ლაშა იაშვილი

enctype ატრიბუტი

HTTP post მეთოდის საშუალებით მონაცემებს აგზავნის სერვერზე, ამ დროს გამოიყენება *enctype* ატრიბუტი, რომელიც ბრაუზერში ახდენს მონაცემების ენკოდირებას. ეს ხდება მანამ, სანამ მონაცემებს გადააგზავნის სერვერზე, მაგრამ ბრაუზერს, როგორც წესი აქვს სამი ენკოდირების მხარდაჭერა:

- *application/x-www-form-urlencoded* სტანდარტულად ბრაუზერი ამ მეთოდს იყენებს ისეთი სიმბოლოები, რომლის დროს არ შეადგენს ალფავიტურ სიმბოლოებს და მხოლოდ მოიცავს ისეთ სიმბოლოებს, როგორიცაა პლუს ნიშანი, გაყოფა ან თუნდაც კლავიატურაზე ე. წ space-ი, Web სერვერზე, სანამ გადაიგზავნება ყველა სიმბოლოს გარდაქმნის „+“ ნიშნად, ხოლო სპეციალური სიმბოლოები, რომელიც შესაძლებელი არის, მათი დეკოდირება გარდაქმნის *ASCI HEX* სიდიდების მიხედვით.
- *multipart/form-data* იძლევა შესაძლებლობას, რომ გადაიგზავნოს მონაცემები ნაწილებად და თითოეული ნაწილი საბოლოო ჯამში წარმოადგენს მთლიან ფორმას, მაგალითისათვის ავიღოდ ფაილების (მაგალითი სურათის) ატვირთვა სერვერზე. ეს ატრიბუტი არ ახდენს სიმბოლოების დეკოდირებას, სერვერისათვის გასაგები უნდა იყოს სერვერზე რა ტიპის მონაცემი იტვირთბა.
- *text/plain* აგზავნის მონაცემებს სერვერზე, როგორც არამოდიფიცირებულს გარდაქმნის რა ცარელ სივრცებს „+“ებად და უცვლელად ტოვებს *ASCI* სიმბოლებს.

სინტაქსი არის:

<form enctype="მეთოდი">

autocomplete ატრიბუტი

ამ ატრიბუტის მითითება გვაძლევს იმის შესაძლებლობას, რომ ველები რომლებიც დამახსოვრებული აქვს სახელი, გვარი, ტელეფონი ან თუნდაც პაროლი ველებში ავტომატურად ჩნდება მისი სინტაქსია:

<input autocomplete="on ან *off">*

უნდა აღინიშნოს, რომ ზოგ ბრაუზერს სჭირდება პარამეტრებიდან ჩართვა, რათა წაითხოს, ამ ატრიბუტი აქვს ორი თვისება *on*, რომელიც სტანდარტულად სულ ჩართულია და გაკვეული პირობებისა თუ ამოცანების გათვალისწინებელით აქვს *off* ფუნქციაც. ეს ატრიბუტი მუშაობს *<input>* ისეთ ტიპებთან, როგორებიც არის: *text, search, url, tel, email, password, datepickers, range,* და *color* ამ ტიპებს ცოტა ქვემოთ აღვწერთ, რაც შეეხება მაგალითს, როგორც ქვემოთ დავინახავთ კლავიატურაზე დაწოლით დამახსოვრებულ ინფორმაციას ამოგვიგდებს.

მაგალითი 1.39

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="utf-8">
</head>
<body>
<form action=" " method="get" autocomplete="on">
სახელი:<input type="text" name="saxeli"><br>
  E-mail: <input type="email" name="email"><br>
<input type="submit" name="დასტური">
</form>
</body>
</html>
```

ტექტის შეტანა ველებში(*textbox)* შეგვიძლია ნებისმიერ ვებგვერდებზე. ყველა ველს აქვს თავის დანიშულება და შესაბამისად გვპასუხობს იმ მოთხოვნაზე, რასაც ვავსებთ, საერთოდ არსებობს, კონტროლერი ფორმის შევსების სამი ტიპი:

- *Single-line* კონტროლეერი-ერთ სტრიქონიან ჩასაწერი ველი, ეს შემთხვევა გამოყენება როგორც მარტო <input> ელემენტთან ისე *textbox*-ებთან.
- *Password* კონტროლერი-ძალიან ჰგავს ერთ სტრიქონანი შეტანის ველს. განსხვავება მხოლოდ ის არის, რომ ტექსტს რომ შევიტანთ გამოჩნდება, ფიფქებით ან შავი წერილებით. ამ კონტროლერს ძირითად გამოყენებენ საიტზე ავტორიზაციისას, როდესაც სახელს და პაროლს მიუთითებენ.

მაგალითი

1.40

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<form action="">
სახელი:<br>
<input type="text" name="სახელი">
<br>
პაროლი:<b
r>
<input type="password" name="პაროლი">
</form>
</body>
</html>
```

