </title>

<script language="javascript">

<!-- ვინიღბებით!

var a=1;

document.write("a="+a);

if (a==1) window.alert("მოიძებნა ერთი ერთეული!");

if (a==1) { window.alert("მოიძებნა ერთეული!");

 a=0;

document.write(" a="+a);

}

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

</body>

</html>

ჩამოვთვალოთ - გავიხსენოთ JavaScript-ში გამოყენებული პირობითი ოპერატორების ლოგიკური პირობების სახეები:

**= =** ტოლია ( და არა = ); = = = - ტოლია ტიპების გათვალისწინებით, მაგალითად 5 = = "5" ჭეშმარიტი ტოლობაა, რადგანაც მნიშვნელობები ემთხვევა, მაგრამ 5 = = = "5" მცდარია, რადგანაც ტიპებით შესადარებელი მნიშვნელობები განსხვავდება, ერთი რიცხვითია, მეორე სტრინგია.

**! =** განსხვავდება (არ უდრის), !== არ უდრისან მნიშვნელობით ან ტიპით.

 **< <= > >=**

პირობების კომპაქტურად ჩასაწერად იყენებენ ლოგიკურ, ანუ ბულის

ოპერატორებს. ესენია:

**| |** - ლოგიკური “ან” ოპერატორი,

**&&** - ლოგიკური “და” ოპერატორი,

**!**  - უარყოფის ოპერატორი.

მოვიყვანოთ გამოყენების მაგალითები:

ლისტინგი 2.

<html>

 <head>

 <title> რთული პირობები</title>

<script language="javascript">

<!-- ვინიღბებით!

phone=prompt("თქვენი ტელეფონის ნომერი","აქ შეიყვანეთ თქვენი ტელეფ. ნომერი ");

email=prompt("თქვენი e-mail-ი","აქ შეიყვანეთ თქვენი e-mail-ი");

if (phone==null||email== null) window.alert ("შეცდომაა აქ!");

if (phone==null&&email==null) window.alert ("შეცდომაა!");

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

</body>

</html>

**პირობითი ოპერატორი if...else**

მას აქვს სახე

**if** (პირობა) {ოპერატორი ან ოპერატორები}

**else** {ოპერატორი ან ოპერატორები}

პირობითი ოპერატორი **if...else** მუშაობს შემდეგნაირად: თუ პირობა ჭეშმარიტია მაშინ შესრულდება პირველ ფიგურულ ფრჩხილებში ჩაწერილი ოპერა­ტორი ან ოპერატორები, შემდეგ კი მართვა გადაეცემა **else** -ს შემდეგ ფიგურულ ფრჩხილებში ჩაწერილი ოპერა­ტორი ან ოპერატორების მომდევნო ოპერატორს. თუ პირობა მცდარია მაშინ არ შესრულდება პირველ ფიგურულ ფრჩხილებში ჩაწერილი ოპერა­ტორი ან ოპერატორები, მართვა პირდაპირ გადაეცემა **else** -ს შემდეგ ფიგურულ ფრჩხილებში ჩაწერილი ოპერა­ტორს ან ოპერატორებს, რის შემდეგაც დაიწყება ამ ფიგურული ფრჩხილების შემდეგ ჩაწერილ ოპერატორს.

მაგალითი: განვსაზღვროთ თებერვალში დღეების რიცხვი იმის გათვალისწი­ნებით წელიადი ჩვეულებრივია თუ ნაკიანი.

ლისტინგი 3

<html>

 <head>

 <title> რამდენი დღეა თებერვალში</title>

<script language="javascript">

<!-- ვინიღბებით!

var weli, febr=0;

weli=prompt("შეიყვანეთ წელი", "ჩაწერეთ წელიწადის რიცხვითი მნიშვნელობა");

if (weli%4==0){ febr=29}

else {febr=28}

 document.write(weli+" წლის "+"თებერვალში არის "+febr+" დღე");

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

</body>

</html>

ლისტინგი 4

<html>

 <head>

 <title> თუ ... მაშინ..</title>

<script language="javascript">

<!-- ვინიღბებით!

var a;

a=prompt("შეიყვანეთ რიცხვი 1","რიცხვის შეყვანა ხდება აქ");

if ( a == 1) {window.alert ("მოიძებნა ერთი ერთეული!");

a = 0;

