**ლექცია №3**

**CSS - საზღვრები, ბლოკები**

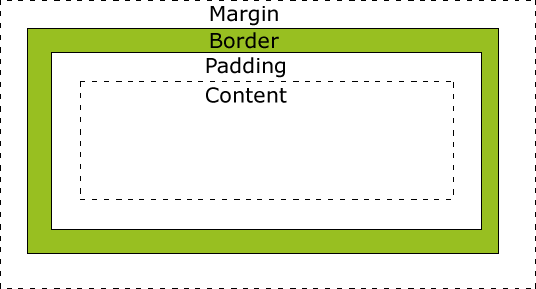
**CSS Box მოდელი**

ყველა HTML ელემენტები შეიძლება ჩაითვალოს, ან განხილულ იქნეს როგორც **ბლოკი** (box - ყუთი). CSS -ში, ტერმინი "ბლოკერი (box) - მოდელი" გამოიყენება, როდესაც ვსაუბრობთ ვებ გვერდის დიზაინზე, პროექტირებასა და მისი ნაწილების განლაგების მაკეტზე.

CSS ბლოკური- მოდელი არსებითად არის ყუთი, რომელითაც გარშემორტყმულია HTML ელემენტები, და იგი შედგება: **კიდის** (Margin - ნაპირი, ველი), **ჩარჩოს** (Border), **დაშორების** - შრე (Padding) და **შიგთავსის** -გან(Content).

ბლოკური მოდელი საშუალებას გვაძლევს მოვაქციოთ საზღვრებში ელემენტები და სივრცითი ელემენტები, გამოვყოთ ისინი სხვა ელემენტებისგან.

ქვემოთ მოყვანილი გამოსახულება ასახავს ბლოკურ მოდელს:



განვმარტოთ ეს ნაწილები:

* **კიდე -** Margin - თავისუფალი არე ჩარჩოს გარშემო. ნაპირს -კიდეს არ აქვს ფონის ფერი, ის არის სრულიად გამჭვირვალე
* **ჩარჩო**  - Border -ჩარჩოს რომლითაც გარშემოტყმულია დაშორება - შრე და შიგთავსი. ჩარჩო მემკვიდრეობით იღებს ბლოკის ფერს
* **დაშორება - შრე** - თავისუფალი არე შიგთავსის გარშემო. დაშორებაზე ზეგავლენას ახდენს ბლოკის ფონური ფერი
* **შიგთავსი** - ბლოკის ის ნაწილი, რომელშიც განთავსდება ტექსტი და გამოსახულება - სურათები

იმისათვის, რომ ელემენტის მითითებული სიგანის და სიმაღლის ერთობლიობა ყველა ბრაუზერმა სწორად აღიქვას - წაიკითხოს, უნდა ვიცოდეთ, თუ როგორ მუშაობს ბლოკური მოდელი.

## ელემენტის სიგანე და სიმაღლე

## მივაქციოთ ყურადღება: როცა თქვენ მიუთითებთ ელემენტის სიგანისა და სიმაღლის თვისებებს CSS-ის საშუალებით, ამით თქვენ ფაქტიურად უთითებთ შიგთავსის სიგანესა და სიმაღლეს.

## ელემენტის სრული ზომის გამოსათვლელად, თქვენ უნდა დაამატოთ დაშორებების, ჩარჩოს და კიდეების ზომები.

|  |
| --- |
|  |

ელემენტის საერთო სიგანე ქვემოთ მოყვანილ მაგალითში არის 300px:

width: 250px;  
padding:10px;  
border: 5px solid gray;  
margin: 10px;

მოდით გამოვთვალოთ   
250px (სიგანე)   
+ 20px (მარცხენა + მარჯვენა დაშორება , თითო 10 px)   
+ 10px (მარცხენა + მარჯვენა ჩარჩო, თითო 5 px)   
+ 20px (მარცხენა + მარჯვენა ზღვარი, თითო 10 px)   
= 300px

ვივარაუდოთ, რომ თქვენ გაქვთ სივრცეს მხოლოდ ეს -250px, ზომა. მოდით შევქმნათ ელემენტი საერთო სიგანით - 250px:

## მაგალითად

width:220px;  
padding:10px;  
border:5px solid gray;  
margin:0px;

შევადგინოთ **ლისტინგი 1**, ამის საილუსტრაციოდ:

<!DOCTYPE html>

<html>

<head>

<style>

div.ex

{

width:220px;

padding:10px;

border:5px solid gray;

margin:0px;

}

</style>

</head>

<body>

<img src="w3css.gif" width="250" height="250">

<div class="ex">ზემოთ მოყვანილ სურათს ზომა სიგანეში - 250px -ია.

ამ ელემენტის სიგანის ერთიანი ზომაც ასევე 250px -ია.</div>

</body>

</html>

*ელემენტის საერთო სიგანე უნდა გამოითვალოს ასე:*

ელემენტს სიგანე სულ = მითითებული სიგანე (შიგთავსის) + მარცხენა დაშორების სიგანე + მარჯვენა დაშორების სიგანე + ჩარჩოს მარცხენა მხარის სიგანე + ჩარჩოს მარჯვენა მხარის სიგანე+ მარცხენა კოდის სიგანე + მარჯვენა კოდის სიგანე

*ელემენტის საერთო სიმაღლე უნდა გამოითვალოს ასე:*

სულ ელემენტს სიმაღლე = მითითებული სიმაღლე (შიგთავსის) + ზედა დაშორების სიმაღლე + ქვედა დაშორების სიმაღლე + ჩარჩოს ზედა მხარისსიმაღლე + ჩარჩოს ქვედა მხარის სიმაღლე + ზედა კიდის სიმაღლე + ქვედა კიდის სიმაღლე

**ბრაუზერები თავსებადობა ამ ნორმასთან**

IE8 და IE-ის ადრინდელ ვერსიებში, დაშორებისა და ჩარჩოს სიგანე ჩართულია ელემენტის მითითებულ სიგანეში - ანუ შიგთავსისი სიგანე შემცირდება. ამ პრობლემის გადასაწყვეტად დავამატოთ <! DOCTYPE html> ტეგი HTML გვერდს.

