ინტერნეტ დაპროგრამება. ლექცია №10. **ხდომილობა (მოვლენები) და მათი დამუშავება**.

**ხდომილობა.** ვებ-გვერდთან ურთიერთობისას წარმოიშვება ხდომილობები. ხდო­მილობა ეს არის სისტემის ერთგვარი შეტყობინება იმის შესახებ, რომ ადგილი აქვს რაღაც ქმედებების შესრულებას მომხმარებლის მიერ ან თავად სისტემაში შეიქმნა გარკვეული პირობები.

**ხდომილობათა სახეები**. ხდომილობები წარმოიქმნება ელემენტზე დაწკაპუნებით, მაუსის (თაგვის) გადაადგილებით, კლავიატურაზე დაჭერით, ფანჯრის ზომების შეცვლით, ვებ-გვერდის ჩატვირთვის დასრულებით და ა. შ.

შეგვიძლია ვებ-გვერდის ამა თუ იმ ელემენტის მიერ გენერირებული ხდომილობის­თვის დაიწეროს ამ ხდომილობის დამუშავების ***ფუნქცია*.** ზოგჯერ ხდომილობის და­მუშავება ხდება სპეციალურად შექმნილი ფუნქციის გარეშეც. ხდომილობის ეს დამუ­შავება შეიძლება იქნეს ჩაშენებული, მაგალითად ***ფორმის*** ატრიბურის სახით. მაგა­ლითად ფორმებით მონაცემთა გაგზავნისას წარმოიშვება ხდომილობა **onSubmit**. ამ ხდომილობის დადგომისას შეიძლება შევამოწმოთ ის მონაცემები, რომელიც მომხმა­რებელმა შეიყვანა და თუ ისინი არ შეესაბამებიან მოსალოდნელს, აღარ მოხდეს მათი გადაგზავნა, ვთქვათ სერვერზე.

ხდომილობას შეიძლება მივანიჭოთ **JavaScript**-ის ფუნქცია ან ობიექტის მეთოდი ისეთნაირად, რომ ფუნქციის ან მეთოდის კოდი შესრულებულ იქნეს ხდომილობის გამოძახებისთანავე. ამ პროცესს ***ხდომილობის დამუშავება*** ჰქვია.

*ხდომილობის დამუშავება* იძლევა საშუალებას დავაკავშიროთ **JavaScript**-ის ფუნქცია

ხდომილობას **JavaScript**-ის გამოცხადებით რა დროსაც გამოვიყენებთ სინტაქსის:

 **onevent=function(){**//JavaScript-ის კოდი**}**

ხდომილობის დამუშავების სახელი მიენიჭება დამუშავების ფუნქციას ან მეთოდს, რომელიც შესრულდება ხდომილობის გამოძახებისას. ამდენად ჩვენ გვაქვს: ხდომი­ლობა, ხდომილობის დამუშავება და ხდომილობის დამუშავების ფუნქცია. მაგალითად:

 **onclick=myFunction(){alert(“გამარჯობა მსოფლიოვ!”}**

ამ მაგალითში ხდომილობა (**event**) არის დაჭერა (დაწოლა, ზეწოლა) (**click**), **onclick** - ხდომილობის დამუშავება, ხოლო **myFunction()** - ქმედება, რომელსაც ჩაიდენს ფუნ­ქცია, როგორც კი ადგილი ექნება ხდომილობას. შევნიშნოთ, რომ ხდომილობის და­მუშავების ყველა დასახელება იწყება პრეფიქსით (თავსართით) **on** . ჩვენს მაგალითში **click**-ს ექნება ადგილი, ყოველთვის, როცა კი მომხმარებელი დააჭერს ობიექტს. მაგრა ხდომილობის დამუშავება არ იქნება გამოძახებული, თუ კი **click** შემთხვევა არ იქნება დაკავშირებული შესრულებად ფუნქციასთან ან მეთოდთან.

