**ინდერნეტ დაპროგრამება.** ლექცია 2

**Javascript**

შესავალი.

**WWW** (მსოფლიო აბლაბუდა) **Web**-დოკუმენტების შესაქმნელად თავდაპირ­ველად მხოლოდ **HTML** ენის შესაძლებლობებს იყენებდა. ამ მარტივი ენის დესკრიპტორებით (ელემენტებით) ხდება დოკუმენტის ფორმატირება, რაც ბროუზერს საშუალებას აძლევს **Web**-ფურცელი ავტომატურად ასახოს ეკრან­ზე. მაგრამ ინტერაქტიური (დიალოგის) ხასიათის ამოცანებისთვის საჭირო გახდა უფრო რთული, ე.წ. სცენარების მომზადების ენების გამოყენება. ისინი, ჩვენთვის ნაცნობი დაპროგრამების ენებისაგან განსხვავებით, არ საჭიროებენ პროგრამების კომპილაციას - მიმდევრობით სრულდება პროგრამის (რომელ­საც აქ სცენარი ეწოდება) თითოეული სტრიქონი, ანუ ხდება მისი ინტერპრე­ტირება. საკითხის ამგვარი გადაწყვეტა ძალზე აადვილებს სცენარების შექმნას და კორექტირებას - ყოველი ცვლილება ძალაში შედის ბროუზერის ფანჯარაში **Web** -ფურცლის ხელახლა გამოყვანისთანავე. სცენარების მომზადების ენებს შორის დრეს მშოფლიოში ყველაზე პოპულარულია კომპანია **Netscape Com­munication Corporation** -ის მიერ შექმნილი **Javascript** ენა. მასზე შექმნილი სცე­ნარები გასაგებია ყველა პოპულარული ბროუზერისთვის (მაგალითად, **VBScript** ენისაგან განსხვავებით, რომელიც მხოლოდ **Internet Explorer**-ზეა ორიენტირებული). **Javascript** ენის კონსტრუქციებს გარეგნულად ბევრი აქვს საერთო **Java** ენის შესაბამის კონსტრუქციებთან, მაგრამ ეს უკანასკნელი უფრო მძლავრი, მაშასადამე, უფრო რთული ენაცაა. **Java**-ზეც შეიძლება **Web**-ფურ­ცლებზე გამოსაყვანი ფრაგმენტებისათვის სპეციალური პროგრამების - ე.წ. **აპლეტების** დაწერა, მაგრამ ბროუზერის მიერ მათი უშუალოდ გამოყენება ვერ ხერხდება - აპლეტები ჯერ საკლასიფიკაციო ფაილებში უნდა იქნეს კომპი­ლირებული. არის სხვა სირთულეებიც, რის გამოც, თუ განსაკუთრებით რთული ამოცანების გადაწყვეტა არ გვიწევს, **Web** -ფურცლების შექმნისას უპი­რატესობას ვანჭებთ **Javascript** ენას.

**ჩვენი პირველი სცენარები**

**Notepad**  ტექსტურ რედაქტორში ავკრიფოთ პროგრამული კოდი, რომელშიც გამოყენებული იქნება მხოლოდ **HTML** ენის შესაძლებლობები:

ლისტინგი 1

<HTML>

<HEAD>

<TITLE>FIRST PAGE</title>

<STYLE>

H3, P {font-family: LitNusx}

</style>

</head>

<BODY>

<H3>კეთილი იყოს თქვენი მობრძანება ინტერნეტის სამყაროში! </h3>

<P> მაშ ასე, ორშაბათიდან ვიწყებთ ახალი საქართველოს შენებას!

</p>

</body>

</html>

**Javascript** ენაზე დაწერილ სცენარს კი განვალაგებთ დესკრიპტორების შემდე­გი წყვილის შიგნით: **<SCRIPT>** სცენარი **</script>** და მოვათავსებთ მას **HTML** ენაზე დაწერილ პროგრამაში:

ლისტინგი 2

<HTML>

<HEAD>

<TITLE>Next Page</title>

<STYLE>

H1, P {font-family: LitNusx}

</style>

</head>

<BODY>

<H1> კეთილი იყოს თქვენი მობრძანება ინტერნეტის სამყაროში! </h1>

<P> ვაგრძელებთ შრომით საქმიანობას! ვისახავთ ახალ მიზნებს. თან ოპერატიულად გატყობინებთ ჩვენი დოკუმენტის ბოლო ცვლილების თარიღს.

</p>

<SCRIPT LANGUAGE=”JavaScript”>

document.write(document.lastModified);

</script>

</body>

</html>

დავიმახსოვროთ ეს პროგრამები ჯერ **txt** და შემდეგ html გაფართოებით. ვხე­დავთ, რომ მეორე მაგალითში სცენარი მოთავსებულია **Web** -დოკუმენტის პროგრამის სხეულში. მაგრამ, საერთოდ, შესაძლებელია, იგი სათაურის უბანშიც განვალაგოთ. აშეთ შემთხვევაში სცენარი ჩატვირთვისთანავე არ სრულდება - იგი სხვა სცენარების მიერ გამოიძახება, როგორც *ფუნქცია*.