**CSS-ის საზღვრები**

CSS-ის საზღვრების თვისებები გვაძლევენ საშუალებას განვსაზღვროთ ელემენტის ჩარჩოს სტილი და ფერი

**ჩარჩოს სტილი:**

border-style -ის თვისება გვაძლევს საშუალებას ავირჩიოთ ჩარჩოს ისა თუ ეს სახე.

შევნიშნით, რონ ჩარჩოს არცერთი თვისება არ იმუშავებს, თუ კი არ ენება მითითებული border-style -ს თვისებები.

border-style -ის თვისების შესაძლო მნიშვნელობები;

border-style:

none - ჩარჩო არაა.

dotted - წერტილოვანი ჩარჩო.

dashed - ჩარჩო პუნქტირის ხაზებით.

solid - ჩარჩო უწყვეტი ხაზით.

double - ორმაგი ხაზის ჩარჩო.

groove - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, ღარის ეფექტით.

ridge - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, კეხის ეფექტით.

inset - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, შიგთავსის გვერდის სიბრტყეში ჩაწევის ეფექტით.

outset - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, შიგთავსის გვერდის სიბრტყიდან ამოწევის ეფექტით.

hidden - დამალული ჩარჩო.

ლისტინგი 2. ჩარჩოს სტილი:

<!DOCTYPE html>

<html>

<head>

<style>

p.none {border-style:none;}

p.dotted {border-style:dotted;}

p.dashed {border-style:dashed;}

p.solid {border-style:solid;}

p.double {border-style:double;}

p.groove {border-style:groove;}

p.ridge {border-style:ridge;}

p.inset {border-style:inset;}

p.outset {border-style:outset;}

p.hidden {border-style:hidden;}

</style>

</head>

<body>

<p class="none">ჩარჩო არაა.</p>

<p class="dotted"> dotted - წერტილოვანი ჩარჩო.</p>

<p class="dashed"> dashed - ჩარჩო პუნქტირის ხაზებით.</p>

<p class="solid"> solid - ჩარჩო უწყვეტი ხაზით.</p>

<p class="double">double - ორმაგი ხაზის ჩარჩო.</p>

<p class="groove">groove - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, ღარის ეფექტით.</p>

<p class="ridge">ridge - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, კეხის ეფექტით.</p>

<p class="inset">inset - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, შიგთავსის გვერდის სიბრტყეში ჩაწევის ეფექტით.</p>

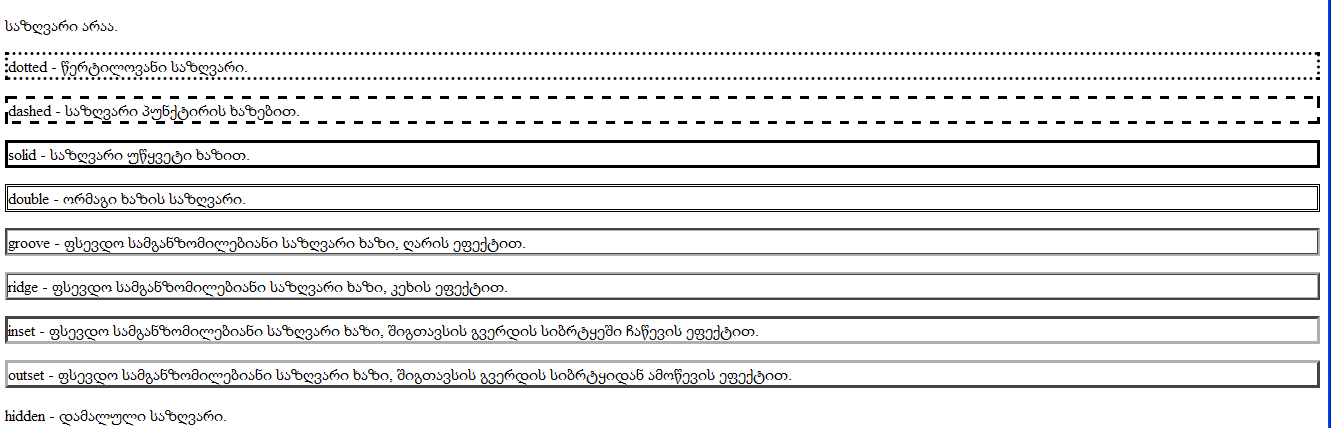
<p class="outset">outset - ფსევდო სამგანზომილებიანი ჩარჩო ხაზი, შიგთავსის გვერდის სიბრტყიდან ამოწევის ეფექტით.</p>

<p class="hidden">hidden - დამალული ჩარჩო.</p>

</body>

</html>

ამ კოდის - ლისტინგ 2-ის რეალიზაციით მივიღებთ შემდეგ სურათს:



ნახატი 1. ჩარჩოს სტილი

**ჩარჩოს სისქე**

**border-width** თვისება გამოიყენება ჩარჩოს სისქის მნიშვნელობების მინიჭებით. სისქის მნიშვნელობები შეიძლება მოცემული იქნენ პიქსელებში ან სიტყვიერად, ერთ ერთი ამათგან: **thin** (წვრილი), **medium** (საშუალო), ან **thick** (სქელი).

ლისტინგი 3. ჩარჩოს სისქე

<!DOCTYPE html>

<html>

<head>

<style>

p.one

{

width:220px;

padding:10px;

margin:30px;

border-style:solid;

border-width:5px;

}

p.two

{

width:220px;

padding:10px;

margin:30px;

border-style:solid;

border-width:medium;

}

p.three

{

width:220px;

padding:10px;

margin:30px;

border-style:solid;

border-width:1px;

}

</style>

</head>

<body>

<p class="one">რაიმე ტექსტი.</p>

<p class="two">რაიმე ტექსტი.</p>

<p class="three">რაიმე ტექსტი.</p>

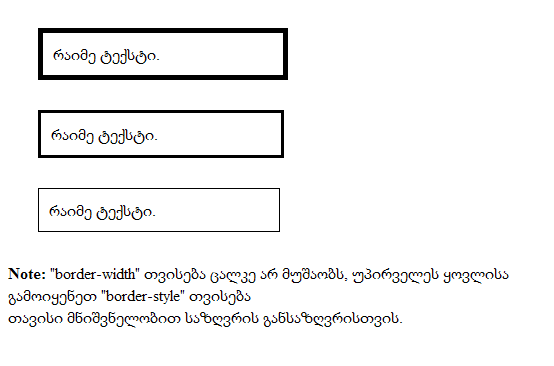
<p><b>Note:</b> "border-width" თვისება ცალკე არ მუშაობს, უპირველეს ყოვლისა <br>

