რა არის HTML

**HTML (HyperText Markup Language) წარმოადგენს** ჰიპერტექსტური მარკირების ენას, რომელიც ძირითადად გამოიყენება ინტერნეტის ქსელის დოკუმენტების შესაქმნელად. HTML სათავეს იღებს გასული საუკუნის 90-იანი წლებიდან, როგორც ვებგვერდების შესაქმნელი პრიმიტიული ენა, ხოლო დღეისთვის ინტერნეტი წარმოუდგენელია ამ ენის გარეშე. საიტების აბსოლუტური უმრავლესობა ასე თუ ისე იყენებს HTML-ს.

2014 წელს ოფიციალურად დასრულდა მუშაობა ახალ სტანდარტზე - HTML5-ზე, რომელმაც ფაქტიურად რევოლუციური ცვლილებები შეიტანა HTML-ში.

კონკრეტულად რა მოიტანა HTML5-მა?

* HTML 5-მა განსაზღვრა პარსინგის ახალი ალგორითმი DOM-ის (Document Object Model) სტრუქტურის შესაქმნელად;
* ახალი ელემენტების და ტეგების დამატება, როგორიცაა ელემენტები video, audio და სხვა;
* HTML-ის არსებული ელემენტების გამოყენების წესების და სემანტიკის ხელახალი განსაზღვრა.

ახალი ფუნქციების დამატებით HTML5 გახდა არა უბრალოდ HTML-ის ახალი ვერსია, არამედ ფაქტიურად გახდა პლატფორმა პროგრამული უზრუნველყოფის შესაქმნელად, ხოლო მისი გამოყენების სფერო შორს გასცდა ინტერნეტის ფარგლებს: HTML5 გამოიყენება მობილური აპლიკაციებისთვის ანდროიდის, iOS, Windows Mobile ბაზაზე, ასევე დესკტოპ-პროგრამების შესაქმნელადაც.

საბოლოოდ, HTML5 ძირითადად გამოიყენება შემდეგი დანიშნულებით:

* HTML5, როგორც ჰიპერტექსტური მარკირების განახლებული ენა, როგორც წინა HTML4-ის განვითარება;
* HTML5, როგორც მძლავრი პლატფორმა ვებაპლიკაციების შექმნისთვის, რომელიც HTML5-თან ერთად ასევე მოიცავს პროგრამირების ენა JavaScript -ს და სტილების კასკადურ ცხრილებს CSS3-ს.

HTML5-ის განვითარებით დაკავებულია World Wide Web Consortium (შემოკლებით W3C - მსოფლიო ქსელის კონსორციუმი) - დამოუკიდებელი საერთაშორისო ორგანიზაცია, რომელიც განსაზღვრავს HTML5-ის სტანდარტებს სპეციფიკაციების სახით. მიმდინარე სრული სპეციფიკაციის ნახვა შესაძლებელია მისამართზე <https://www.w3.org/TR/html5/>.

ბრაუზერების მხარდაჭერა

უნდა აღინიშნოს, რომ HTML5-ის სპეციფიკაციებსა და ვებბრაუზერების მიერ ამ ტექნოლოგიების გამოყენებას შორის ყოველთვის იყო სხვაობა, თუმცა ბრაუზერების უმეტესობამ ამ სტანდარტების დანერგვა დაიწყო ჯერ კიდევ მის ოფიციალურად გამოქვეყნებამდე და დღეისათვის ბრაუზერების ბოლო ვერსიების უმრავლესობა იყენებს HTML5-ის ფუნქციონალის უმეტეს ნაწილს (Google Chrome, Firefox, Opera, Internet Explorer 11, Microsoft Edge), ამასთნ ბრაუზერების ძველ ვერსიებს არ გააჩნიათ HTML5-ის მხარდაჭერა, მაგალითად Internet Explorer 8 და უფრო ადრეული ვერსიები, ხოლო IE 9, 10-ს გააჩნია სტანდარტების მხოლოდ ნაწილის მხარდაჭერა.

ამასთან, იმ ბრაუზერებს, რომელთაც გააჩნიათ HTML5-ის სტანდარტების მხარდაჭერა, შესაძლოა არ გააჩნდეთ კონკრეტული ელემენტების მხარდაჭერა და ეს უნდა გავითვალისწინოთ მუშაობისას.

ბრაუზერის მიერ HTML5-ს სტანდარტების მხარდაჭერის დონის შემოწმება შესაძლებელია სპეციალური სერვისის მეშვეობით მისამართზე: [http://html5test.com](http://html5test.com/).

### საჭირო ინსტრუმენტები

რა არის საჭირო HTML5-თან სამუშაოდ? პირველ რიგში საჭიროა ტექსტური რედაქტორი ვებგვერდის html-ის ტექსტის ასაკრეფად. ამ დროისათვის ერთ-ერთი ტექსტური რედაქტორი, რომელიც ფართოდ გამოიყენება არის Notepad++, მისი ნახვა და გადმოწერა შეიძლება შემდეგი მისამართიდან: [http://notepad-plus-plus.org](http://notepad-plus-plus.org/). მისი უპირატესობა მდგომარეობს იმაში რომ არის უფასო და გააჩნია კოდის შემოწმების ინსტრუმენტები.

