

ლაშა იაშვილი

https://gtu.ge/book/L_iashvili_HTML.pdf

hreflang

hreflang ატრიბუტი განსაზღვრავს დოკუმენტის ლინკის ენას, ენა ირჩევა *Language code*-დან, რომელსაც ქვემოთ შევხებით, ქართული *Language code* არის *ka*, მაგალითში ნაჩვენებია ილუსტრირება მოცემული ატრიბუტისა.

მაგალითი 1.22

```
<!DOCTYPE html><html>
<head>
<meta charset="utf-8">
</head>
<body>
<p>
<a hreflang="ka" href="http://scripts.ge">SCRIPTS.GE</a>
</p>
<p>
<a hreflang="ka" href="http://gtu.ge">საქართველოს ტექნიკური
უნივერსიტეტი </a>
</p>
</body>
</html>
```

[SCRIPTS.GE](http://scripts.ge)

[საქართველოს ტექნიკური უნივერსიტეტი](http://gtu.ge)

rel

rel ატრიბუტს დამხმარე ატრიბუტია, რომელიც ურთიერთქმედებს არსებულ დოკუმენტსა და მიზნულ დოკუმენტს შორის, მოცემულ ატრიბუტს აქვს თავის სიდიდე:

ცვლადი	აღწერა
<i>alternate</i>	ალტერნატიული ლინკი დოკუმენტის მაგალითი, საბეჭდი გვერდი,თარგმანი და ა. შ.
<i>author</i>	დოკუმენტის ავტორი
<i>bookmark</i>	ჩასანიშნი ლინკი
<i>help</i>	დოკუმენტის help-ი
<i>License</i>	დოკუმენტის საავტორო უფლებები
<i>next</i>	შემდეგი დოკუმენტის მონიშვნა
<i>noFollow</i>	საძიებო სისტემამ რომ ვერ იპოვოს კონკრეტული ლინკი
<i>noreferrer</i>	ბრაუზერი არ აგზავნის <i>HTTP header</i> -ში ნახსენებ <i>hyperLink</i> ებს

ლაშა იაშვილი

https://gtu.ge/book/L_iashvili_HTML.pdf

<i>prefetch</i>	ქეშირებული დოკუმენტის მონაცემები
<i>prev</i>	წინა დოკუმენტი ჩვენება
<i>search</i>	დოკუმენტში ძებნა
<i>tag</i>	საკვანძო სიტყვები დოკუმენტებში
<i>stylesheet</i>	დიზაინი- CSS ფაილთან ურთერთობა

მაგალითი

1.23

```
<!DOCTYPE
```

```
html>
```

```
<html><body>
```

```
<p>
```

```
<a rel="nofollow" href="http://script.ge"> არ მოძებნის ამ საიტს </a>
```

```
</p>
```

```
</body>
```

```
</
```

```
ht
```

```
ml
```

```
>
```

<Link>

rel ატრიბუტში მონათესავე ტეგი, რომლის საშუალებით ხდება დოკუმენტში, ლინკების მითითება, პრაქტიკაში დიზაინის ფაილის მისამართებს აღწერენ ხოლმე, რაც შეეხება, ატრიბუტებს იგივე ატრიბუტები აქვს, რაც *rel*-ს.