სცენარი **HTML**- დესკრიპტორშიც შეიძლება განვალაგოთ ე.წ. ხდომილობის დამმუშავებელი კონსტრუქციის სახით. ხდომილობის დამმუშავებლისთვის **<SCRIPT> </script>** ფრჩხილების გამოყენება საჭირო აღარ გახლავთ.

და ბოლოს, სცენარი შეიძლება გაფორმდეს **js** გაფართოების მქონე ფაილის სახითაც. პროგრამაში მისი გამოძახება ხორციელდება ფაილის სახელის ჩვე­ნებით **<SCRIPT>** წყვილის შიგნით.

აღვნიშნოთ, რომ სცენარში ხშირად მითითებულია **JavaScript** ენის ვერსიაც. ეს კეთდება იმ მიზნით, რომ ძველმა (მაშასადამე, ნაკლები შესაძლებლობების მქონე) ბროუზერებმა უნაყოფოდ არ სცადონ მათთვის გაუგებარი სცენარების შესრულება, რაც ზოგჯერ ბროუზერის “ჩამოკიდებასაც” კი იწვევს. დავუშვათ, გვსურს **Web** -ფურცელზე ავსახოთ ჩვენი დაბადების დღიდან გასული წამების რაოდენობა. გავითვალისწინოთ ის ფაქტი, რომ კომპიუტერში თარიღები მილიწამებში აითვლება და დავწეროთ შემდეგი კოდი:

ლისტინგი 03 wamebi 3

<HTML>

<HEAD>

<TITLE> ჩემი ასაკი წამებში </title>

<STYLE>

H2, P {font-family: LitNusx}

</style>

</head>

<BODY>

<H2> წამია კაცის ცხოვრება, მხოლოდ საკითხავია რამდენი? </h2>

<HR>

<SCRIPT LANGUAGE="JavaScript">

now=new Date();

y2k=new Date("jan 01 1981 00:00:00"); // აქ უჩვენეთ დაბადების თარიღი

seconds=(now-y2k)/1000;

document.write("<P> ჩემი დაბადებიდან გავიდა " + seconds + “წამი”);

</script>

</body>

</html>

დოკუმენტი ჩავტვირთოთ ბროუზერში. მივაქციოთ ყურადრება შემდეგ გა­რემოებას: **JavaScript** ენა ცვლადებსა და სხვა კონსტრუქციებში ერთმანეთისგან განასხვავებს დიდსა და პატარა ასოებს.

ჩავტვირთედ ლისტინგი 03 wamebi 3 და შედეგი ვერ მივიღეთ. მოვახდინოთ ამ კოდის მოდიფიცირება

ლისტინგი 03-1 wamebi

<HEAD>

<TITLE> ჩემი ასაკი წამებში </title>

<STYLE>

H2, P {font-family: LitNusx}

</style>

</head>

<BODY>

<H2> წამია კაცის ცხოვრება, მხოლოდ საკითხავია რამდენი? </h2>

<HR>

<script language="JavaScript">

var now = new Date();

var from1970 = (now.getTime() / (1000));

var ny = new Date(1981,0,1);

var fromYear = (ny.getTime() / (1000));

document.write("ჩემი დაბადებიდან გავიდა: " + ( from1970 - fromYear) + " წამი" + "<br>");

</script>

</body>

</html>

ეკრანზე გამოყვანილ რიცხვს წერტი­ლის შემდეგ მოსდევს სამი ათობითი ნიშანი (რასაც **JavaScript** დუმილით ით­ვალისწინებ), შედეგის დამრგვალებისთვის შეიძლება გამოვიყენოთ **Math.round** ს პეციალური ფუნქცია. იგი ოპერატორ **document. write** -ში ( from1970 - fromYear ) -ის წინ წინ უნდა ჩავსვათ:Math.round( from1970 - fromYear )

კორექტირების შემდეგ ბროუზერის ფანჯარაში **Web**-ფურცლის ხელახლა გამოსაყვანად დავაწკაპუნოთ **Refresh** ღილაკზე.