ასევე ფართოდ გამოიყენება კროსპლატფორმული ტექსტური რედაქტორი [Visual Studio Code](https://code.visualstudio.com/). მას გააჩნია უფრო მეტი შესაძლებლობები, ვიდრე Notepad++-ს და ასევე არის უფასო, გარდა ამისა, მისი გამოყენება შეიძლება როგორც Windows-ში, ასევე MacOS-ში და Linux-ის ბაზაზე არსებულ ოპერაციულ სისტემებში. ამ სახელმძღვანელოს შექმნის პროცესში გამოყენებულ იქნა Visual Studio Code.

ასევე დაგვჭირდება ვებბრაუზერი შექმნილი ვებგვერდების გაშვებისა და შემოწმებისთვის. ამ მიზნებისთვის გამოდგება ნებისმიერი გავრცელებული ბრაუზერი - Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.

HTML5-ის ელემენტები და ატრიბუტები

სანამ შევუდგებით საკუთარი ვებგვერდის შექმნას, განვიხილოთ ძირითადი სამშენებლო ბლოკები, აგურები, რისგანაც შედგება ვებგვერდი.

HTML5-ის დოკუმენტი, ისევე როგორც ნებისმიერი სხვა დოკუმენტი, შედგება ელემენტებისაგან, ხოლო ელემენტები შედგება ტეგებისაგან. როგორც წესი, ელემენტს გააჩნია საწყისი და საბოლოო ტეგი, რომლებიც მოქცეულია კუთხოვან ფრჩხილებში. მაგალითად:

| <div>div ელემენტის ტექსტი</div> |
| --- |

აქ განსაზღვრულია ელემენტი div, რომელსაც გააჩნია საწყისი ტეგი <div> და საბოლოო ტეგი</div>. მათ შორის მოქცეულია div ელემენტის შიგთავსი. ამ შემთხვევაში შიგთავსის სახით წარმოდგენილია ტექსტი "div ელემენტის ტექსტი".

ელემენტი ასევე შეიძლება შედგებოდეს ერთი ტეგისგან, მაგალითად ელემენტი <br />, რომლის დანიშნულებაა სტრიქონის გადატანა:

| <div>div ელემენტის <br />ტექსტი </div> |
| --- |

ასეთ ელემენტს ასევე უწოდებენ ცარიელ ელემენტს. თუმცა მაგალითში გამოყენებულია სლეში ("/"), მაგრამ მისი გამოყენება შესაძლებელია სლეშის გარეშეც შემდეგი სახით: <br>.

თითოეულ ელემენტს საწყის ტეგში შეიძლება ჰქონდეს ატრიბუტები, მაგალითად:

| * <div style="color:red;">ღილაკი</div> * <input type="button" value="დააჭირეთ"> |
| --- |

აქ მოცემულია 2 ელემენტი: div და input. div ელემენტს გააჩნია ატრიბუტი style. ტოლობის ნიშნის შემდეგ ბრჭყალებში წერია ატრიბუტის მნიშვნელობა type="color:red;". ამ შემთხვევაში "color:red;" განსაზღვრავს, რომ ტექსტის ფერი არის წითელი. მეორე ელემენტს - input-ს გააჩნია 2 ატრიბუტი: type (მიუთითებს ელემენტის ტიპს - ღილაკი) და value (განსაზღვრავს ღილაკის ტექსტს).

არსებობს გლობალური, ანუ ყველა ელემენტისათვის საერთო ატრიბუტები, როგორიცაა მაგალითად, style, და არის სპეციფიური, რომელიც გამოიყენება მხოლოდ ზოგიერთი ელემენტისათვის, მაგალითად type. ჩვეულებრივი ატრიბუტების გარდა არსებობს ლოგიკური ატრიბუტები, რომელთაც შეიძლება მნიშვნელობა არ მიენიჭოს, მაგალითად disabled:

| <input type="button" value="დააჭირეთ" disabled> |
| --- |

ატრიბუტი disabled მიუთითებს, რომ ელემენტი არააქტიურია.

გლობალური ატრიბუტები

HTML5-ში არის გლობალური ატრიბუტები, რომლებიც გამოიყენება ყველა ელემენტისათვის:

* accesskey: განსაზღვრავს სწრაფ კლავიშს ელემენტისთვის
* class: ელემენტს ანიჭებს CSS-ის კლასს
* contenteditable: განსაზღვრავს, რედაქტირებადია თუ არა ელემენტი
* contextmenu: განსაზღვრავს ელემენტის კონტექსტურ მენიუს,რომელიც გაიხსნება ელემენტზე მაუსის მარჯვენა კლავიშის დაჭერისას
* dir: განსაზღვრავს ტექსტის მიმართულებას ელემენტში
* draggable: განსაზღვრავს, შეიძლება თუ არა ელემენტის გადათრევა
* dropzone: განსაზღვრავს, კოპირდება თუ არა გადასატანი მონაცემები ელემენტზე გადათრევისას
* hidden: მალავს ელემენტს
* id: ელემენტის უნიკალური იდენტიფიკატორი. ვებგვერდზე იდენტიფიკატორი არ უნდა მეორდებოდეს
* lang: განსაზღვრავს ელემენტის ენას
* spellcheck: განსაზღვრავს, იქნება თუ არა მართლწერის შემოწმება გამოყენებული ელემენტის მიმართ
* style: განსაზღვრავს ელემენტის სტილს
* tabindex: განსაზღვრავს თანმიმდევრობას, რომლითაც მოხდება გადართვა ელემენტიდან ელემენტზე TAB კლავიშით
* title: ელემენტის დამატებითი აღწერა
* translate: განსაზღვრავს, უნდა ითარგმნოს თუ არა ელემენტის შინაარსი

სამომხმარებლო ატრიბუტები

წინა ვერსიებისგან განსხვავებით, HTML5-ში დაემატა სამომხმარებლო ატრიბუტები (custom attributes). ახლა უკვე ვებგვერდის შემქმნელს შეუძლია დაამატოს ნებისმიერი პარამეტრი, რომელსაც წინსართად უნდა ჰქონდეს data. მაგალითად:

| <input type="button" value="დააჭირეთ" data-color="red"> |
| --- |

აქ ელემენტზე დამატებულია სამომხმარებლო ატრიბუტი data-color, რომელსაც აქვს მნიშვნელობა red .