გამოიყენეთ "border-style" თვისება <br>თავისი მნიშვნელობით ჩარჩოს განჩარჩოსთვის.</p>

</body>

</html>

შედეგად მივიღებთ



ნახატი 2. ჩარჩოს სისქე

**ჩარჩოს ფერი**

border-color თვისებით ხდება ჩარჩოს ფერის განსაზღვრა. ფერები შესაძლებელია მოცემულ იქნენ შემდეგნაირად:

* სახელით, როდესაც მიეთითება ფერის სიტყვიერი დასახელება, მაგალითად “red”
* RGB კოდით, მაგალითად ”rgb(255,0,0)”
* თექვსმეტობითი კოდით “#ff0000” ან“#f00”

ასევე შესაძლებელიაborder-color თვისებას ჰქონდეს მნიშვნელობა ”transparent” - გამჭირვალე.

შევნიშნოთ - border-color თვისება ცალკე არ მუშაობს, უპირველეს ყოვლისა გამო­იყენეთ "border-style" თვისება თავისი მნიშვნელობით ჩარჩოს განსაზღვრისთვის.

ლისტინგი 4. ჩარჩოს ფერი

<!DOCTYPE html>

<html>

<head>

<style>

p.one

{

width:220px;

padding:10px;

margin:30px;

border-style:solid;

border-color:red;

}

p.two

{

width:220px;

padding:10px;

margin:30px;

border-style:solid;

border-color:#98bf21;

}

</style>

</head>

<body>

<p class="one"> უწყვეტი წითელი ჩარჩო</p>

<p class="two">უწყვეტი მწვანე ჩარჩო</p>

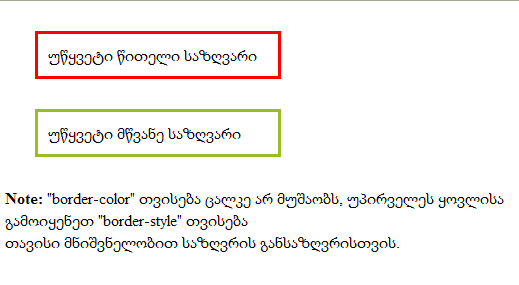
<p><b>Note:</b> "border-color" თვისება ცალკე არ მუშაობს, უპირველეს ყოვლისა <br>

გამოიყენეთ "border-style" თვისება <br>თავისი მნიშვნელობით ჩარჩოს განჩარჩოსთვის.</p>

</body>

</html>

ლისტინგ 4 -ის რეალიზაციით მივიღებთ

****

ნახატი 4. ჩარჩოს ფერები

**ჩარჩოს ცალკეული მხარეები**

CSS -ში შესაძლებელია ჩარჩოს სხვადასხვა მხარის განუსაზღვროთ სხვადასხვა სახე:

მაგალითად **ლისტინგი 5** ჩარჩოს 2-2 მხარის განსხვავებული სტილი და ფერი

<!DOCTYPE html>

<html>

<head>

<style>

p

{width:220px;

padding:10px;

margin:30px;

border-top-style:dotted;

border-right-style:solid;

border-bottom-style:dotted;

border-left-style:solid;

border-top-color:rgb(255,0,0);

border-bottom-color:rgb(255,0,0);

border-left-color:green;

border-right-color:green;

}

</style>

</head>

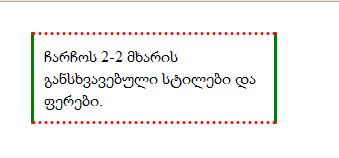
<body>

<p> ჩარჩოს 2-2 მხარის განსხვავებული სტილი და ფერი.</p>

</body>

</html>

ლისტინგი 5-ის რეალიზაციის შემდეგ



ნახარი 5. ჩარჩოს მხარეების სხვადასხვა სტილი და ფერი

მოყვანილი მაგალითისთვის შეიძლებოდა გამარტივებული ჩანაწერის (კოდის) შექმნა, როცა ჩარჩოს ზედა და გვერდითი მხარეები ჯგუფდება და მათ ერთი სტილი, თუ ფერი განესაზღვრება, ამდენად ზემოთ მოყვანილო მაგალითის კოდი შეიძლ;ება უფრო მარტივად ჩავწეროთ და მან მაინც მოგვცეს ნახატ 5-ზე გამოსახული სურათის მნიშვნელობა:

ლიტინგი 6 ჩარჩოს 2-2 მხარის განსხვავებული სტილი და ფერი

<!DOCTYPE html>

<html>

<head>

<style>

p

{width:220px;

padding:10px;

margin:30px;

border-style:dotted solid;

border-color:rgb(255,0,0) green;

}

</style>

</head>

<body>

<p> ჩარჩოს 2-2 მხარის განსხვავებული სტილი და ფერი.</p>

</body>

</html>

border-style -ს შეიძლება ჰქონდეს ერთ-ერთი ასეთი სახის მნიშვნელობები

* **border-style:dotted solid double dashed;** 
  + ჩარჩოს ზედა მხარე არის წერტილოვანი
  + ჩარჩოს მარჯვენა მხარე არის უწყვეტი
  + ჩარჩოს ქვედა მხარე არის ორმაგი
  + ჩარჩოს მარცხენა მხარე არის პუნქტირული
* **border-style:dotted solid double;**
  + ჩარჩოს ზედა მხარე არის წერტილოვანი
  + ჩარჩოს მარჯვენა და მარცხენა მხარები არიან უწყვეტი
  + ჩარჩოს ქვედა მხარე არის ორმაგი
* **border-style:dotted solid;**
  + ჩარჩოს ზედა და ქვედა მხარეები არიან წერტილოვანნი
  + ჩარჩოს მარჯვენა და მარცხენა მხარები არიან უწყვეტი
* **border-style:dotted;**
  + ჩარჩოს ოთხივე მხარე არის წერტილოვანი

ზემოთ მოყვანილი მაგალითის არსი, განხილული border-style თვისებისთვის სამართლიანია ასევე border-width და border-color თვისებებისთვის. განიხილეთ ისინი დამოუკიდებლად.

**ჩარჩო - სტენოგრაფიული (შემოკლებული) თვისებები**

როგორც ხედავთ, ზემოთ მოყვანილი მაგალითებიდან, არსებობს მრავალი თვისებები, რომლების განიხილება ჩარჩოს მიმართ.