}

else {

alert (" a ცვლადის არასწორი მნიშვნელობაა! " + a );

}

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

</body>

</html>

პროგრამირების თანამედროვე ენებში დასაშვებია ვისარგებლოთ ასეთი კონსტრუქციითაც:

value **= ( a == 1 ) ?** 1**:** 0;

იგი ტოლფასია შემდეგი ფრაგმენტის:

if ( a == 1)

value = 1;

else

value = 0;

მაგალითი, ლისტინგი 5

<html>

 <head>

 <title> if.esle ..alternativa.</title>

<script language="javascript">

<!-- ვინიღბებით!

a=prompt("შეიყვანე რიცხვი","აქ და ახლა");

value = ( a == 1 ) ? 1: 0;

document.write("პირველი გამოტანა="+a+" ");

if ( a == 1) value = 1;

else

value = 0;

document.write("მეორე გამოტანა="+a+" ");

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

</body>

</html>

**else if** პირობითი ოპერატორი გვაძლევს საშუალებას შევქმნათ ახალი if else ოპერატორი ჩავვათ, თუკი საწყისი პირობა არ სრულდება, მას აქვს სახე:

 if (პირობა1) {

 // კოდის ბლოკი, რომელიც უნდა შესრულდეს, თუ პირობა1 მართალია

} else if (პირობა2) {

 // კოდის ბლოკი, რომელიც უნდა შესრულდეს, თუ პირობა1 არის false(„მცდარი) და პირობა2 არის true („ჭეშმარიტი“)

} else {

 // კოდის ბლოკი, რომელიც უნდა შესრულდეს, თუ პირობა1 არის false(„მცდარი) და პირობა2 არის false(„მცდარი)}

მაგალითად კოდი:

if (time < 10) {

 greeting = "დილა მშვიდობის";

} else if (time < 20) {

 greeting = "დღე მშვიდობისა";

} else {

 greeting = "საღამო მშვიდობისა";

}

გვაძლეს საშუალებას ავიჩიოთ მისალმების ფორმა დღის ამა თუ იმ მონაკვეთისთვის. ანუ ჯერ ხდება განსაზღვრა - არის თუ არა time (დრო) 10 საათზე ნაკლები, მაშინ მისალმების ფორმაა: greeting = "დილა მშვიდობის"; თუკი დრო 10-ზე მეტია ან ტოლია, მაშინ მუშაობს პირველი else და იწყება შემოწმწბა იმ დროის რომელიც 10 საათზე მეტია - თუ იგი ნაკლებია საღამოს რვა საათისა, მაშინ მისალმების ფორმაა: greeting = "დღე მშვიდობისა"; წინააღმდეგ შემთხვევისა - ანუ მუშაობს მეორე else, რაც იმის მაჩვენეელია, რომ დრო საკმაო გვიანაა - საღამოს რვა საათია - ამიტომ გვექნება შესაბამისი მისალმებაც - greeting = "საღამო მშვიდობისა" .

**if** ოპერატორების ნაცვლად რიგ შემთხვევებში შეიძლება ვიხმაროთ **switch** ოპე­რატორი. იგი მოსახერხებელია გამოყენებულ იქნას იმ შემთხვევებში, როცა პირობა ტოლობას შეიცავს. ამასთან ტოლობა უნგა იყოს ტიპების დონეზე:

**switch** (გამოსახულება)

**{**

**case** ნიშანი 1**:**

//ოპერატორთა სია

**break**

**case** ნიშანი 2**:**

//ოპერატორთა სია

 **break**

**}**

ტიპების დონეზე ტოლობა ნიშნავს, მაგალითად: გამოსახულება = = = ნიშანი2.

ოპერატორთა სია თუ ერთი ოპერატორისგან შედგება, მაშინ მის მერე შეიძლება არ დავსვათ წერტილ-მძიმე, თუკი ამ ოპერატორის შემდეგ მოხდა კარეტკით გადასვლა შემდეგ სტრიქონზე. **break** შეიძლება იყოს წერტილ-მძიმის გარეშე, თუ ის მარტო წერია ერთ სტრიქონში. მაგრამ ყოველთვის უმჯობესია ოპერატორის შემდეგ დაისვას წერტილ-მძიმე. ოპერატორთა სიის გაერთიანება ფიგურულ ფრჩხილებში ბლოკის სახით სავალდებულო არაა, **case** და მომდევნო **case**, ან **break** ფიგურული ფრჩხილების მაგივრობას სწევენ.