ერთმაგი და ორმაგი ბრჭყალები

ატრიბუტების მნიშვნელობის განსაზღვრისას შეიძლება გამოყენებულ იქნეს როგორც ერთმაგი, ისევე ორმაგი ბრჭყალები, თუმცა უფრო ხშირად გამოიყენება ორმაგი ბრჭყალები. ერთმაგი ბრჭყალები ძირითადად გამოიყენება იმ შემთხვევაში, როცა ატრიბუტის მნიშვნელობა თვითონ შეიცავს ბრჭყალებს. მაგალითად:

| <input type="button" value='ღილაკი "გამარჯობა, მსოფლიო"'> |
| --- |

HTML5-ის დოკუმენტის შექმნა

ელემენტები წარმოადგენენ აგურებს, რომლებისგანაც შედგება html5-ის დოკუმენტი. დოკუმენტის შესაქმნელად საჭიროა შევქმნათ ტექსტური ფაილი და გაფართოება შევცვალოთ .html-ზე. შევქმნათ ფაილი და დავარქვათ index.html. შემდეგ გავხსნათ ეს ფაილი ნებისმიერი ტექსტური რედაქტორით და დავამატოთ შემდეგი ტექსტი:

| * <!DOCTYPE html> * <html> * </html> |
| --- |

HTML5-ის დოკუმენტის შესაქმნელად პირველ რიგში საჭიროა 2 ელემენტი: DOCTYPE და html. ელემენტი doctype ანუ Document Type Declaration ამცნობს ბრაუზერს დოკუმენტის ტიპს, ჩვენს შემთხვევაში დოკუმენტის ტიპია html, რაც იმას ნიშნავს, რომ გამოიყენება HTML5 და არა რომელიმე წინა ვერსია. ხოლო html საწყის და საბოლოო ტეგებს შორის მოქცეული იქნება მთელი დოკუმენტი. html ელემენტის შიგნით დავამატოთ ორი ახალი ელემენტი: head და body. ელემენტი head შეიცავს ვებგვერდის მეტამონაცემებს: გვერდის სათაური, კოდირების ტიპი, და ა.შ., ასევე მიმართვებს რესურსებისადმი: სტილები, სკრიპტები, თუ ისინი გამოიყენება ვებ-გვერდზე. ელემენტი body კი შეიცავს ვებგვერდის მთელ შინაარსს.

მოდით, შევცვალოთ index.html ფაილის შიგთავსი შემდეგი სახით:

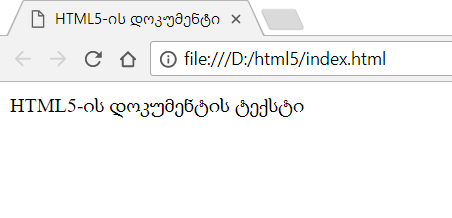
| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>HTML5-ის დოკუმენტი</title> * </head> * <body> * <div>HTML5-ის დოკუმენტის ტექსტი</div> * </body> * </html> |
| --- |

ელემენტში head გვაქვს ორი ელემენტი:

* ელემენტი title წარმოადგენს ვებგვერდის სათაურს
* ელემენტი meta წარმოადგენს გვერდის მეტაინფორმაციას. ჩვენს შემთხვევაში ატრიბუტი charset="utf-8" უჩვენებს, რომ გვერდის კოდირება არის utf-8. ქართული ნაწერის კორექტულად გამოსაჩენად უნდა მივუთითოთ ეს კოდირება.

ელემენტ body-ის შიგნით გვაქვს მხოლოდ ერთი ელემენტი div, რომელიც განსაზღვრავს ცალკე ბლოკს. შიგნით უბრალო ტექსტია.

შევინახოთ ფაილი და გავუშვათ ბრაუზერში:



ამგვარად, ჩვენ შევქმენით HTML5-ის დოკუმენტი, რომელსაც სათაურად აქვს "HTML5-ის დოკუმენტი" და ბრაუზერში ჩანართზე სწორედ ეს წარწერაა, ბრაუზერის ძირითად ველში კი ჩანს ის ტექსტი, რომელიც ჩავწერეთ body -ის div ელემენტში.

HTML5-ის სინტაქსის სტილები

HTML5-ის დოკუმენტის შექმნისას შეიძლება გამოვიყენოთ ორი განსხვავებული სტილი: HTML და XML.

სტილი HTML ითვალისწინებს შემდეგ პრინციპებს:

* ელემენტს შეიძლება არ ჰქონდეს საწყისი და საბოლოო ტეგები
* ელემენტს შეიძლება არ ჰქონდეს საბოლოო ტეგი
* მხოლოდ ცარიელი ტეგები (მაგალითად br,img,link) შეიძლება დაიხუროს სლეშის მეშვეობით />
* ტეგების და ატრიბუტების რეგისტრს არა აქვს მნიშვნელობა
* ატრიბუტების მნიშვნელობები შეიძლება არ მოვაქციოთ ბრჭყალებში
* ზოგიერთ ატრიბუტს შეიძლება არ ჰქონდეს მნიშვნელობა (checked, disabled)
* სპეციალური სიმბოლოების ეკრანირება არ ხდება
* დოკუმენტს უნდა ჰქონდეს ელემენტი DOCTYPE

ეს არის ე.წ. დაშვებული სტილი, რომელიც გარკვეულწილად ლოიალურია წესების მიმართ დოკუმენტის შექმნისას.