ხდომილობის დამუშავება არ ეკუთვნის ხდომილობას, ისინი მიეკუთვნებიან ობიექ­ტებს, რომლებიც იძახებენ ხდომილობას. ეს კი ნიშნავს, იმას, რომ, ხდომილობა შეიძ­ლება ”მიტაცებულ” იქნეს ერთზე მეტი ობიექტის მიერ, გააჩნია, თუ სად ხდება ეს ხდომილობა. მაგალითად, ხდომილობა **click** შეიძლება დაკავშირებული იყოს ვებ-გვერდთანაც და ამ გვერდზე არსებულ ღილაკთანაც. ხდომილობის დამმუშავებელი გამოიყენება ობიექტების მიერ: ვებ-გვერდის ან ღილაკის მიერ ხდომილობის მიტა­ცებისთვის და გარკვეული ქმედებების შესასრულებლად.

**მაუსის მოვლენები**

|  |  |
| --- | --- |
| **ატრიბუტი** | **აწერა** |
| onclick | მართვის ელემენტზე მოხდა მაუსის ღილაკის დაწკაპუნება. მოვლენა დადგება მაშინვე, როგორც კი ღილაკს აუშვებთ თითს.  |
| onmousedown | მაუსის ღილაკის დაჭერა.  |
| onmouseup | მაუსის ღილაკის აშვება.  |
| ondblclick | ორმაგი დაწკაპუნება.  |
| onmousemove | ელემენტის ხედვის არეში მოძრაობს მაუსის მიმთითებელი.  |
| onmouseout | ელემენტის ხედვის არიდან გავიდა მაუსის მიმთითებელი.  |
| onmouseover | ელემენტის ხედვის არეში იმყოფება მაუსის მიმთითებელი.  |

**კლავიატურის მოვლენები**

|  |  |
| --- | --- |
| onkeydown | კლავიატურაზე ღილაკს დააჭირეს.  |
| onkeypress | კლავიატურაზე ღილაკს დააჭირეს და არ აუშველ იმაზე მეტი ხნით, ვიდრე ეს არის განმეორების უნტერვალი - რომელიც სისტემური პარამეტრია და მომხმარებლის მიერ ოპერაციული სისტემის მომართვაზეა დამოკიდებული.  |
| onkeyup | კლავიატურის ადრე დაწერილი ღილაკი აუშვეს.  |

**ფორმისა და გვერდის ელემენტების მოვლენები**

|  |  |
| --- | --- |
| onchange | ელემენტმა დაკარდა შეყვანის ფოკუსი, ხოლო ელემენტის მნიშვნელობა შეიცვალა იმ დროში, სანამ ელემენტი იყო ფოკუსში.  |
| onselect | ელემენტს შიგნით ტექსტის რომელიღაც ნაწილი გამოყოფილ იქნა.  |
| onsubmit | ფორმაში დაჭერილია ღილაფი "გაგზავნა", მაგრამ სერვერზე ფორმა ჯერ არ გაგზავნილა.  |

|  |  |
| --- | --- |
| onblur | მართვის ელემენტი კარგავს შეყვანის ფოკუსს, ანუ კურსორი გადადის სხვა ელემენტზე.  |
| onfocus | გამოსახულმა ელემენტმა მიირო შეყვანის ფოკუსი.  |
| onload | დასრულდა გვერდის ჩამოტვირთვა.  |
| onunload | ხდება დოკუმენტიდან გამოსვლა (დახურვა ან სხვა გვერდის მისამართზე გადაყვანა).  |
| onscroll | ელემენტის სკროლირება - გასრიალება.  |

**ხდომილობის დამუშავება HTML-ის ტეგის შიგნით**

ხდომილება შეიძლება მინიჭებულ იქნეს HTML-ის ელემენტისთვის, ამ ელემენტის ტეგში თვისების დამატებით. გარკვეული ქმედებებისთვის, რომელსაც ადგილი უნდა ჰქონდეს, როცა მომხმარებელი დააჭერს ღილაკს, ხდომილობის დამმუშავებე­ლი **onClick** უნდა იყოს დაკავშირებული ღილაკთან. ზუსტად ასევე, რომ შესრულდეს სცენარი, მას შემდეგ რაც ჩამოიტვირთება დოკუმენტი, აუცილებელია ხდომილობის დამმუშავებელი **onLoad** დავნიშნოთ **<body>**-ის ელემენტთან. ამ მაგალითის კოდებია:

**<input type=”button” onClick=”alert(‘ეს არის alert ფანჯარა’)”>**

**<body onLoad=”doSomething()”>**

თუმცა კი HTML-ის სინტაქსისი არაა მგრძნობიარე რეგისტრების მიმართ, მიღებუ-ლია, რომ პირველი ასო ხდომილობისა ხდომილობის დამუშავების დასახელებაში ჩაიწეროს მაღალ რეგისტრში (მაგალითან ”**on**”+”**Click**”=**onClick**). როცა კი ხდომილო­ბის დამმუშავებელი გამოიყენება HTML-ის ტეგის გარეთ, მაგალითად JavaScript-ში, აუვილებელია ყველა დასახელება ჩაწერილ იქნენ დაბალ რეგისტრებში, რადგანაც JavaScript მგრძნობიერეა რეგისტრების მიმართ (მაგალითად HTML -ში **onClick**; JavaScript-ში **onclick**)

**ხდომილობის დამუშავება <script> ტეგის შიგნით**

კიდევ ერთი საშუალება იმისა, რომ მარტო ხდომილობა არის **<script>** ტეგის გამო­ყენება. ამ შემთხვევაში ხდომილობის დამმუშავებელი უნდა იყოს დაკავშირებული **<script>** ტეგთან. მაგალითად:

**<script language=”JavaScript” event=”onclick” for=”elementName”>**

**//ხდომილობის დამმუშავებელის ბრძანებები**

**</script>**

**<element id=”elementName”>content(შიგთავსი)</element>**

ხდომილობის დამმუშავებელი **<script>** ტეგში არ საჭიროებს ფუნქციის განსაზღვრე­ბას. შესაძლებელია ხდომილობისადმი გამიზნული კოდი განვათავსოთ **<script>** გამხსნელ და დამხურავ ტეგებს შორის.

**ხდომილობათა დამმუშავებლების დაბრუნებადი მნიშვნელობები**

ზოგჯერ ხდომილობის დამმუშავებელ სკრიპტში ბრძანებები არ სრულდება. ეს გამოწვეული შეიძლება იყოს იმის გამო, რომ ბრაუზერი არის ძველი და მას არაა აქვს სკრიპტების მხარდაჭერა, ანაც ფორმების გადაცემა ვერ ხერხდება ამათუ იმ მიზეზე­ბის გამო. იმისათვის, რომ ბრაუზერმა ხდომილობის დამუშავებისას არ მოგვცეს შეცდომის შეტყობინება, აუცილებელია დავუბრუნოთ მნიშვნელობები true და false

ან იმ ფუნქციის ბოლოს, რომელიც გამოიყენებოდა ხდომილობის დასამუშავებლად, ანაც გამოვაცხადოთ ახალი მნიშვნელობა ხდომილობის დასამუშავებელი ფუნქციის გამოძახების შემდეგ.

მაგალითად:

**<a href=”myDocument.html” onClick=”dosomething(); return false”> …</a>**

ამ მაგალითში დასაბრუნებელი მნიშვნელობა გადაეცემა ხდომილობის დასამუშავე­ბელი ატრიბუტებს შიგნით, ფუნქციის გამოძახების გარეთ. იგი ემიჯნება ფუნქციის გამოძახებას წერტილ-მძიმით, რითაც მიუთითებს, რომ ეს **JavaScript** -ის ახალი გამოცხადებაა.