მოვახდინოთ მისი მუშაობის დემონსტრირება შემდეგ მაგალითზე.

ლისტინგი 6

**<html>**

 **<head>**

 **<title>** ...ამორჩევა switcg...**</title>**

**<script language="javascript">**

<!-- ვინიღბებით!

**var yourchoice;**

**yourchoice = prompt**("აირჩიეთ მთელი რიცხვი 1-დან 4-ის ჩათვლით", "1, 2, 3 ან 4");

**switch (yourchoice)**

{

**case "1":**

**alert**("თქვენ აირჩიეთ 1"); }

**break;**

**case "2":**

**alert**("თქვენ აირჩიეთ 2");

**break;**

**case "3":**

**alert**("თქვენ აირჩიეთ З");

**break;**

**case "4":**

**alert**("თქვენ აირჩიეთ 4");

**break;**

**default:**

**alert**("თქვენ არ არჩიეთ მთელი რიცხვი რომელიც 1-დან 4-ის ჩათვლით მონაკვეთში მოხვდა");

**break**;

**}**

 // ვიხსნით ნიღაბს. -->

**</script>**

**</head>**

 **<body>**

**</body>**

**</html>**

რადგანაც **prompt()** ოპერატორს შეჰყავს სტრინგის ტიპის მონაცემები, ანუ შეყვანილი რიცხვითი მნიშვნელობა ხდება სტრინგის ტიპის, ამიტომაც გვიწერია case -თან რიცხვები ბრჭყალებში. რათა ისინიც იყვნენ სტრინგები და შეყვანილ რიცხვებს ტიპების დონეზე დაემთხვნენ. თუკი ამ ბრჭყალებს მოვაშორებთ, მასინ კოდი იქნება არაკორექტული. შეამოწმეთ ეს დამოუკიდებლად.

ზოგადად, **break** აუცილებლად უნდა დაისვასიმ ოპერატორის შემდეგ, რომელიც აირჩა, თუ **break** არ დაისვა, მაშინ **switch** -დან გამოსვლა არ მოხდება და შესრულდება მომდევნო ოპერატორი.

აი კიდევ ერთი მაგალითი:

**switch** (**button) {**

**case "მომდევნო გვერდი" :**

**window.location="next.html";**

**break;**

**case "წინა გვერდი" :**

**window.location=”prev.html”;**

**break;**

**case “საწყისი გვერდი” :**

**window.location="home.html";**

**break;**

**case "უკან დაბრუნება" :**

**window.location="back.html";**

**break;**

**default :**

**window.location="არა, მეგობრებო, ამ ღილაკზე ხელის დაჭერით ვერ გავფრინდებით!";**

**}**

მოვიყვანოთ სრული ტექსტი

ლისტინგი 7

<html>

 <head>

 <title> საიტის არჩევა</title>

<h3> აირჩიეთ საიტი </h3>

<script language="javascript">

<!-- ვინიღბებით!

where = window.prompt ("დღეს რომელ საიტს ვესტუმროთ?","posta , microsoft, yahoo" );

switch (where) {

 case "posta" :

window.location = "http://www.posta.ge";

break;

 case "microsoft" :

window.location = "http://www.microsoft.com";

break;

 case "yahoo" :

window.location ="http://www.yahoo.com";

break;

 default :

window.location = "http://www.kereseli.besaba.com";

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

</body>

</html>

მოვიყვანოთ პირობითი ოპერატორის მაგალითები confirm (პარამეტრი1) მეთოდის გამოყენებით, შევნიშნოთ, რომ მეთოდ confirm () -ს გამოაქვს მონიტორზე დიალოგური ფანჯარა რომელზეც არის ორი ღილაკი Cancel და OK. თუ კი დავაჭერთ OK ღილაკს, მაშინ ხდება true (ჭეშმარიტი)-ს დაბრუნება, ხოლო Cancel -ს, მაშინ დაბრუნდება მნიშვნელობა false (მცდარი). ცხადია ეს დაბრუნებული მნიშვნელობები შეიძლება მივანიჭოთ რაიმე ცვლადს, რომელსაც ცხადია ექნება boolean ტიპი, ხოლო ასეთი ტიპის ცვლადი შეიძლება მოთავსებულ იქნეს პირობითი ოპერატორის პირო-ბით ნაწილში. მაგალითი

ლისტინგი 8.