HTML5 დოკუმენტი ასევე შეიძლება იყოს აღწერილი XML სინტაქსის მეშვეობით. ასეთ სტილს ასევე უწოდებენ XHTML. ის გამოიყენება, თუ დოკუმენტის თავში content type -ს აქვს მნიშვნელობა application/xml+xhtml. ამ სტილისათვის დამახასიათებელია შემდეგი წესები:

* ყველა ელემენტს უნდა ჰქონდეს საწყისი ტეგი
* არაცარიელ ტეგებს ასევე უნდა ჰქონდეთ საბოლოო ტეგი
* ნებისმიერი ელემენტის დახურვა შესაძლებელია სლეშით />
* ტეგების და ატრიბუტების სახელები მგრძნობიარეა რეგისტრის მიმართ, როგორც წესი, იწერება დაბალ რეგისტრში
* ატრიბუტების მნიშვნელობები უნდა იყოს მოქცეული ბრჭყალებში
* არ დაიშვება ატრიბუტები მნიშვნელობის გარეშე (checked="checked" უბრალოდ checked-ის ნაცვლად)
* უნდა მოხდეს სპეციალური სიმბოლოების ეკრანირება

შევადაროთ ეს ორი სინტაქსის სტილი. HTML5:

| * <!DOCTYPE html> * <html> * <head> * <meta charset=utf-8> * <title>სათაური</title> * </head> * <body> * <p>HTML5-ის დოკუმენტის ტექსტი<br> * <input type=button value=დააჭირეთ> * </body> * </html> |
| --- |

იგივე მაგალითი XHTML სტილში:

| * <!doctype html> * <html xmlns="http://www.w3.org/1999/xhtml"> * <head> * <meta charset="utf-8"> * <title>სათაური</title> * </head> * <body> * <p>HTML5-ის დოკუმენტის ტექსტი<br /> * <input type="button" value="დააჭირეთ" /></p> * </body> * </html> |
| --- |

XHTML სტილის გამოყენებისას ასევე უნდა მივუთითოთ დოკუმენტის სახელების არე <html xmlns="http://www.w3.org/1999/xhtml">.

კონკრეტული სტილის არჩევა პროგრამისტის ან ვებდიზაინერის სურვილზეა დამოკიდებული. ხშირად გამოიყენება შერეული სტილი, როცა წესები აღბულია როგორც ერთი, ასევე მეორე სტილისგან. ამასთან, საჭიროა გავითვალისწინოით, რომ ელემენტისთვის საწყისი და საბოლოო ტეგების არსებობა ამცირებს იმის ალბათობას, რომ ელემენტი არასწორად იქნება ინტერპრეტირებული ბრაუზერის მიერ. ასევე, ატრიბუტის მნიშვნელობების ბრჭყალებში ჩასმით თავს დავიზღვევთ მოსალოდნელი შეცდომებისაგან. მაგალითად, ატრიბუტმა კლასმა შეიძლება ერთდროულად რამდენიმე მნიშვნელობა მიიღოს: <div class="navmenu bigdesctop">. თუ მნიშვნელობას არ ჩავსვამთ ბრჭყალებში, ბრაუზერი class-ის მნიშვნელობად მიიღებს მხოლოდ navmenu-ს, ხოლო bigdesctop-ს ჩათვლის სხვა ატრიბუტად, რის გამოც ელემენტი არასწორად აისახება ბრაუზერში.

იმის შესამოწმებლად, თუ რამდენად სწორადაა შედგენილი html დოკუმენტი, შეგვიძლია ვისარგებლოთ ვალიდატორით: [https://validator.w3.org](https://validator.w3.org/).

ელემენტი head და ვებგვერდის მეტამონაცემები

როგორც წესი, html დოკუმენტის ერთ-ერთ უმნიშვნელოვანეს ელემენტს წარმოადგენს head, რომლის დანიშნულებაც მდგომარეობს ვებ-გვერდის მეტამონაცემების და თანმდევი ინფორმაციის განსაზღვრაში. მეტამონაცემები შეიცავენ ინფორმაციას ვებ-გვერდის შესახებ.

სათაური

სათაურის მისათითებლად გამოიყენება ელემენტი title:

| <title>ვებ-გვერდის სათაური</title> |
| --- |

ელემენტი base

ელემენტი base განსაზღვრავს ვებ-გვერდის საბაზისო მისამართს, რომლის მიმართაც განისაზღვრება გვერდზე გამოყენებული სხვა მისამართები:

| * <!doctype html> * <head> * <base href="content/"> * <meta charset="utf-8"> * <title>ვებ-გვერდის სათაური</title> * </head> * <body> * <a href="newpage.html">ახალ გვერდზე გადასვლა</a> * </body> * </html> |
| --- |

მიუხედავად იმისა, რომ ბმულს "ახალ გვერდზე გადასვლა" მისამართში უწერია "newpage.html", ფაქტიური მისამართი იქნება "content/newpage.html". ანუ ვებ-გვერდის საქაღალდეში უნდა იყოს ქვესაქაღალდე content და მასში ფაილი newpage.html.