კოდიი შემცირება შესაძლებელია თუ კი ჩარჩოს ყველა ინდივიდუალურ თვისებას გავაერთიანებთ ერთ თვისებაში. ამას ეწოდება სტენოგრაფიული (შემოკლებული) თვისება.

border თვისება წარნოადგენს ჩარჩოს შემდეგი ცალკეული ჱბისებების შემოკლება - სტენოგრაფირებას:

* border-width
* border-style (აიცილებელი)
* border-color

მაგალითად, ლისტინგი 7 სტენოგრაფიული თვისება

<!DOCTYPE html>

<html>

<head>

<style>

p

{

width:220px;

padding:10px;

margin:30px;

border:5px solid red;

}

</style>

</head>

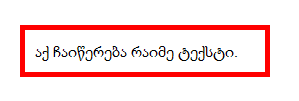
<body>

<p>აქ ჩაიწერება რაიმე ტექსტი.</p>

</body>

</html>

ლისტინგი 7-ის რეალიზაციით გვაქვს



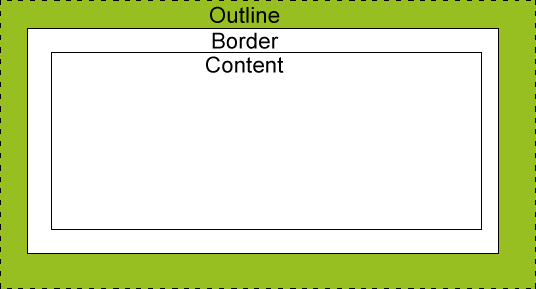
ნახატი 7. სტენოგრაფიული თვისება

|  |  |
| --- | --- |
| **თვისება** | **აღწერა** |
| [border](http://www.w3schools.com/cssref/pr_border.asp) | border -ს ყველა თვისების ერთი დეკლარაციით განსაზღვრა |
| [border-bottom](http://www.w3schools.com/cssref/pr_border-bottom.asp) | bottom border -ს ყველა თვისების ერთი დეკლარაციით განსაზღვრა |
| [border-bottom-color](http://www.w3schools.com/cssref/pr_border-bottom_color.asp) | ჩარჩოს ქვედა მხარისთვის ფერის განსაზღვრა |
| [border-bottom-style](http://www.w3schools.com/cssref/pr_border-bottom_style.asp) | ჩარჩოს ქვედა მხარისთვის სტილის განსაზღვრა |
| [border-bottom-width](http://www.w3schools.com/cssref/pr_border-bottom_width.asp) | ჩარჩოს ქვედა მხარისთვის სისქის განსაზღვრა |
| [border-color](http://www.w3schools.com/cssref/pr_border-color.asp) | ჩარჩოს ოთხივე მხარისთვის ფერის განსაზღვრა |
| [border-left](http://www.w3schools.com/cssref/pr_border-left.asp) | ჩარჩოს მარცხენა მხარისთვის ყველა თვისებების ერთი დეკლარაციით განსაზღვრა |
| [border-left-color](http://www.w3schools.com/cssref/pr_border-left_color.asp) | ჩარჩოს მარცხენა მხარისთვის ფერის განსაზღვრა |
| [border-left-style](http://www.w3schools.com/cssref/pr_border-left_style.asp) | ჩარჩოს მარცხენა მხარისთვის სტილის განსაზღვრა |
| [border-left-width](http://www.w3schools.com/cssref/pr_border-left_width.asp) | ჩარჩოს მარცხენა მხარისთვის სისქის განსაზღვრა |
| [border-right](http://www.w3schools.com/cssref/pr_border-right.asp) | ჩარჩოს მარჯვენა მხარისთვის ყველა თვისებების ერთი დეკლარაციით განსაზღვრა |
| [border-right-color](http://www.w3schools.com/cssref/pr_border-right_color.asp) | ჩარჩოს მარჯვენა მხარისთვის ფერის განსაზღვრა |
| [border-right-style](http://www.w3schools.com/cssref/pr_border-right_style.asp) | ჩარჩოს მარჯვენა მხარისთვის სტილის განსაზღვრა |
| [border-right-width](http://www.w3schools.com/cssref/pr_border-right_width.asp) | ჩარჩოს მარჯვენა მხარისთვის სისქის განსაზღვრა |
| [border-style](http://www.w3schools.com/cssref/pr_border-style.asp) | ჩარჩოს ოთხივე მხარისთვის სტილის განსაზღვრა |
| [border-top](http://www.w3schools.com/cssref/pr_border-top.asp) | ჩარჩოს ზედა მხარისთვის ყველა თვისებების ერთი დეკლარაციით განსაზღვრა |
| [border-top-color](http://www.w3schools.com/cssref/pr_border-top_color.asp) | ჩარჩოს ზედა მხარისთვის ფერის განსაზღვრა |
| [border-top-style](http://www.w3schools.com/cssref/pr_border-top_style.asp) | ჩარჩოს ზედა მხარისთვის სტილის განსაზღვრა |
| [border-top-width](http://www.w3schools.com/cssref/pr_border-top_width.asp) | ჩარჩოს ზედა მხარისთვის სისქის განსაზღვრა |
| [border-width](http://www.w3schools.com/cssref/pr_border-width.asp) | ჩარჩოს ოთხივე მხარისთვის სისქის განსაზღვრა |

**CSS კონტური**

კონტური არის ხაზი, რომელიც ელემენტის გარშემო შემოიხაზება (ჩარჩოს გარეთ) და გამოყოფს ამ ელემენტას.

outline -ს აქვს კონტურის style, color და width -ის თვისებები



|  |  |  |
| --- | --- | --- |
| **თვისება** | **აღწერა** | **მნიშვნელობა** |
| [outline](http://www.w3schools.com/cssref/pr_outline.asp) | outline -ს ყველა თვისების ერთი დეკლარაციით განსაზღვრა | *outline-color outline-style outline-width* inherit (მემკვიდრეობითობა) |
| [outline-color](http://www.w3schools.com/cssref/pr_outline-color.asp) | კონტურისთვის ფერის მინიჭება | *color\_name hex\_number rgb\_number* invert (კონტრასტული) inherit |
| [outline-style](http://www.w3schools.com/cssref/pr_outline-style.asp) | კონტურისთვის სტილის მინიჭება | none dotted dashed solid double groove ridge inset outset inherit |
| [outline-width](http://www.w3schools.com/cssref/pr_outline-width.asp) | კონტურისთვის სისქის მინიჭება | thin medium thick *length* inherit |

**CSS -პოზიცირება**

CSS -ში პოზიცირების თვისება გვაძლევს საშუალებას ელემენტის მდებარეობის განსაზღვრისა ვებ-გვერდზე . ის ასევე იძლევა საშუალებას, ერთი ელემენტი მეორის უკან განვათავსოთ და როგორ უნდა მოვიქცეთ, როცა ელემენტის შიგთავსი არის ძალიან დიდი.