მაგალითი

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
</head>
```

```
<link rel="stylesheet" type="text/css" href="/დიზაინი.css">
```

```
</head>
```

```
<body>
```

```
</body> </html> მოცემული ტეგი მხოლოდ და მხოლოდ გაიწერება
```

```
<head> ტეგებს შორის </head>.
```

გლობალური ატრიბუტები

ორიოდე სიტყვით გლობალურ ატრიბუტს, ეს ისეთი ატრიბუტია, რომელიც ნებისმიერი ტეგის შემთხვევაში შეგვიძლია გამოვიყენოთ, ქვემოთ მოცემულია ცხრილის სახით, საჩამ გავარჩევდეთ თითოეულ მათგანს უნდა, შევხვით DOM-ს, რადგანაც ზოგი მათგანი DOM-ის ელემენტებს შეიცავს.

ატრიბუტები	აღწერა
------------	--------

accesskey	კლავიატურის ერთგვარი იარაღი (shortcut)
class	კლასი იქმნება, სადაც შეგვიძლია ერთი ან მეტი სახელის დარქმევა ელემენტისათვის
contenteditable	შესაძლებელია ტექსტის ჩასწორება
contextmenu	მენიუს კონტექსტი, რომლის მხარდაჭერა მხოლოდ <i>firefox</i> -ს აქვს
data-*	მონაცემების შემცველის გვერდის ამოვარდნა
dir	დირექტორი, რომელიც განსაზღვრავს მიმართულებას
draggable	სადაც შესაძლებელია ტექსტისა და სურათის გადაადგილება
dropzone	იძლევა საშუალებას ჩამოშლილი ტექსტის ან კოპირების სხვა ადგილას გადატანას, სამწუხაროდ, ამ ატრიბუტს მხარდაჭერა არცერთი ბრაუზერისგან არ აქვს
hidden	ტექსტის დამალვა
id	რომელიც უნიკალურია
Lang	განსაზღვრავს ენას, თუ ტექსტი რომელ ენაზეა დაწერილი
spellcheck	განსაზღვრავს მართლწერას
style	სტილისტური ატრიბუტი
tabindex	კლავიატურაზე <i>tab</i> ღილაკით ნავიგაციაზე გადასვლა
title	სათაური, როგორც ატრიბუტი, შეიცავს ტექსტზე ინფორმაციას
translate	თარგმნის, ამ ტეგის მხარდაჭერა არცერთ ბრაუზერს არ აქვს

რა არის DOM?

DOM (*Document Object Model*)-ანუ დოკუმენტის ობიექტური მოდელი, რომელიც HTML დოკუმენტის ელემენტებს წარმოადგენს, როგორც ობიექტის ელემენტს და აძლევს შესაძლებლობას, მის მეშვეობით მოხდეს გარკვეული ცვლილება HTML-ის დოკუმენტში, თითოეული ატრიბუტის განხილვისას შევცხებით DOM-საც, რომელიც *Javascript*-ს სინტაქსის შემადგენელი ნაწილია.

Javascript არის პროგრამირების ენა, რომელიც *HTML* და *CSS* დახმარებით ვებუბნებით Web საიტს რისი გაკეთება გვსურს; მაგალითად ტექსტის შეცვლა, კალკულაცია და ა. შ

✎ იმიტაციისათვის მოცემულია პატარა სკრიპტი

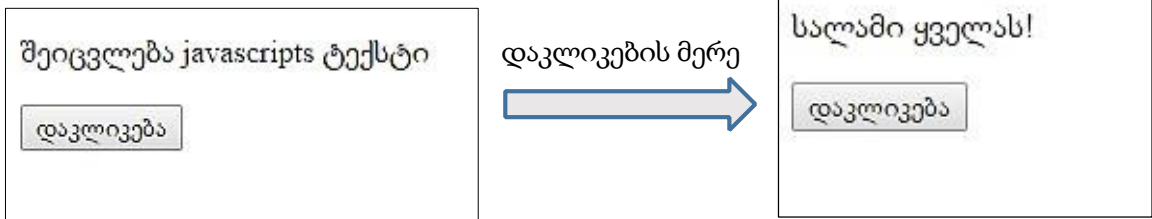
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p id="html">შეიცვლება javascripts ტექსტი</p>
<button type="button">
```

ლაშა იაშვილი

https://gtu.ge/book/L_iashvili_HTML.pdf

```
onClick="document.getElementById('html').innerHTML = 'სალამი  
ყველას!'">  
დაკლიკება  
</button>  
</body>  
</html>
```

ესეც შედეგი:



HTML წინამდებარე ვერსიაში აღწერა ხდებოდა შემდეგი სახით:

```
<div id="mydiv" brand="toyota" model="rav  
4"> Rav4 კარგი მოდელია.  
</div> ხოლო შემდეგ შეიმუშავეს ატრიბუტი data, რომელიც  
მონაცემებს აღწერდა  
<div id="mydiv" data-brand="toyota" data-  
model="rav4"> Rav4 კარგი მოდელია.  
</div> var
```

```
mydiv=document.getElementById('mydiv')  
// ჯავასკრიპტის სკრიპტის აღწერისას var  
brand=mydiv.dataset.brand //დააბრუნებს  
„toyota“-ს  
mydiv.dataset.brand='Volkswagen' // შეცვლის „data-brand“  
„Volkswagen“ mydiv.dataset.brand=null //წაშლის „data-brand“  
ატრიბუტს
```

რაც შეეხება // ამ სიმბოლოს ეს ჯავასკრიპტში ერთი სტრიქონიანი კომენტარია გაკეთებული როგორც HTML-ში <!--მოთავსებული ტექსტი--> .

Accesskey

არსი ის არის, რომ როდესაც მივუთითებთ რაიმე საკვანძო სიტყვას, კლავიატურაზე კომბინაციით შესაძლებელია მოვახდინოთ მარტივი გადამისამართება. ამ შემთხვევაშიც სხვადასხვა ბრაუზერის შემთხვევაში სხვადასხვანაირად ხდება ხოლმე:

- IE, Chrome, Safari, Opera: [ALT] + accesskey
- Opera 15 წინამდებარე ვერსიას: [SHIFT] [ESC] + accesskey

ლაშა იაშვილი

https://gtu.ge/book/L_iashvili_HTML.pdf

☞ *Firefox:[ALT] [SHIFT] + accesskey*

მაგალითი 1.24

გვაქვს მოცემული სხვადასხვა უნივერსიტეტის საიტი, რომლის კლავიატურაზე კომბინაციით გვინდა, გადამისამართდეს შესაბამის საიტზე:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<a href="http://www.gtu.ge" accesskey="g"> სტუ
</a> <a href="http://seu.edu.ge/"
accesskey="s">სეუ </a>
<a href="http://tsu.ge" accesskey="t">თსუ </a>
</body>
</html>
```

თუ ვიყენებთ *google Chrome*-ს და დავაწვებით [ALT] + გ, ფაილი ავტომატურად გადავა ტექნიკური უნივერსიტეტის საიტზე, იმავე კომბინაციით და დავაწვებით s-ს , მაშინ სეუ-ს საიტზე და ა. შ.

class-კლასი, კლასის შემთხვევაში შეგვიძლია გვქონდეს უამრავი კლასი სხვადასხვა სახელებით. კლასებთან დაკავშირებით ვილაპარაკეთ როდესაც *div* ტეგს შევხებით.

contenteditable-შეგვიძლია მოთავსებულ ტეგებში არსებული ტექსტის რედაქტირება, მაგრამ ასევე აქვს თავისი ატრიბუტები, რომლის საშუალებით განისაზღვრება რედაქტირების უფლებები შესაძლებელია თუ არა.

true და *false*, თუ მივუთითებთ *true*-ს, მაშინ მოვახდენთ ტექსტის რედაქტირებას, *false*-ს შემთხვევაში ვერა.

სინტაქსი გამოიყურება შემდეგნაირად

◀ ელემენტი *contenteditable="true ან false"* ▶

მაგალითი 1.26

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p contenteditable="true">
შესაძლებელია ტექსტის რედაქტირება
</p>
<p contenteditable="false"> არ
არის შესაძლებელი ტექსტის
რედაქტირება
```