შეიძლება base ელემენტში მივუთითოთ სრული მისამართიც:

| <base href="http://microsoft.com/"> |
| --- |

### ამ შემთხვევაში ბმული გადაგვიყვანს მისამართზე "http://microsoft.com/newpage.html".

### ელემენტი meta

ელემენტი meta განსაზღვრავს ვებგვერდის მეტამონაცემებს.

იმისათვის, რომ უნიკოდის ტექსტის შემცველი გვერდები კორექტულად გამოჩნდეს ბრაუზერში, აუცილებელია მივუთითოთ კოდირება utf-8:

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

### ამასთან, თვითონ დოკუმენტის კოდირებაც უნდა ემთხვეოდეს meta კოდირებას.

ელემენტს meta ასევე აქვს 2 ატრიბუტი: name და content. name არის ატრიბუტის სახელი, content - ატრიბუტის მნიშვნელობა.

HTML დოკუმენტში არის 5 ტიპის მეტამონაცემები:

* application name: ვებ-აპლიკაციის სახელი, რომლის ნაწილიცაა ეს დოკუმენტი
* author: დოკუმენტის ავტორი
* description: დოკუმენტის მოკლე აღწერა
* generator: პროგრამის სახელი, რომელშიც დაგენერირდა დოკუმენტი
* keywords: საკვანძო სიტყვები (გამოიყენება საძიებო სისტემების მიერ ძებნისთვის)

უნდა აღინიშნოს, რომ მეტად მნიშვნელოვანი ატრიბუტია description. მის მნიშვნელობას საძიებო სისტემები ხშირად იყენებენ, როგორც ანოტაციას ვებგვერდისთვის.

დავამატოთ დოკუმენტში რამდენიმე meta ელემენტი:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <base href="content/"> * <title>ვებგვერდის სათაური</title> * <meta name="description" content="ჩემი პირველი HTML5 დოკუმენტი"> * <meta name="author" content="ავტორი"> * </head> * <body> * <a href="newpage.html">HTML5 დოკუმენტის შინაარსი</a> * </body> * </html> |
| --- |

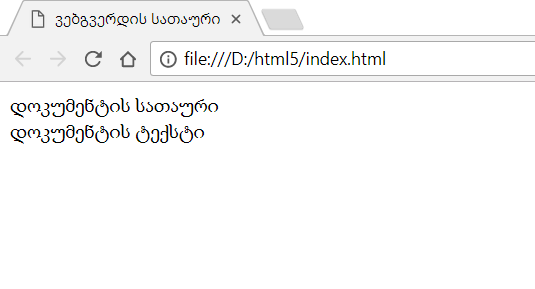
დაჯგუფების ელემენტები

HTML-ში არის რიგი ელემენტებისა, რომლებიც განკუთვნილია ვებგვერდის შიგთავსის (content) დასაჯგუფებლად.

ელემენტი div

ელემენტი div გამოიყენება ვებგვერდის შიგთავსის სტრუქტურულად დაჯგუფებისთვის, ცალ-ცალკე ბლოკებში მოსაქცევად. div ქმნის ბლოკს, რომელიც ბრაუზერის მთელ სიგანეზე ვრცელდება (თუ სხვა რამეს არ მივუთითებთ). div-ის მომდევნო ელემენტი გადადის ახალ სტრიქონზე. მაგალითად:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <base href="content/"> * <title>ვებგვერდის სათაური</title> * <meta name="description" content="ჩემი პირველი HTML5 დოკუმენტი"> * <meta name="author" content="ავტორი"> * </head> * <body> * <div>დოკუმენტის სათაური</div> * <div>დოკუმენტის ტექსტი</div> * </body> * </html> |
| --- |



პარაგრაფები

პარაგრაფები იქმნება <p> და </p> ტეგების მეშვეობით. თითოეული პარაგრაფი იწყება ახალი სტრიქონიდან.მაგალითად:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <base href="content/"> * <title>ვებგვერდის სათაური</title> * <meta name="description" content="ჩემი პირველი HTML5 დოკუმენტი"> * <meta name="author" content="ავტორი"> * </head> * <body> * <div>დოკუმენტის სათაური</div> * <div> * <p>პირველი პარაგრაფი</p> * <p>მეორე პარაგრაფი</p> * </div> * </body> * </html> |
| --- |

### HTML5-ის დოკუმენტი

### თუ ერთი პარაგრაფის ფარგლებში გვინდა სტრიქონის გადატანა, უნდა გამოვიყენოთ ელემენტი <br>:

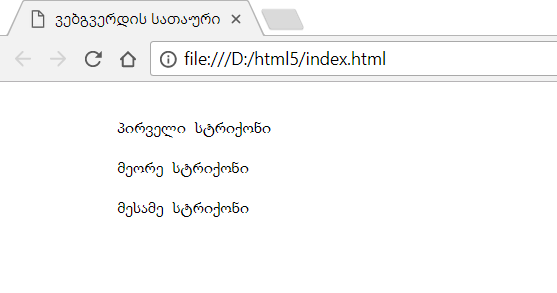
| <p>პირველი სტრიქონი.<br />მეორე სტრიქონი.</p> |
| --- |