შედეგი

Multiline-მრავალ სტრიქონიანი კონტროლერი არის ის სადაც შეგვიძლია ერთ სტრიქონზე მეტი ჩაწერა, ამისთვის გამოიყენებენ *<textarea>* ელემენტს. HTML 5 აქვს დამატებული ბევრი ახალი *<input>* ტიპი, რომელიც ზოგადად აღწერენ ინფორმაციას Web გვერდზე:

- ***color:*** ფერების არჩევა ;
- ***data***: კალენდარული თარიღის შეტანა;
- ***datatime:*** დროისა და რიცხვის შეტანა *Greenwich*-ის დროით;
- ***datatime-local:*** ადგილობრივი დრო და თარიღი;
- **email:** ელ-ფოსტის მითითება, შესაძლებელია როგორც ერთ სტრიქონიანი, ისე მრავალ სტრიქონზე მითითებით;
- ***month:*** თვისა და წლის მითითება;
- ***number:*** რიცხვის შეტანა;
- ***range:***შეტანილი ინფორმაციის გადადგილება როგორც სლაიდად;
- ***search:*** ძებნა;
- ***tel:*** ტელეფონის რიცხვი;
- ***time:*** საათი, წუთების და წამების აღწერა;
- ***url:*** ვებსაიტის მისამართი;
- ***week:*** ასახავს არსებულ კვირას.