ლისტინგი 03-2 wamebi

<TITLE> ჩემი ასაკი წამებში </title>

<STYLE>

H2, P {font-family: LitNusx}

</style>

</head>

<BODY>

<H2> წამია კაცის ცხოვრება, მხოლოდ საკითხავია რამდენი? </h2>

<HR>

<script language="JavaScript">

var now = new Date();

var from1970 = (now.getTime() / (1000));

var ny = new Date(1981,0,1);

var fromYear = (ny.getTime() / (1000));

document.write("ჩემი დაბადებიდან გავიდა: " + Math.round( from1970 - fromYear ) + " წამი" + "<br>");

</script>

</body>

</html>

ვნახოთ, თუ როგორ რეაგირებს ბროუზერი შეცდომაზე. თუ ოპერა­ტორს მივცეთ შემდეგი სახე:

**Math.round (**from1970 - fromYear**;**

ფურცლის ჩატვირთვისას, ცხადია, თავს იჩენს შეცდომა. **Netscape Navigator** - ბროუზერის ბოლო ვერსიებში მისამართის ველში javascript: ინფორმაციის შეტანის შემდეგ ეკრანზე აისახება **JavaScript Console** სამუშაო ფანჯარა. მასში გამოიყვანება შეცდომის შემცველი სტრიქონი შეცდომის მიზეზის მითითებით (თუმცა ზოგჯერ ეს მიზეზი შეიძლება, არასწორად იყოს ახსნილი). ამავე ფანჯარაში გვეძლევა შეცდომის გამოსწორების შესაძლებლობაც.

ამჯერად, მიზნად დავისახოთ ამოცანაში გამოთვლილი დროის გამოყვანა წუთებშიც. **document.write** ოპერატორის შემდეგ კოდში ვცვლით **ფრაგმენტს: (from1970 - fromYear) + " წამი"** შემდეგი ფრაგმენტით **( from1970 - fromYear)/60 + " წუთი"**

**ლისტინგი 03-3** wuTebi

<HTML>

<HEAD>

<TITLE> ჩემი ასაკი წამებში </title>

<STYLE>

H2, P {font-family: LitNusx}

</style>

</head>

<BODY>

<H2> წამია კაცის ცხოვრება, მხოლოდ საკითხავია რამდენი? </h2>

<HR>

<script language="JavaScript">

var now = new Date();

var from1970 = (now.getTime() / (1000));

var ny = new Date(1981,0,1);

var fromYear = (ny.getTime() / (1000));

document.write("ჩემი დაბადებიდან გავიდა: " + ( from1970 - fromYear)/60 + "წუთი" +"<br>");

</script>

</body>

</html>

დამრვალების გათვალისწინებით მივიღებთ

ლისტინგი 03-4 wuTebi

<TITLE> ჩემი ასაკი წამებში </title>

<STYLE>

H2, P {font-family: LitNusx}

</style>

</head>

<BODY>

<H2> წამია კაცის ცხოვრება, მხოლოდ საკითხავია რამდენი? </h2>

<HR>

<script language="JavaScript">

var now = new Date();

var from1970 = (now.getTime() / (1000));

var ny = new Date(1981,0,1);

var fromYear = (ny.getTime() / (1000));

document.write("ჩემი დაბადებიდან გავიდა: " + Math.round(( from1970 - fromYear)/60 )+ "

წუთი" + "<br>");

</script>

</body>

</html>

**აქ document.write -ში** ( from1970 - fromYear)/60 ფრაგმენტი შევცვალეთ Math.round(( from1970 - fromYear)/60 ) -ით.

შევადგინოთ **JavaScript**-ის რედაქტორი HTML კიდით და იგი მოცემულია ლისტინგ 8-ში.

ლისტინგ 8

<html>

<head>

<title>javascript -ის რედაქტორი</title>

</head>

<body>

<h3>JavaScript -ის კოდების რედაქტორი</h3>

<p>კ ო დ ე ბ ი :</p>

<textarea id="firstcode" rows=12 cols= 60></textarea>

<p>შ ე დ ე გ ი :</p>

<textarea id="myrezult" rows=2 cols= 60></textarea>

<p>

<button onclick="document.all.myrezult.value=eval(firstcode.value)">შეასრულე</button>

<button onclick="document.all.firstcode.value=' '; document.all.myrezult.value=' '">წაშლა</button>

</p>

<p><!-- კომენტარი -->

შეიყვანეთ გამოსახულება ზედა ველში.<br>

გამოსახულებები გამოყავით ერთმანეთისგან წერტილ-მძიმით.<br>

ასევე შესაძლებელია სხვადასხვა სტრიქონში ჩაიწეროს ცალ-ცალკე გამოსახულება.<br>

ერთი სტრიქონის დასრულებისათვის და მეორე სტრიქონზე გადასვლისათვის დაჭირეთ კლავიშას - enter.

</p>

</body>

</html>

ქვედა ველში გამოვა ზედა ველში ჩაწერილი ბოლო გამოსახულების შედეგად მიღებული მნიშვნელობა. ზედა ველში ჩაიწერება **JavaScript**-ის ოპერატორები.

ჩვენ გავეცანით **JavaScript**-ის მეშვეობით შექმნილ რამდენიმე მარტივ სცენარს. ენის უფრო რთულ კონსტრუქციებს მომდევნო ლექციებში შევისწავლით.