### ელემენტი pre

ელემენტ pre-ს გამოაქვს წინასწარ დაფორმატებული ტექსტი:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <base href="content/"> * <title>ვებგვერდის სათაური</title> * <meta name="description" content="ჩემი პირველი HTML5 დოკუმენტი"> * <meta name="author" content="ავტორი"> * </head> * <body> * <pre> * პირველი სტრიქონი * მეორე სტრიქონი * მესამე სტრიქონი * </pre> * </body> * </html> |
| --- |

### 



### ელემენტი span

ელემენტი span-თ ხდება გარკვეული ტექსტის შემოგარსვა მთელ სიგრძეზე და მის სტილიზაციას ემსახურება. განსხვავებით div ელემენტისაგან და პარაგრაფისგან, span-ს არ გადააქვს ტექსტი ახალ სტრიქონზე:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>HTML5-ის დოკუმენტი</title> * </head> * <body> * <div>HTML5-ის დოკუმენტის სათაური</div> * <div> * <p><span style="color:red;">პირველი</span> პარაგრაფი</p> * <p><span>მეორე</span> პარაგრაფი</p> * </div> * </body> * </html> |
| --- |

### HTML5-ის დოკუმენტი

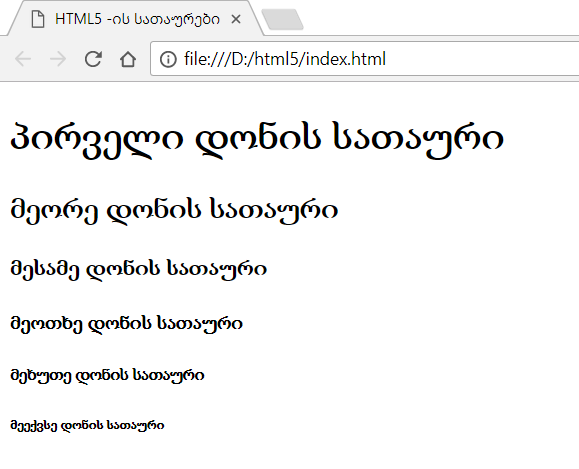
### აღსანიშნავია, რომ თვითონ span არაფერს არ აკეთებს, მეორე პარაგრაფში იგი არ ცვლის ტექსტს არანაირად. პირველ პარაგრაფში span-ს გააჩნია ატრიბუტი style="color:red;", რომელიც ცვლის ტექსტის სტილს (ცვლის ფერს).

თუ div და p არის ბლოკის ტიპის ელემენტები და ისინი შეიძლება შეიცავდნენ ნებისმიერ ელემენტებს, span არის სტრიქონული ტიპის. ამასთან არაა რეკომენდირებული span-ის შიგნით მოვაქციოთ ბლოკური ელემენტები.

სათაურები

ელემენტები <h1>, <h2>, <h3>, <h4>, <h5> и <h6> ემსახურება სხვადასხვა დონის სათაურების შექმნას:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>HTML5 -ის სათაურები</title> * </head> * <body> * <h1>პირველი დონის სათაური</h1> * <h2>მეორე დონის სათაური</h2> * <h3>მესამე დონის სათაური</h3> * <h4>მეოთხე დონის სათაური</h4> * <h5>მეხუთე დონის სათაური</h5> * <h6>მეექვსე დონის სათაური</h6> * </body> * </html> |
| --- |



სათაურებს გააჩნია გამუქებული შრიფტი და შრიფტის ზომა ყველაზე დიდი <h1> -დან ყველაზე პატარა <h6>-მდე.

სათაურების დამატებისას უნდა გავითვალისწინოთ, რომ ვებგვერდს უნდა ჰქონდეს მხოლოდ ერთი პირველი დონის სათაური <h1>. ის ასრულებს ვებგვერდის ძირითადი სათაურის როლს.

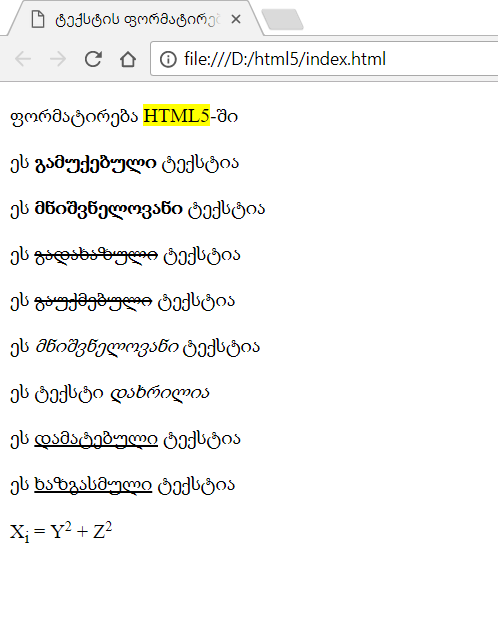
ტექსტის ფორმატირება

html-ის მთელი რიგი ელემენტები განკუთვნილია ტექსტის ფორმატირებისთვის, მაგალითად გასამუქებლად, ხაზგასასმელად და ა. შ. განვიხილოთ ეს ელემენტები:

* <b>: ტექსტის გამუქება
* <del>: ტექსტის გადახაზვა
* <i>: ტექსტის დახრა
* <em>: ტექსტის დახრა, ოღონდ <i>-საგან განსხვავებით, ატარებს ლოგიკურ ხასიათს და ხაზს უსვამს მის მნიშვნელობას
* <s>: ტექსტის გადახაზვა
* <small> ტექსტის დაპატარავება
* <strong> ტექსტის გამუქება, ოღონდ <b>-საგან განსხვავებით, ატარებს ლოგიკურ ხასიათს და ხაზს უსვამს მის მნიშვნელობას
* <sub>: ტექსტის ჩამოწევა სტრიქონის ქვემოთ
* <sup>: ტექსტის აწევა სტრიქონის ზევით
* <u>: ტექსტის ხაზგასმა
* <ins>: ჩასმული (დამატებული) ტექსტი
* <mark>: ტექსტის გამოყოფა ფერით. ხაზს უსვამს მის მნიშვნელობას