**ხდომილობის დამმუშავებლები, როგორც ობიექტის პარამეტრები**

სკრიპტებში ხდომილობა შეიძლება იქნეს დანიშნული ობიექტისადმი, როგორც მისი თვისება. იმისათვის, რომ განვსაზღვროთ დამოკიდებულება ოპერატორსა და მის თვისებას შორის გამოიყენება ოპერატორი წერტილი (**.**)

მაგალითად, ვთქვათ ელემენტ **<button>**-ს სახელით **buttonName** , უკავია შემდეგი ადგილი დოკუმენტის ობიექტების იერარქიაში:

 **document.formName.buttonName.onclick = doSomething**

ამ მაგალითში ჩვენ ჩავწერეთ ხდომილობის დამუშავების დასახელება დაბალ რეგის­ტრებში, რადგანაც **JavaScript** მგრძნობიერეა რეგისტრების მიმართ. მივაქციოთ ყუ­რადღება იმას, რომ ფუნქციის დასახელებაში არაა მრგვალი ფრჩხილები. საერთოდ ფუნქციის დასახელების შემდეგ იწერება მრგვალი ფრჩხილები, მაგრამ ისინი არ დაისმის, როცა ფუნქციები უკავშირდება თვისებებს.

შემდეგი მაგალითი **msg()** ფუნქციას წარმოაჩენს, როგორც ხდომილობის დამუშავე­ბის **onClick** -ის ქმედებებს, მას შემდეგ, რაც დააწკაპუნებენ **myButton** ღილაკზე:

<script language=”JavaScript”>

document.myForm.myButton.onclick=msg;

function msg() {

alert(“შეტყობინება”);

}

</script>

<form name=”myForm”>

 <input type=”button” value=”Click me” name=”myButton”>

</form>

**ხდომილობის ინფორმაციის ”მიტაცება”**

როცა მომხმარებელი ან სისტემა ქმნიან ხდომილობას, იგი შეიძლება ”მიტაცებულ” იქნას ხდომილობის დამუშავებით.

**Internet Explorer 4** და **Netscape Navigator 4** გამოშვებამდე ხდომილობა შეიძლება ”მიტაცებულ” ყოფილიყო მნოლოდ როგორც **HTML**-ის ელემენტის ატრიბუტი, მაგალითად:

**<a onClick=”function()”></a>**

ამ მაგალითში ხდომილობის დამუშავება **onClick** - **<a>** ელემენტის ატრიბუტია. ხდომილობის მიტაცების ეს მოძველებული მეთოდიკა არაკორექტულია, რადგანაც, **HTML**-ის ატრიბუტი **onClick** მიუთითებს მხოლოდ იმას, რომ ხდომილობას ჰქონდა ადგილი, მაგრამ არ იძლევა დამატებით ინფორმაციას ამ ხდომილობის შესახებ. როცა მომხმარებელი დააჭერს ღილაკს გვერდზე, შესრულდება ხდომილობა **Click** ამ ღილაკით, რაც გამოიწვევს **onClick** ხდომილობის დამუშავების გამოძახებას. მაგრამ ჩვენ არ შეგვძლია მივუთითოდ არც ის თუ მაუსის რომელი ღილაკი იქნა გამოყენებული, ვერც ის თუ სად იყო (რომელ ადგილას) ეს დაჭერა.

**Internet Explorer** და **Netscape Navigator 4.0** ვერსიების შექმნის შემდეგ ხდომილებები გახდნენ დაპროგრამებადი ობიექტები თავიანთი თვისებებით და ამდენად შეიცავენ მნიშვნელოვან ინფორმაციას ხდომილობაზე. ეხლა კი ხდომილობა **Click** -ის თანმდე­ვი ინფორმაციაა, თუ რომელმა ღილაკმა მოახდინა ამ ხდომილობის გენერირება და იმ ადგილის კოორდინარებია ცნობილი, სადაც მოხდა მაუსის ღილაკის დაჭერა. შემდეგ მაგალითში მოცვემულია ეს სრულყოფა:

<script language=”JavaScript”>

 function theButton(){

var x=event.button;//

//

if(x==1){y=”დაჭერილ იქნა მაუსის მარცხენა ღილაკი.”}

if(x==2){y=”დაჭერილ იქნა მაუსის მარჯვენა ღილაკი.”}

if(x==3){y=”დაჭერილ იქნა მაუსის ორივე ღილაკი.”}

if(x==4){y=”დაჭერილ იქნა მაუსის საშუალო ღილაკი.”}

if(x==5){y=”დაჭერილ იქნა მაუსის საშუალო და მარცხენა ღილაკი.”}

if(x==6){y=”დაჭერილ იქნა მაუსის საშუალო და მარჯვენა ღილაკი.”}

if(x==7){y=”დაჭერილ იქნა მაუსის სამივე ღილაკი.”}

alert(y);//

}

</script>

<a id=”myL” href=”<http://www.microsoft.com/>” target=\_blank onMousedown=”theButton()”>Microsoft home page</a>

**მაუსის (თაგვის) ხდომილობები**:

მაუსის საშუალებით წარმოიქმნება შემდეგი ძირითადი ხდომილობები:

* **onMouseDown** - ვებ-გვერდზე ან მის ელემენტზე მაუსის ღილაკის (ნებისმი­ერის) დაჭერისას; აღსანიშნავია, რომ ხდომილობის დამუშავებისას, როცა გა­მოყენებული იქნება HTML-ის ტეგი და მასში ჩაიწერება ხსომილობის დასა­ხელება, რადგანაც ტეგში ლიტერის რეგისტრს მნიშვnელობა არა აქვს, ხდომი­ლობის დასახელება ჩაიწერება, ძირითადად კლავიატურაზე სწრაფი აკრეფის­თვის, დაბალ რეგისტრში. ისე როგორც ეს ნაჩვენებია ლისტინგ №1 -ში.

**ლისტინგი 1.**

<html>

 <head>

 <title> onMouseDown</title>

<script language="javascript">

<!-- ვინიღბებით!

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#0000CD">

გამარჯობა!

<input type="ღილაკი" value="დააჭირე აქ" onmousedown="alert('თქვენ ეხლახან

დააჭირეთ მაუსის ღილაკს')">

</body>

</html>

* **onMouseUp** – ადრე დაჭერილი მაუსის ღილაკის აშვებისას, ამ დროს მიმთითებელი ობიექტზეა.

 **ლისტინგი 2 onMouseUp**

<html>

 <head>

 <title> onMouseUp</title>

<script language="javascript">

<!-- ვინიღბებით!

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#0000CD">

გამარჯობა!

<input type="ღილაკი" value="დააჭირე აქ" onmouseup="alert('ეხლა კი თქვენ

აუშვით ღილაკს თითი')">

</body>

</html>

* **onClick**

**ლისტინგი 3. onClick**

<html>

 <head>

 <title> onClick</title>

<script language="javascript">

<!-- ვინიღბებით!

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#006400">

გამარჯობა!

<input type="ღილაკი" value="დააჭირე აქ" onclick="alert('ეხლა კი თქვენ

დააჭირეთ ღილაკს თითი')">

</body>

</html>

* **ondblclick** - მაუსის ორჯერ დაწკაპუნებით

**ლისტინგი 4 onDblclick**

<html>

 <head>

 <title> onDblClick</title>

<script language="javascript">

<!-- ვინიღბებით!

// ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#006400">

გამარჯობა!

<input type="ღილაკი" value="დააჭირე აქ" ondblclick="alert('ეხლა კი თქვენ

ორჯერ დააჭირეთ ღილაკს თითი')">

</body>

</html>

* **onMouseMove** – მაუსის ნებისმიერი გადაადგილებისას ელემენტზე;

**ლისტინგი 5** **onMouseMove**

<html>

 <head>

 <title> onMouseUp</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

alert("დიახ "+event.type+' ხდომილობას ჰქონდა ადგილი')

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#006400">

გამარჯობა!

<div id="myDiv" onmousemove="B1()">

გადაადგილეთ მაუსი განყოფილების არეზე

</div>

</body>

</html>

* **onMouseOver** - ვებ-გვერდის ელემენტზე მაუსის მიმთითებლის მიყვანა.