მაგალითი 1.41

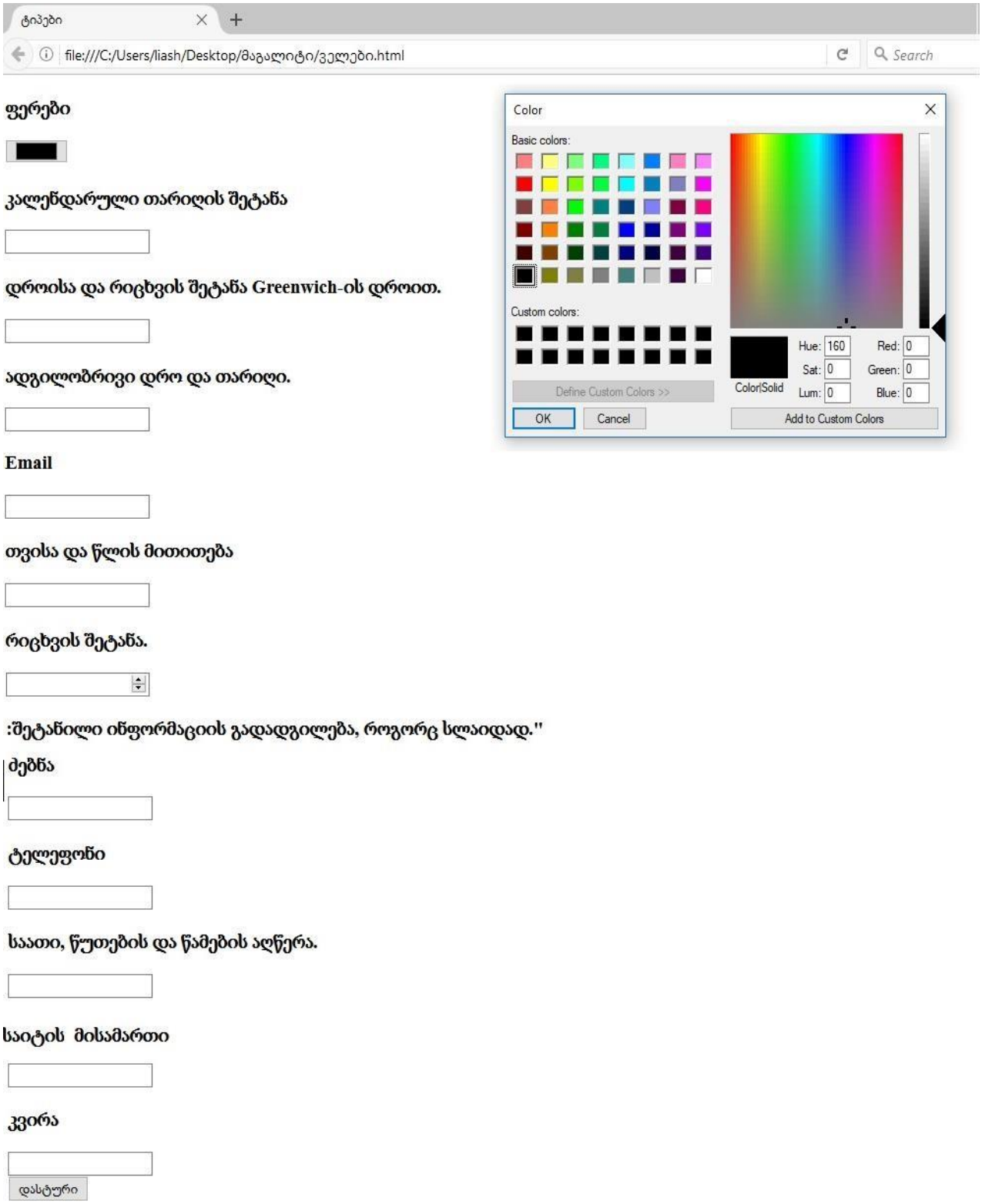
```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> ტიპები </title>
</head>
<body>
<div>
```

```
 <h3> ფერები </h3>
    <input type="color" name="color">
</div>
<div>
 <h3> კალენდარული თარიღის შეტანა </h3>
    <input type="date" name="date">
</div>
<div>
    <h3>  დროისა და რიცხვის შეტანა Greenwich-ის დროით.  </h3>
    <input type="datetime" name="datetime">
</div>

<div>
    <h3 >ადგილობრივი დრო და თარიღი. </h3>
    <input type="datetime-local" name="datetime-local">
</div>
<div>
    <h3> Email </h3>
    <input type="email" name="email">
</div>
<div>
    <h3> თვისა და წლის მითითება </h3>
    <input type="month" name="month">
</div>
<div>
    <h3>რიცხვის შეტანა.</h3>
    <input type="number" name="number">
</div>
<div>
    <h3>:შეტანილი ინფორმაციის გადადგილება, როგორც სლაიდად.</h3>
    <input type="range" id="range" name="range">
    <output for="range" id="output"></output>
</div>
<div>
    <h3> ძებნა </h3>
    <input     type="search"     name="search"     results="5"
autosave="savedsearches"> </div>
<div>
    <h3>ტელეფონი </h3>
    <input type="tel" name="tel">
</div>
<div>
    <h3>საათი, წუთების და წამების აღწერა.</h3>
    <input type="time" name="time">
</div>
```

```
<div>
    <h3> საიტის მისამართი საიტის </h3>
    <input type="url" name="url">
</div>
<div>
    <h3>კვირა</h3>
    <input type="week" name="week">
</div>
<div>
    <input type="submit" value="დასტური">
</div>
 </body>
</html>
```

შედეგი



<input> ელემენტი, როდესაც იყენებს type ატრიბუტს და მინიჭებული აქვს text. იგი შეგვიძლია ატრიბუტი შეუძლია განავრცო და გამოიყენო სხვა ატრიბუტებშიც როგორიცაა:

- ყველა უნივერსალური ატრიბუტი;
- *disabled, readonly, form, autocomplete, autofocus, list, pattern, required* და *dirname*.

ცხრილში მოცემულია სტანდარტული ტექსტის შესაყვანი ატრიბუტები.

ატრიბუტი	აღწერა
name	სახელობითი ატრიბუტი, რომელისაც ენიჭება *name/value* წყვილი იგზავნება სერვერზე. დავიმახსოვროთ, რომ თითოეული *name/value* წყვილის წარმოდგენა არის ფორმის კონტროლისათვის, სადაც *name*სახელი ინდეფიცირებას ახდენს ფორმის კონტროლერის და *value* არის, რასაც მომხარებელი შეიყვანს.
value	სიდიდე ან ცვლადი, რომლის საშუალებით ფორმა იტვირთება
size	სიმბოლოების შეყვანის რაოდენობა, რომელიც განსაზღვრავს, თუ რამდენი სიმბოლო უნდა შეიყვანოს მომხმარებელმა, თუ მითითებულია 10 სიმბოლო, ხოლო მომხმარებელმა შეიყვანა 10 ზე მეტი, მას აღარ ჩაწერს.