გამოვიყენოთ ეს ელემენტები:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>ტექსტის ფორმატირება HTML5-ში</title> * </head> * <body> * <p>ფორმატირება <mark>HTML5</mark>-ში</p> * <p>ეს <b>გამუქებული</b> ტექსტია</p> * <p>ეს <strong>მნიშვნელოვანი</strong> ტექსტია</p> * <p>ეს <del>გადახაზული</del> ტექსტია</p> * <p>ეს <s>გაუქმებული</s> ტექსტია</p> * <p>ეს <em>მნიშვნელოვანი</em> ტექსტია</p> * <p>ეს ტექსტი <i>დახრილია</i> </p> * <p>ეს <ins>დამატებული</ins> ტექსტია</p> * <p>ეს <u>ხაზგასმული</u> ტექსტია</p> * <p>X<sub>i</sub> = Y<sup><small>2</small></sup> + Z<sup><small>2</small></sup></p> * </body> * </html> |
| --- |



სურათებთან მუშაობა

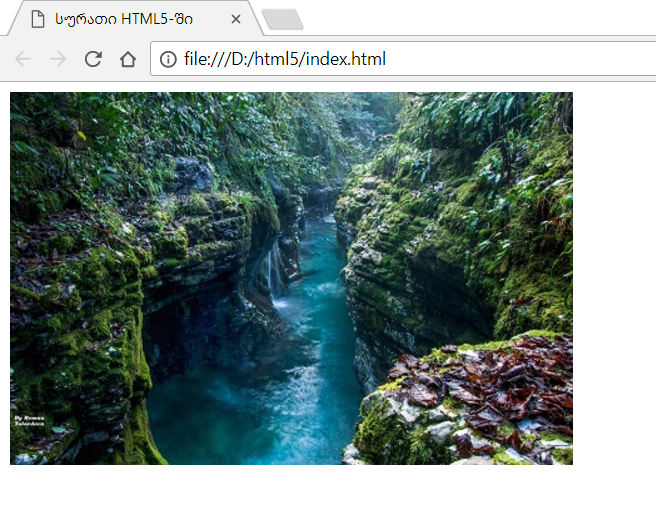
სურათების გამოსატანად HTML-ში გამოიყენება ელემენტი <img>, რომელსაც გააჩნია ორი მნიშვნელოვანი ატრიბუტი:

* src: სურათის მისამართი. ეს შეიძლება იყოს ფარდობითი ან აბსოლუტური მისამართი ფაილურ სისტემაში ან მისამართი ინტერნეტში
* alt: სურათის ტექსტური აღწერა. თუ რაღაც მიზეზით ვერ ხერხდება სურათის გამოტანა ეკრანზე, მის ნაცვლად გამოჩნდება alt ატრიბუტში მითითებული მნიშვნელობა

alt-ს კიდევ ერთი დანიშნულება აქვს. საძიებო სისტემებში სურათის ინდექსირება ხდება სწორედ მისი მნიშვნელობით.

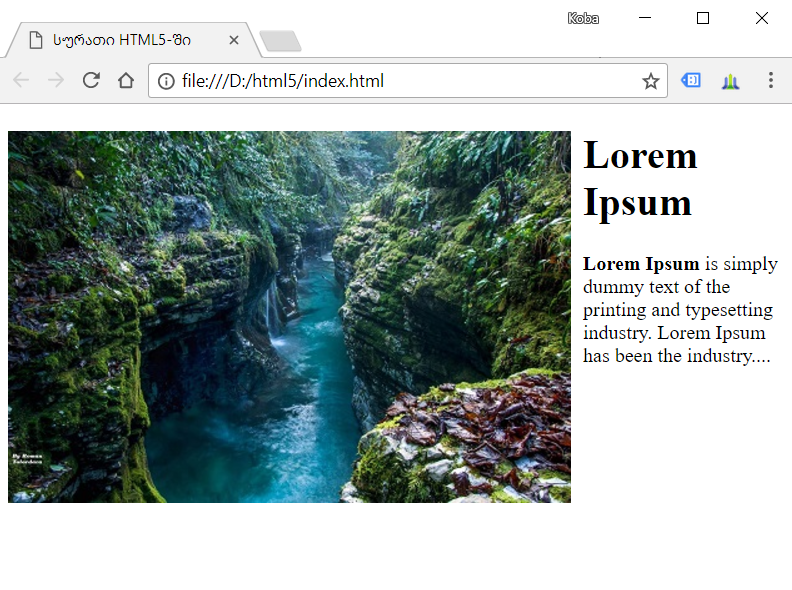
იგივე საქაღალდეში, სადაც index.html-ია დავდოთ რამე სურათი და გამოვიტანოთ ის ვებგვერდზე:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>სურათი HTML5-ში</title> * </head> * <body> * <img src="kanioni.jpg" alt="martvilis kanioni"> * </body> * </html> |
| --- |