ელემენტის პოზიცირება შესაძლებელია top, bottom, left და right თვისებების საშუ­ალებით, მაგრამ ეს თვისებები არ მუშაობენ თუ კი პირველად არ არის პოზიცირების- position თვისების მინიჭება. ისინი ასევე სხვადასხვანაირად მოქმედებენ პოზიცირების სხვადასხვა მეთოდისთვის.

სულ არის პოზიცირების ოთხი მეთოდი, რომელიც **position** თვისებისთვის 4 სხვადასხვა მნიშვნელობის მინიჭებით მიიღებიან:

**static**  - პოზიცირება

static - პოზიცირებაეს არის ჩუმათობის პრინციპით განსაზღვრული, ელემენტები ჩვეულებრივად განთავსდებიან, ნორმალური ნაკადის შესაბამისად. top, bottom, left და right თვისებები არ გამოიყენება და არც **position** თვისება ჩაიწერება.

ელემენტების ნორმალური ნაკადი წარმოადგენს ელემენტების ისეთი თანამიმდევრობით გამოტანას მონიტორზე, როგორც ისინი არიან განთავსებული HTML კოდში.

ლისტინგი 8. სტატიკური პოზიცირება

<!DOCTYPE html>

<html>

<head>

<title>სტატიკური პოზიცირება</title>

<style type="text/css">

html, body {

margin: 0px;

padding: 0px;

border: 0px;

}

body {

color: #000;

background: #fff;

}

div {

width: 200px;

height: 100px;

padding: 20px;

border: solid 1px;

}

#first {

background: #ffcc33;

border-color: #000;

color:#006;

}

#second {

background: #006;

color:white;

border-color: #000;

}

#third {

background: #d8d8d8;

border-color: #000; }

#fourth {

background: #006;

color:white;

border-color: #000; }

#fifth {

background: #fc3;

color: #006;

border-color: #000; }

</style>

</head>

<body>

<div id="first">საიტის ქუდი</div>

<div id="second">მენიუს ბლოკი</div>

<div id="third">კონტენტი</div>

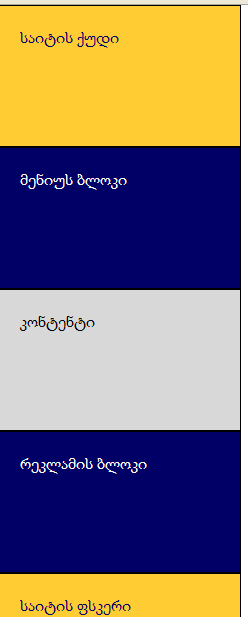
<div id="fourth">რეკლამის ბლოკი</div>

<div id="fifth">საიტის ფსკერი</div>

</body>

</html>

ლისტინგი 8-ს რეალიზაციით მივიღებთ



ნახატი 8. სტატიკური პოზიცირება

ელემენტების განლაგება ნორმალური ნაკადის შეცვლით შესაძლებელია **float** თვისებით, რომელსაც ორი მნიშვნელობა აქვს: left და right; მათი საშუალებით განისაზღვრება ამ ელემენტების გარშემორტყმა, თუ გვაქვს **float:** left მაშინ ელემენტების გარშემორტყმა ხდება მისგან -მისი მხრიდან მარცხნიდან, მაყურებლის მხრიდან მარჯვნიდან და ვღებულობთ არა ვერტიკალურ, არამედ ჰორიზონტალურ პოზიცირებას, ამასთან თანამიმდებრობა ელემენტების გამოჩენისა ვებ-გვერდზე დაცულია

**ლისტინგი 9** გარშემორტყმა

<!DOCTYPE html>

<html>

<head>

<title>სტატიკური პოზიცირება</title>

<style type="text/css">

html, body {

margin: 0px;

padding: 0px;

border: 0px;

}

body {

color: #000;

background: #fff;

}

div {

width: 200px;

height: 100px;

padding: 20px;

border: solid 1px;

float:left;

}

#first {

background: #ffcc33;

border-color: #000;

color:#006;

}

#second {

background: #006;

color:white;

border-color: #000;

}

#third {

background: #d8d8d8;

border-color: #000;

}

#fourth {

background: #006;

color:white;

border-color: #000;

}

#fifth {

background: #fc3;

color: #006;

border-color: #000;