**ლისტინგი 6 onMouseOver**

<html>

 <head>

 <title> onMouseOver</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

alert("დიახ "+event.type+' ხდომილობას ჰქონდა ადგილი')

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#B22222">

გამარჯობა!

<div id="myDiv" onmouseover="B1()">

გადაადგილეთ მაუსი განყოფილების არეზე

</div>

</body>

</html>

* **оnMouseOu**t - ვებ-გვერდის ელემენტზე მაუსის მიმთითებლის გამოყვანა.

**ლისტინგი 7 onMouseOut**

<html>

 <head>

 <title> onMouseOut</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

alert("დიახ ბატონო! "+event.type+' ხდომილობას ჰქონდა ადგილი')

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#FF1493">

გამარჯობა!

<div id="myDiv" onmouseout="B1()">

გადაადგილეთ მაუსი განყოფილების არეზე

</div>

</body>

</html>

* **onMouseEnter** - როცა მაუსის მიმთითებელი ხვდება ელემენტის არეში;

**ლისტინგი 8 onMouseEnter**

<html>

 <head>

 <title> onMouseOut</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

alert("დიახ ბატონო! "+event.type+' ხდომილობას ჰქონდა ადგილი')

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#8B0000">

გამარჯობა!

<div id="myDiv" onmouseenter="B1()">

გადაადგილეთ მაუსი განყოფილების არეზე

</div>

</body>

</html>

* **onSelectStart - ელემენტის გამოყოფისას;**

**ლისტინგი 9 onSelectStart**

<html>

 <head>

 <title> onMouseOut</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

alert("W3C კონსორციუმის მთავარი გვერდი გაიხსნება მას მერე რაც დაიხურება alert ფანჯარა");

window.open("http://www.w3c.org","","")

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#8B0000" contenteditable=true>

<a style="text-decoration:underline" href="http://www.w3c.org" onselectstart="B1()">World Wide Web Consortium</a>

გამარჯობა!

</body>

</html>

* **onSelect -ტექსტის გამოყოფის დაწყება;**
* **onContextMenu** - მაუსის მარჯვენა ღილაკის დაჭერისას კონტექსტური მენიუს გამოყვანისთვის

**ლისტინდი 11 onContextMenu**

<html>

 <head>

 <title> onMouseOut</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

alert("ეს სურათი კოპირებადია");

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#8B0000" oncontextmenu="B1()">

<p> მარჯვენა ღილაკით სურათზე</p>

<img oncontextmenu="B1()" src="sokhumi.jpg" alt="" width="140" height="180">

</body>

</html>

თუ კი გვჭირდება ავკრძალოთ კონტექსტური მენიუს გამოყვანა <body> ტეგში დავამატოთ oncontextmenu="B1(); return false;"

**ლისტინგი 12 კონტექსტური მენიუს დაბლოკვა.**

 <html>

 <head>

 <title> onMouseOut</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

alert("ეს სურათი არაა კოპირებადი");

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body bgcolor="#FFEFFF" text="#8B0000" oncontextmenu="B1()">

<p> მარჯვენა ღილაკით სურათზე</p>

<img oncontextmenu="B1(); return false;" src="sokhumi.jpg" alt="" width="140"

height="180">

</body>

</html>

**კლავიატურის ხდომილობა**

* **onKeyDown** – კლავიატურაზე დაჭერისას;

**ლისტინგი 13 onKeyDown**

<html>

 <head>

 <title> onMouseOut</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

msg='on'+event.type+' ხდომილობას ადგილი აქვს '+'"'+event.srcElement.id+'"';

alert(msg);

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body id="myBody" onkeydow="B1()">

<p> დააჭირე ნებისმიერ კლავიშა რომ გამოიწვიო ხდომილობა</p>

</body>

</html>

* **onKeyPress** – მსაგავსია onKeyDown ხდომილობის, მაგრამ ამ შემთხვევაში კლავიატურაზე დაჭერისას გვიბრუნებს დაჭერილო ღილაკის - სიმბოლოს კოდს Unicode -ში; ხდომილობა მანამაა სანამ დაჭერილია კლავიში;