ლაშა იაშვილი

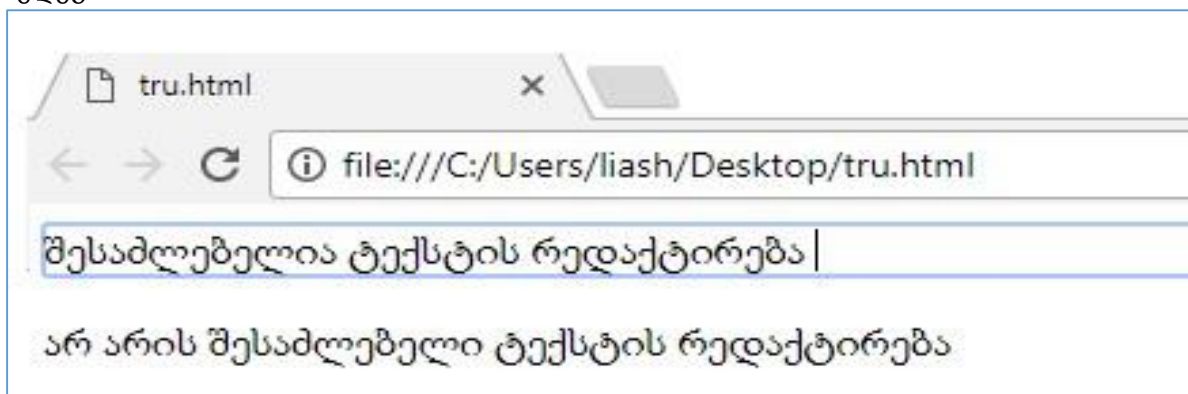
https://gtu.ge/book/L_iashvili_HTML.pdf

</p>

</body>

</html>

შედეგი



true-ს შემთხვევაში მაუსით ორჯერ დაკლიკეთ და თქვენთვის სასურველი ტექსტი ჩასწორდება, თუმცა სკრიპტში ცვლილება არ მოხდება. **contextmenu**-კონტექსტური მენიუ, რომლის მხარდაჭერა ბრაუზერებიდან აქვს მხოლოდ *firefox*-ს, და გამოჩნდება ისიც იმ შემთხვევაში, თუ მაუსის მარჯვენათი დაკლიკებით, მივიღებთ ისეთ შედეგს, რასაც აღწერთ სკრიპტში. ჩვენი მაგალითიდან გამომდინარე, გავუწერეთ სოციალური ქსელები, როგორებიცაა: *Twitter* და *Facebook*, რაც სურათიდან ჩანს, რომ თუ დავაკლიკებთ მარჯვენა მაუსს, გამოჩნდება სურათზე მოცემული ფანჯარა, უნდა აღინიშნოს ისიც, რომ ამ შემთხვევაში ვეხებით *javascript*-ის ფაილებს. სანამ HTML-ის სკრიპტზე გადავიდოდეთ, უნდა განვიხილოთ ისეთი მნიშვნელოვანი ტეგი, როგორიცაა **<menu>** და **<menuitem>**.

<menu> ტეგი განსაზღვრავს მენიუს, სადაც შეიძლება მენიუს ან სხვა რაიმე ბრძანების ჩამატება,

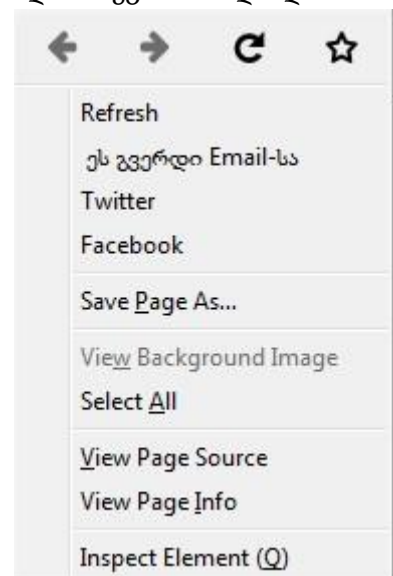
მისი „დამატებითი“ ტეგი არის **<menuitem>**

რომელიც განსაზღვრავს მენიუს ტიპებს და *label* ატრიბუტის საშუალებით მომხარებელს აცოხინებს ინფორმაციას ან რაიმე გარკვეულ ბრძანებას, მისი სინტაქსი ასეთია:

<menuitem label="ტექსტი ">

რაც შეეხება მის ატრიბუტებს გამოიყენება *Label* რომელზეც ვილაპარაკეთ და *type* ანუ იგივე „ტიპი“, რომლის სინტაქსი შესაბამისად გამოყურება, შემდეგი სახით:

<menu type="list ან context ან toolbar">



ლაშა იაშვილი

https://gtu.ge/book/L_iashvili_HTML.pdf

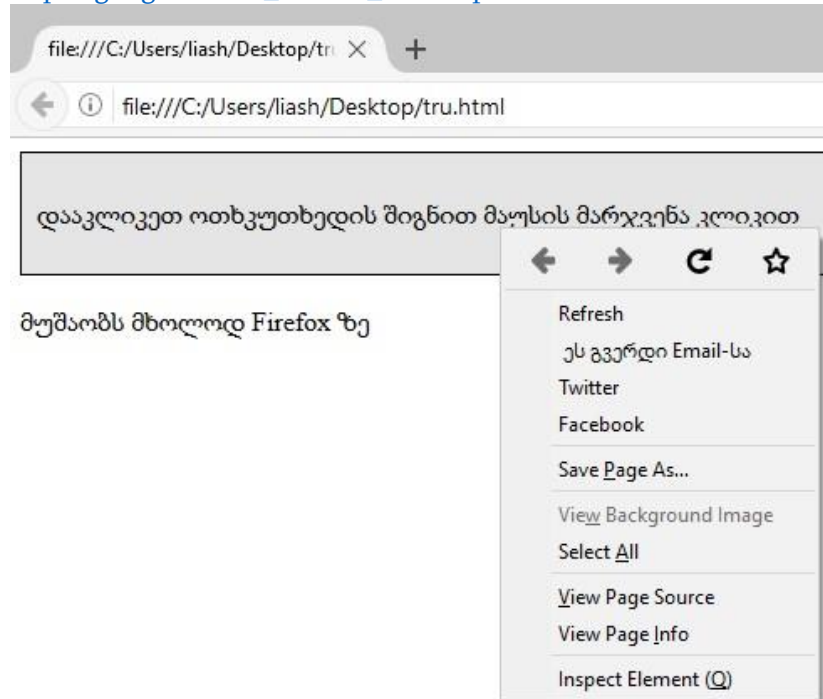
List, context და toolbox საშუალებით შესაძლებელია მენიუში სიის, კონტექსტის toolbar-ების შექმნა. დავუბრუნდეთ contextmenu-ს, რომელის სკრიპტი გამოიყურება შემდეგნაირად: მაგალითი 1.27 <!DOCTYPE html>

```
<html>
<head>
<style>
div
{   background: #e5e5e5;
    border: 1px solid black;
    padding: 9px;
}
</style>
</head>
<body>
<div contextmenu="mymenu">
<p>დააკლიკეთ ოთხკუთხედის შიგნით მაუსის მარჯვენა კლიკით
<menu type="context" id="mymenu">
  <menuitem label="Refresh" onclick="window.location.reload();"
  icon="ico_reload.png">
</menuitem>
  <menuitem label="ეს გვერდი Email-სა"
  onclick="window.location='mailto:?body='+window.location.href;">
</menuitem>
  <menuitem label="Twitter" icon="ico_twitter.png"
  onclick="window.open('/twitter.com/intent/tweet?text=' +
  window.location.href);">
</menuitem>
  <menuitem label="Facebook" icon="ico_facebook.png"
  onclick="window.open('/facebook.com/sharer/sharer.php?u=' +
  window.location.href);">
</menuitem>
</menu>
</div>
<p>მუშაობს მხოლოდ Firefox ზე </p>
</body>
</html>
```

შედეგი

ლაშა იაშვილი

https://gtu.ge/book/L_iashvili_HTML.pdf



რაც შეეხება icon-ებს, მისი მეშვეობით იკონების მითითება შეგვიძლია, რაც შეეხება `onClick="window.open"`-ეს *Event*-ებია, რომელზეც ცოტა ქვემოთ ვიმსჯელოთ, ჯერ ჯერობით ამ ბრძანების მეშვეობით დავიმახსოვროთ, ხოლო დაკლიკებით ახალ ფანჯარაში უნდა გაიხსნას.