<html>

 <head>

 <title> ამორჩევა confirm -ით</title>

<script language="javascript">

<!-- ვინიღბებით!

var response=confirm(" გააგრძელებთ თუ არა javascript-ის შესწავლას ჩვენს ლექციებზე? დააჭირეთ ОК -ს თუ ასე მოიქცევით, თუ არა დააჭირეთ Cancel");

if (response == true )

{

alert("სწორი არჩევანია! გილოცავთ!")

}

else

{alert("დაფიქრდით კიდევ ერთხელ, შეცდომას უშვებთ");

}

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

</body>

</html>

რიცხვების დასალაგებლად გამოიყენება პირობითი ოპერატორი

ვთქვათ გვაქვს 4 რიცხვი a,b,c,d და უნდა დავალაგოთ ისინი

ლისტინგი 9

<html>

 <head>

 <title> ოთხი რიცხვის მნიშვნელობის დალაგება</title>

<script language="javascript">

<!-- ვინიღბებით!

 function maxmin (obj)

 { var a=Number (obj.num1.value);

var b=Number (obj.num2.value);

var c=Number (obj.num3.value);

var d=Number (obj.num4.value);

if (a>b) {r=a; a=b; b=r}

if (a>c) {r=a; a=c;c=r}

if(a>d){r=a;a=d;d=r}

if (b>c) {r=b;b=c;c=r}

if (b>d){r=b;b=d; d=r}

if (c>d){r=c; c=d; d=r}

obj.num1.value=a

obj.num2.value=b

obj.num3.value=c

obj.num4.value=d

}

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body>

<H4> ოთხი რიცხვის მნიშვნელობის დალაგება ზრდადობის მიხედვით </H4>

<FORM name=”form1”>

რიცხვი 1: <input type=”text” size=10 name="num1"><hr>

რიცხვი 2: <input type=”text” size=10 name="num2"><hr>

რიცხვი 3: <input type=”text” size=10 name="num3"><hr>

რიცხვი 4: <input type=”text” size=10 name="num4"><hr>

მაქსიმალური მნიშვნელობა ტოლია

<input type="button" value=დალაგება onClick="sortval(form1)"> <hr>

<input type ="reset" value= გასუფთავება>

</FORM>

</body>

</html>

(საშინაო დავალება -ეს ბოლო კოდი გასამართია)

აი, გამართული კოდი

ლისტინგი 9\_2

<html>

 <head>

 <title> ოთხი რიცხვის მნიშვნელობის დალაგება</title>

 <script language="JavaScript">

 function sortval(obj){

var a=Number(obj.num1.value);

var b=Number(obj.num2.value);

var c=Number(obj.num3.value);

var d=Number(obj.num4.value);

if (a>b) {r=a; a=b; b=r};

if (a>c) {r=a; a=c;c=r};

if(a>d){r=a;a=d;d=r};

if (b>c) {r=b;b=c;c=r};

if (b>d){r=b;b=d; d=r};

if (c>d){r=c; c=d; d=r};

obj.res1.value=a;

obj.res2.value=b;

obj.res3.value=c;

obj.res4.value=d;

}

</script>

</head>

 <body>

<H4> ოთხი რიცხვის მნიშვნელობის დალაგება ზრდადობის მიხედვით </H4>

<FORM name="form1">

რიცხვი 1: <input type="text" size=10 name="num1"><hr>

რიცხვი 2: <input type="text" size=10 name="num2"><hr>

რიცხვი 3: <input type="text" size=10 name="num3"><hr>

რიცხვი 4: <input type="text" size=10 name="num4"><hr>

დალაგებული რიცხვები

<input type="button" value=დალაგება onClick="sortval(form1)"><hr>

<input type="text" size=8 name="res1"><hr>

<input type="text" size=8 name="res2"><hr>

<input type="text" size=8 name="res3"><hr>

<input type="text" size=8 name="res4"><hr>

<input type ="reset" value= გასუფთავება>

</FORM>

</body>

</html>