მრავალ სტრიქონიანი ველი იქმნება *<textarea>* ტეგის მეშვეობით, რომელსაც აქვს დამხურავი ტეგი *</textarea>,* ამ ტეგში შეგვიძლია განუსაზღვრელი სხვადასხვა სიმბოლოს აკრეფა. მისი ტექსტის არეალს განსაზღვრავს მისივე *col* და *row,* რომელიც სტანდარტულია, თუ რა ზომის გვინდა იყოს ტექსტის არეალი მაგალითად: ავიღოთ *col=70*, ხოლო *row=50*, მაშინ გვექნება ასეთი სახე

მაგალითი 1.42

```
<!DOCTYPE html>
<html>
<head>
<title> სტრიქონი </title>
<meta charset="utf-8">
</head>
<body>
<textarea name="textarea" cols="30" rows="10">
მრავალ სტრიქონიანი ველი
</textarea>
```

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

შედეგი

ვხედავთ, რომ ტექსტის არეალი სტანდარტულად როგორც ჩანს, საერთოდ ტექსტის ფორმის საკითხები CSS მეშვეობით წყდება. ცხრილში მოცემულია *<textarea>* ტეგის დამატებითი ატრიბუტები:

ატრიბუტი	აღწერა (გამოყენება **input** ტეგშიც)
autofocus	ლოგიკური ატრიბუტი, რომელიც მიუთითებს, რომ ელემენტი უნდა იყოს ფოკუსირებული, როცა გვერდი იტვირთება
cols	აღწერს *textarea* სიგანეს
disabled	მიუთითებს, რომ მოცემული *textarea* უნდა დაიმალოს
form	<form> ელემენტი აღწერ <textarea>-ს და ურთიერთქმედეს id საშუალებით
maxlength	მაქსიმალური რაოდენობა სიტყვების
placeholder	მოკლედ აღწერს ტექსტური ველის დანიშნულებას
readonly	ტექსტური ველი, სადაც კითხვით მხოლოდ რეჟიმია
required	ველი, რომელიც აუცილებად უნდა შეივსოს
rows	ტექსტური ველის რიგი, აღწერს სიგრძეს
wrap	ტექსტური ველი, როგორ უნდა იქნეს ჩაკეცილი/ჩახვეული

განვიხილოთ თითეული ატრიუტები და მათი სინტაქსი. ისეთი ატრიბუტი როგორიც არის *name, value, size, col* და *row* აღარ ვლაპარაკობთ, რადგან არა ერთხელ შევხდით ამ ატრიბუტებს:

autofocus

HTML 5 ში ჩაემატა ახალი ატრიბუტი, მისი მოვალეობა არის, რომ ფოკუსირეული იყოს, თუ როდის მოხვდება გვერდების ჩატვირთვა. მაგალითად ავიღოთ ფორმა, სადაც აღწერილია, სახელი, პაროლი ელ-ფოსტა, მივუთითოთ, რომ, როდესაც გვერდს ჩატვირთავს (გახსნის) ავტომატურად მაუსის კურსორი ელ-ფოსტაზე იდგეს.

სინტაქსი:

```
<textarea autofocus>
```

ასევე ატრიბუტის გამოყენება ხდება *<input>* ტეგშიც

სინტაქსი:

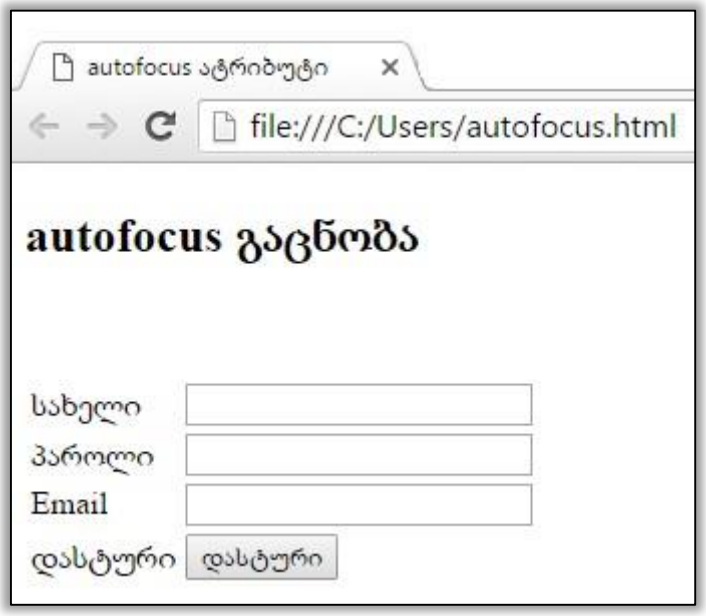
```
<input autofocus>
```

მაგალითი 1.43

```
<!DOCTYPE html>
<html lang="ka">
 <head>
   <meta charset="utf-8">
   <title>autofocus ატრიბუტი </title>
 </head>
 <body>
   <h2>autofocus გაცნობა</h2>
 <form method="get">
<table>
<tr>
<td>სახელი </td>
<td>
<input type="text" Name="სახელი"></td>
</tr>
<br>
<tr><td>პაროლი  </td>
<td><input type="password" name="პაროლი" >
</td></tr>
<br>
<tr><td>Email </td>
<td>
<input type="email" name="ელ-ფოტა" autofocus>
</td>
</tr>
```

```
<tr>
<td>დასტური  </td>
<td><input type="submit" value="დასტური">
</td></tr>
</table>
</form>
</body>
</html>
```

შედეგი



disabled

ველი რომელიც არ გვინდა იყოს აქტიური, შეგვიძლია გამოვიყენოთ *disabled* ატრიბუტი, ამ ატრიბუტს იყენებს იმ შემთხვევაში, თუ წინ რაიმე გარკვეული პირობა აქვს, მაგალითად რაიმე უნდა მოინიშნოს და შემდეგ გააქტიურდეს, მაგრამ ეს ყველაფერი *JavaScript* ურთერთობისას იქნება, ხოლო სტატიკური გვერდებისას მხოლოდ გაუქმებულად გამოჩნდება.

მისი სინტაქსი არის:

<textarea disabled> ან *<input disabled>*

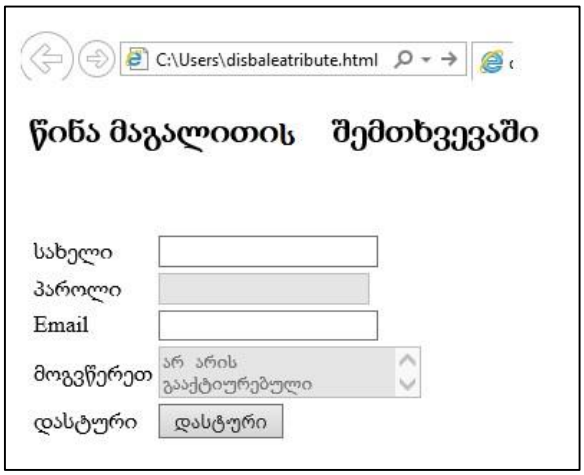
წინა მაგალითში ჩავამატოთ *<textarea>* და პაროლის ველი გავაუქმოთ.

მაგალითი 1.44

```
<!DOCTYPE html>
<html lang="ka">
```

```
 <head>
   <meta charset="utf-8">
   <title>disabled მაგალითი </title>
 </head>
 <body>
   <h2>წინა მაგალითი შემთხვევაში</h2>
 <form method="get">
<table>
<tr>
<td>სახელი </td><td><input type="text" name="სახელი"></td>
</tr><br>
<tr>
<td>პაროლი  </td><td><input type="password" name="პაროლი" disabled
>
</td></tr>
<br>
<tr><td>Email </td><td><input type="email" name="ელ-ფოტა"
autofocus>
</td>
</tr>
<tr>
<td> მოგვწერეთ </td><td><textarea disabled>არ არის
გააქტიურებული</textarea>
</td>
</tr>
<tr><td>დასტური  </td><td><input type="submit"
value="დასტური"></td></tr>
</table>
</form>
</body>
</html>
```

შედეგი



form

ფორმის განსაზღვრა ხდება id საშუალებით, ანუ *form*-შიც რა *id*-საც მიანიჭებს ის, უნდა გამოცხადეს *<textarea>*-ში, ფორმის გარეთ მდებარეობს, მაგრამ მას ეკუთვნის.

სინტაქსი:

<textarea form="ფორმის_id">

შევქმნათ *textbox* და *<textarea>*, სადაც აღწერილი იქნება ფორმის *id* ელემენტი მაგალითი **1.45** *<!doctype html>*

```
<html>
<head>
<meta charset="utf-8">
<title> ფორმა</title>
</head>
<body>
<form action="" id="ფორმა">
სახელი: <input type="text"
name="სახელი">
  <input type="submit" value="დასტური">
</form>
<br>
<textarea rows="14" cols="50" name="saxeli" form="ფორმა">
შევიყვანოთ ტექსტი
</textarea>
</body>
</html>
```


ფორმაატრიბუტის.html
file:///C:/Users/ფორმაატრიბუტის.html
სახელი:
დასტური
შევიყვანოთ ტექსტი