**ლისტინგი 14** **onKeyPress**

html>

 <head>

 <title> onkeypress</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

msg='on'+event.type+' ხდომილობას ადგილი აქვს '+'"'+event.srcElement.id+'"';

alert(msg);

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body id="myBody" onkeypress="B1()">

<p> დააჭირე ნებისმიერ კლავიშას რომ გამოიწვიო ხდომილობა</p>

</body>

</html>

* **onKeyUp** – ადრე დაჭერილი კლავიშის აშვებისას;

**ლისტინგი 15 onKeyUp**

<html>

 <head>

 <title> onkeyup</title>

<script language="javascript">

<!-- ვინიღბებით!

function B1(){

msg='on'+event.type+' ხდომილობას ადგილი აქვს '+'"'+event.srcElement.id+'"';

alert(msg);

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body id="myBody" onkeyup="B1()">

<p> დააჭირე ნებისმიერ კლავიშას რომ გამოიწვიო ხდომილობა და მერე

აუშვი</p>

</body>

</html>

* **onHelp -** <F1> კლავიშზე დაჭერისას.

**ლისტინგი 16** **onHelp**

<html>

 <head>

 <title> onkehelp</title>

<script language="javascript">

<!-- ვინიღბებით!

}

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body onhelp="alert("მომხმარებელს დახმარება ესაჭიროება")">

</body>

</html>

**დოკუმენტის ხდომილობა**

ჩამოვთვალოთ დოკუმენტის ხდომილობები

* **onLoad** –ვებ-გვერდის, სურთის, დოკუმენტის ჩატვირთვის შემდეგ;

**ლისტინგი 17** **onLoad**

<html>

 <head>

 <title> onload</title>

<script language="javascript">

<!-- ვინიღბებით!

 // ვიხსნით ნიღაბს. -->

</script>

</head>

 <body onload="alert('სურათი ჩამოიტვირთა სრულად')">

<IMG SRC="sokhumi.jpg" width="350" height="500">

</body>

</html>

* **onscroll** – გვერდის, დოკუმენტის, ფრეიმის სკროლოს(მოძრაობის) დროს;
* **onresize** – ფანჯრის ზომის შეცვლისას;
* **onBeforeUnload** – დოკუმენტის გადმოტვირთვისას;
* **onUnLoad** – უშუალოდ დოკუმენტის გადმოტვირთვის წინ, onBeforeUnload შემდეგ;
* **onBeforePrint** – დოკუმენტის ბეჭდვის წინ ან წინასწარი გაცნობისას;
* **onAfterPrint** - დოკუმენტის ბეჭდვის შემდეგ ან წინასწარი გაცნობისთვის გამოტანის შემდეგ;

**ფორმის ხდომილობები**

ფორმის ხდომილობებია:

* **onSubmit** – ფორმის მონაცემთა გადაგზავნისას;

**ლისტინგი 18 onSubmit**

<html>

 <head>

 <title> onsubmit</title>

<script language="javascript">

<!-- ვინიღბებით!

 // ვიხსნით ნიღაბს. -->

</script>

</head>

<body>

 <form name="form1" method="post" action="" nsubmit="alert('დაადასტურე ფორმის მონაცემები')">

<input type="text" name="textfield">

<input type="submit" value="Submit">

</form>

</body>

</html>

* **onReset** – ფორმის გასუფთავებისას;
* **onBlur** – ფორმის ელემენტების ფოკუსის დაკარგვისას;
* **onChange** – ტექსტურ ველში მონაცემთა შეცვლისას და ფოკუსის გადაადგილებისას ფორმის სხვა ელემენტზე ან ფორმის მონაცემთა გაგზავნის წინ ( ადგილი აქვს ამ ხდომილობას onBlur -ის წინ)
* **onFocus** - ფორმის ელემენტების ფოკუსის მიღებისას;