}

</style>

</head>

<body>

<div id="first">საიტის ქუდი</div>

<div id="second">მენიუს ბლოკი</div>

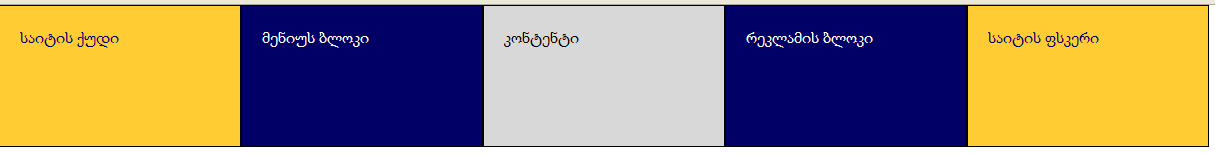
<div id="third">კონტენტი</div>

<div id="fourth">რეკლამის ბლოკი</div>

<div id="fifth">საიტის ფსკერი</div>

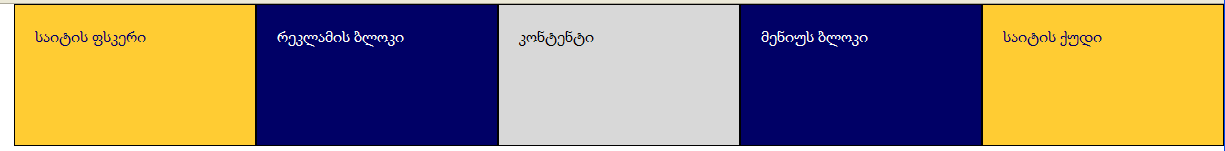
</body>

</html>

****

ნახატი 9. ჰორიზონტალურ პოზიცირება, გარშემორტყმა მარჯვენა მხრიდან

თუ გვაქვს **float:** right , მაშინ ელემენტების გარშემორტყმა ხდება მისგან -მისი მხრიდან მარჯვნიდან, მაყურებლის მხრიდან მარცხნიდან და ვღებულობთ არა ვერტიკალურ, არამედ ჰორიზონტალურ პოზიცირებას, ამასთან თანამიმდებრობა ელემენტების გამოჩენისა ვებ-გვერდზე პირიქითაა, მართლაც თუ ლისტინ 9-ში ყვითლად გამოყოფილ თვისებას შევცვლით **float:** right -ით, მივიღებთ



ნახატი 10. ჰორიზონტალურ პოზიცირება, გარშემორტყმა მარცხენა მხრიდან

clear თვისებით აღიკვეთება გარშემორტმა, თუ მას მიენიჭება მნიშვნელობა left, აკრძალვა გარშემოტყმისა იქნება მარჯვენა მხრიდან; ხოლო თუ კი მას მიენიჭება მნიშვნელობა right, აკრძალვა გარშემოტყმისა იქნება მარცხენა მხრიდან; both მნიშვნელობისთვის ორივე მხრიდანაა აკრძალვა გარშემორტყმისა. თუ კი clear:none; მნიშვნელობა გვაქვს დასაშვებია გარშემორტყმა ორივე მხრიდან.

**position:relative;** - ფარდობითი პოზიცირება. ელემენტის პოზიცია შეიცვლება ნორმალური ნაკადის მიმართ იმ სიდიდეებით, რომელსაც განსაზღვრავს top, bottom, left და right თვისებების მნიშვნელობების. ამ შეცვლილ პოზიციის ელემენტის შემდეგ მომდევნო ელემენტი ისევ ნორმალურ ნაკადშია, ვითომდაც წინა ელემენტს პოზიცია არც კი შეუცვლია. ჩვენს მაგალითში კონტენტის ბლოკს შევუცვალოთ პოზიცირება ფარდობითის საშუალებით. მეცხრე ლისტინგში ფირუზის ფერი კოდი შეიცვლება შემდეგნაირად :

#third {position:relative;

top:-143px;

left:243px;

background: #d8d8d8;

border-color: #000;



ნახატი 11. position:relative; - ფარდობითი პოზიცირება

როცა ლისტინდ 9-ში იცვლება ნაცრისფერი კოდი ამ კოდით:

#fourth {

position:relative;

top:-286px;

left:484px;

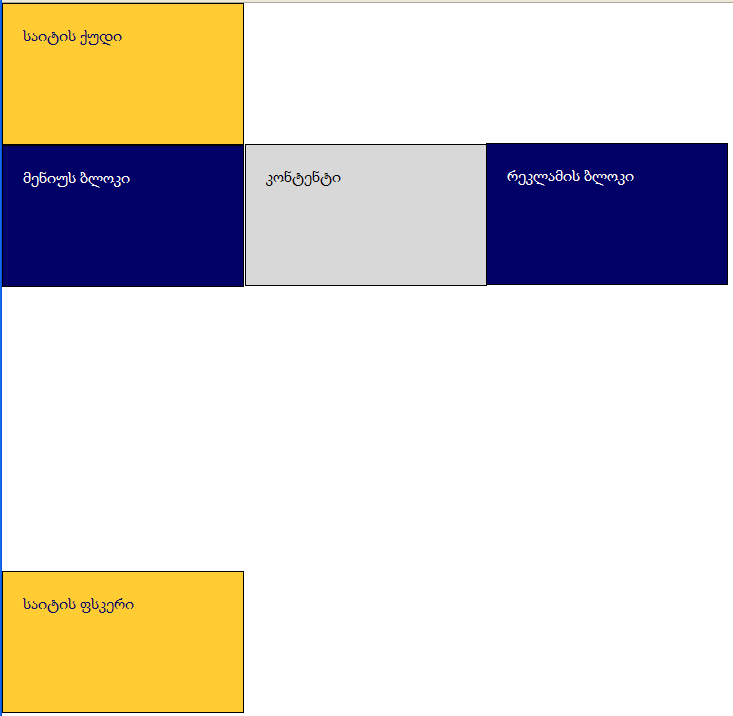
background: #006;

color:white;

border-color: #000;

}

ვღებულობთ ასეთ სურათს



ნახატი 12. position:relative; - ფარდობითი პოზიცირება 2 ბლოკის

თუ კი ეხლა მეხუთე ბლოკის გამოყვანის პოზიციას შევცვლით - ანუ ლისტინგ 9-ში მწვანე კოდს ასეთნაირად წარმოვადგენთ :

#fifth {

position:relative;

top:-286px;

left:0px;

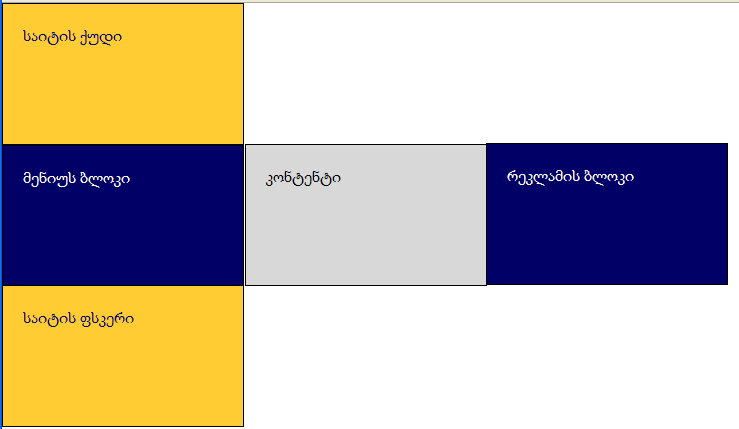
background: #fc3;

color: #006;

border-color: #000;

}

მივიღებთ სურათს:



ნახატი 13. position:relative; - ფარდობითი პოზიცირება 3 ბლოკის

ზოგჯერ პოზიცირებისას ერთი ბლოკი შეიძლება მეორეს გადაეფაროს ან მთბიანად ან ნაწილობრივ, მართლაც თი კი ლისტინგ 9-ში მწვანე ფერის კოდში პოზიციის პიქსელებს შევცვლით:

#fifth {

position:relative;

top:-340px;

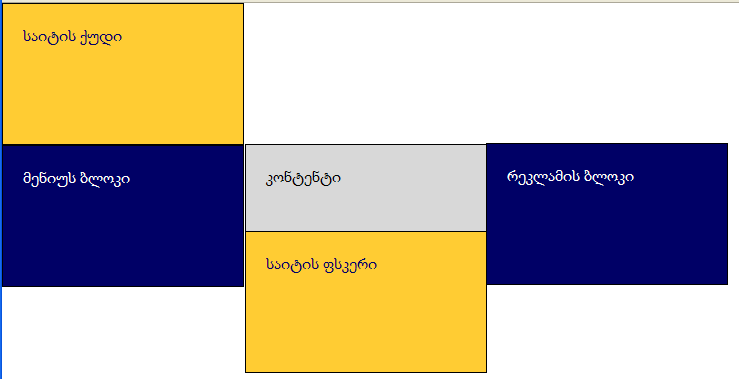
left:243px;

background: #fc3;

color: #006;

border-color: #000;

}



ნახატი 14. position:relative; - ფარდობითი პოზიცირება 3 ბლოკის, ზედდებით

**position:absolute; -** აბსოლუტური პოზიცირებით ელემენტის მდგომარეობა და ზომებიც შეიძლება განისაზღვროს top, bottom, left და right თვისებების მნიშვნე­ლობებით მშობლიური ელემენტის მიმართ. ელემენტები აბსოლუტური პოზიცირებისას გამოდიან ნორმალური ნაკადიდან.

ლისტინგი 10. position:absolute;

<!DOCTYPE html>

<html>

<head>

<title>აბსოლუტური პოზიცირება</title>

<style type="text/css">

html, body {

margin: 0px;

padding: 0px;

border: 0px;

}

body {

color: #000;

background: #fff;

}

div {

width: 200px;

height: 100px;

padding: 20px;

border: solid 1px;

}

#first {

position:absolute;

top:0px;

left:243px;

background: #ffcc33;

border-color: #000;

color:#006;

}

#second {

position:absolute;

top:143px;

left:0px;

background: #006;

color:white;

border-color: #000;

}

#third {

position:absolute;

top:143px;

left:243px;

background: #d8d8d8;

border-color: #000;

}

#fourth {

position:absolute;

top:143px;

left:486px;

background: #006;

color:white;

border-color: #000;

}

#fifth {

position:absolute;

top:286px;

left:243px;

background: #fc3;

color: #006;

border-color: #000;

}

</style>

</head>

<body>

<div id="first">საიტის ქუდი</div>

<div id="second">მენიუს ბლოკი</div>

<div id="third">კონტენტი</div>

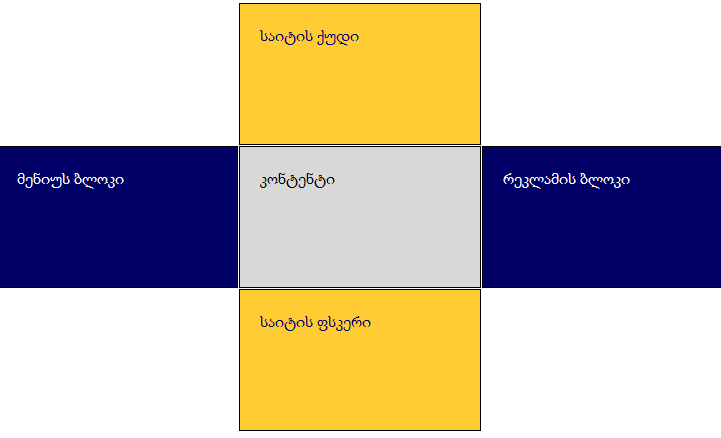
<div id="fourth">რეკლამის ბლოკი</div>

<div id="fifth">საიტის ფსკერი</div>

</body>

</html>

ლისტინგი 10-ის რეალიზაციით ვღებულობთ



ნახატი 15 აბსოლუტური პოზიცირებით მიღებული შედეგი

ლისტინგი 11 აბსოლუტური პოზიცირება ზედდებით

<!DOCTYPE html>

<html>

<head>

<title>აბსოლუტური პოზიცირება</title>

<style type="text/css">

html, body {

margin: 0px;

padding: 0px;

border: 0px;

}

body {

color: #000;

background: #fff;

}

div {

width: 200px;

height: 100px;

padding: 20px;

border: solid 1px;

}

#first {

position:absolute;

top:0px;

left:0px;

background: #ffcc33;

border-color: #000;

color:#006;

}

#second {

position:absolute;

top:50px;

left:55px;

background: #006;

color:white;

border-color: #000;

}

#third {

position:absolute;

top:95px;

left:80px;

background: #d8d8d8;

border-color: #000;

}

#fourth {

position:absolute;

top:140px;

left:100px;

background: #006;

color:white;

border-color: #000;

}

#fifth {

position:absolute;

top:185px;

left:140px;

background: #fc3;

color: #006;

border-color: #000;

}

</style>

</head>

<body>

<div id="first">საიტის ქუდი</div>

<div id="second">მენიუს ბლოკი</div>

<div id="third">კონტენტი</div>

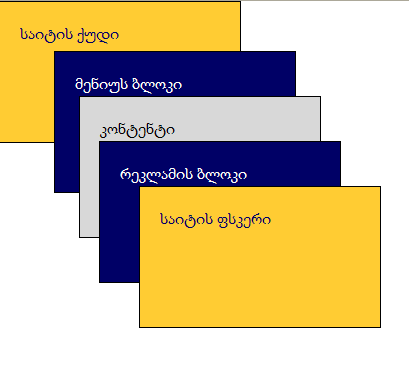
<div id="fourth">რეკლამის ბლოკი</div>

<div id="fifth">საიტის ფსკერი</div>

</body>

</html>

ლისტინგი 11-ის რეალიზაციის შედეგად მიღებული სურათი



ნახატი 16 აბსოლუტური პოზიცირება ზედდებით

ზედდების რიგითობის განსასაზვრა-შეცვლისთვის გამოიყენება თვისება z-index, რომელიც მესამე განზომილებაში ალაგებს ელემენტების ზედდებას. z-index -მა შეიძლება მიიღოს auto - ჩუმათობის პრინციპით განსაზღვრული მდგომარეობა; ასევე მან შეიძლება მიიღოს ნებისმიერი მთელი რიცხვი, რომლითაც განისაზღვრება თუ რომელი ელემენტი იქნება ზევიდან - რაც მეტია მნიშვნელობა, მით უფრო მაღლაა ელემენტი. ასე მაგალითად

ლისტინგი 12 ზედდების შეცვლა

<!DOCTYPE html>

<html>

<head>

<title>აბსოლუტური პოზიცირება</title>

<style type="text/css">

html, body {

margin: 0px;

padding: 0px;

border: 0px;

}

body {

color: #000;

background: #fff;

}

div {

width: 200px;

height: 100px;

padding: 20px;

border: solid 1px;

}

#first {

z-index:5;

position:absolute;

top:0px;

left:0px;

background: #ffcc33;

border-color: #000;

color:#006;

}

#second {

z-index:4;

position:absolute;

top:50px;

left:55px;

background: #006;

color:white;

border-color: #000;

}

#third {

z-index:3;

position:absolute;

top:95px;

left:80px;

background: #d8d8d8;

border-color: #000; }

#fourth {

z-index:2

position:absolute;

top:140px;

left:100px;

background: #006;

color:white;

border-color: #000; }

#fifth {

z-index:1;

position:absolute;

top:185px;

left:140px;

background: #fc3;

color: #006;

border-color: #000; }

</style>

</head>

<body>

<div id="first">საიტის ქუდი</div>

<div id="second">მენიუს ბლოკი</div>

<div id="third">კონტენტი</div>

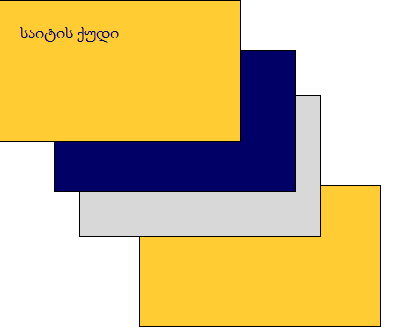
<div id="fourth">რეკლამის ბლოკი</div>

<div id="fifth">საიტის ფსკერი</div>

</body>

</html>

ლისტინგი 12-ის რეალიზაციით მივიღებთ -



ნახატი 17. ზედდების შეცვლა

ამდენად ჩვენ უკვე შეგვიძლია ავაგოთ საიტის უმარტივესი მაკეტი -შევთანხმდეთ, რომ ჰოტიზონტალუტი პოზიციისთვის ის იქნება დრეკადი - ”რეზინის” - რაც საშუალებას მოგვცემს სხვადასხვა მონიტორებისა და ბრაუზერებისთვის სტულად გამოვიყენოთ ეკრანის სიგანე. სიმაღლეში საიტი ფიქსიტებულია. ამასთამ გამოყენებულია მეორე, მესამე და მეოთხე ბლოკებისთვის ე.წ. ”სარჩული” ანუ ისინი მოთავსდნენ კიდევ გარეშე კონტეინერში, რათა IE ბრაუზერებისთვისაც (პრეტენზიული ბრაუზერებისთვის) იქნეს სწორად აღქმული.

**ლისტინგი 13.** საიტის ბლოკური აწყობა

<!DOCTYPE html>

<html>

<head>

<title>სტატიკური პოზიცირება</title>

<style type="text/css">

html, body {

margin: 0px;

padding: 0px;

border: 0px;

}

body {

color: #000;

background: #fff;

}

#first {

padding:2px;

background: #ffcc33;

border-color: #000;

color:#006;

}

#secondcarrier

{

background: #006;

width:20%;

height:550px;

float:left;

}

#thirdcarrier

{

background: #d8d8d8;

height:550px;

width:60%;

float:left;

}

#second {

padding:70px;

background: #006;

color:white;

border-color: #000;

}

#fourthcarrier

{

background: #006;

height:550px;

width:20%;

float:right;

clear:right:

}

#third {

padding:10px;

background: #d8d8d8;

border-color: #000;

}

#fourth {

padding:70px;

background: #006;

color:white;

border-color: #000;

}

#fifth {

padding:2px;

width:100%;

background: #fc3;

color: #006;

border-color: #000;

clear:both;

}

p {

text-align:justify;

}

#cnt {

text-align:center;

font-size:150%;

}

</style>

</head>

<body>

<div id="first"><p id="cnt"><b>საიტის ქუდი</b></p></div>

<div id="secondcarrier">

<div id="second">მენიუს ბლოკი</div>

</div>

<div id="thirdcarrier">

<div id="third">კონტენტი<br><p><img src="w3css.gif" alt="ემბლემა">

<p>CSS -პოზიცირება.</p><p>

CSS -ში პოზიცირების თვისება გვაძლევს საშუალებას ელემენტის მდებარეობის განსაზღვრისა ვებ-გვერდზე . ის ასევე იძლევა საშუალებას, ერთი ელემენტი მეორის უკან განვათავსოთ და როგორ უნდა მოვიქცეთ, როცა ელემენტის შიგთავსი არის ძალიან დიდი.

ელემენტის პოზიცირება შესაძლებელია top, bottom, left და right თვისებების საშუალებით, მაგრამ ეს თვისებები არ მუშაობენ თუ კი პირველად არ არის პოზიცირების- position თვისების მინიჭება. ისინი ასევე სხვადასხვანაირად მოქმედებენ პოზიცირების სხვადასხვა მეთოდისთვის.

სულ არის პოზიცირების ოთხი მეთოდი, რომელიც position თვისებისთვის 4 სხვადასხვა მნიშვნელობის მინიჭებით მიიღებიან:

static - პოზიცირება</p>

</div>

</div>

<div id="fourthcarrier">

<div id="fourth">რეკლამის ბლოკი</div>

</div>

<div id="fifth"><p id="cnt"><b>საიტის ფსკერი</b></p></div>

</body>

</html>

ლისტინგი 13-ის რეალიზაციით ვღებულობთ:



ნახატი 18. ბლოკური აწყობის საიტის