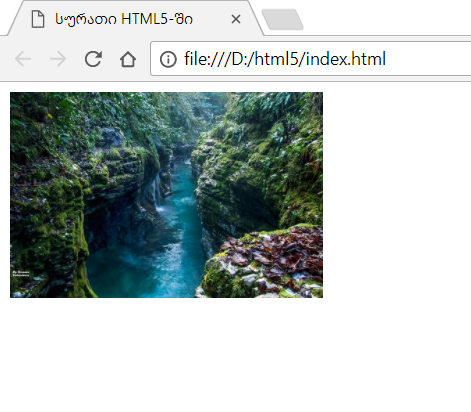
სტილებისა და ფორმატირების ელემენტების გამოყენებით შესაძლებელია სურათისა და ტექსტის კომბინირება. მაგალითად:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>სურათი HTML5-ში</title> * </head> * <body> * <div> * <img src="kanioni.jpg" alt="მარტვილის კანიონი" style="float:left; margin-right:10px;" /> * <h1>Lorem Ipsum</h1> * <b>Lorem Ipsum</b> is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry.... * </div> * </body> * </html> |
| --- |



სურათისთვის მნიშვნელოვანი ატრიბუტებია ასევე ზომები width და height. მათი მითითების გარეშე სურათი იღებს ორიგინალურ ზომებს, ხოლო მითითების შემთხვევაში - მითითებულს. შესაძლებელია მივუთითოთ მხოლოდ ერთ-ერთი (სიგანე ან სიმაღლე), მეორე პროპორციულად მიენიჭება ავტომატურად:

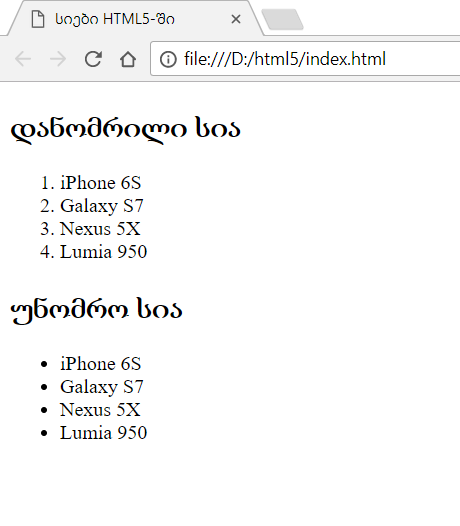
| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>სურათი HTML5-ში</title> * </head> * <body> * <img src="kanioni.jpg" alt="martvilis kanioni" width="250px" height="165"> * </body> * </html> |
| --- |



სიები

სიების შესაქმნელად HTML5-ში გამოიყენება ელემენტები <ol> (დანომრილი სია) და <ul> (უნომრო სია):

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>სიები HTML5-ში</title> * </head> * <body> * <h2>დანომრილი სია</h2> * <ol> * <li>iPhone 6S</li> * <li>Galaxy S7</li> * <li>Nexus 5X</li> * <li>Lumia 950</li> * </ol> * <h2>უნომრო სია</h2> * <ul> * <li>iPhone 6S</li> * <li>Galaxy S7</li> * <li>Nexus 5X</li> * <li>Lumia 950</li> * </ul> * </body> * </html> |
| --- |



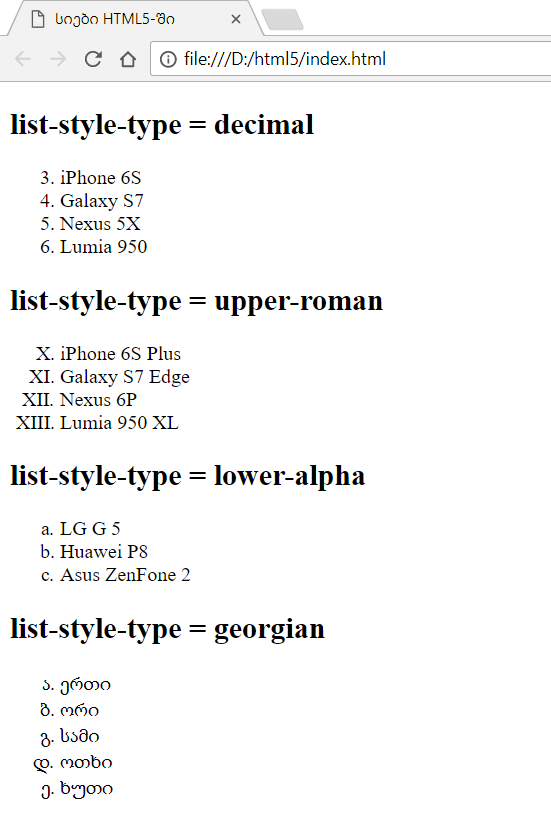
თუ სპეციალურად არ მივუთითებთ, დანომრილ სიაში ნუმერაცია ხორციელდება ნატურალური რიცხვებით 1-დან, სოლო უნომრო სიაში - შავი მუქი წერტილით. საჭიროების შემთხვევაში ჩვენ შეგვიძლია შევცვალოთ ნუმერაცია სტილის list-style-type მეშვეობით. ჩამოვთვალოთ ძირითადი და ხშირად გამოყენებული სტილები:

დანომრილი სიებისთვის:

* decimal: რიცხვები, ათვლა იწყება 1-დან
* decimal-leading-zero: რიცხვები, რომელთაც წინ უძღვის 0, მაგალითად 01,02,03,...98,99
* lower-roman: დაბალი რეგისტრის რომაული რიცხვები, მაგალითად, i, ii, iii, iv, v
* upper-roman: მაღალი რეგისტრის რომაული რიცხვები, მაგალითად, I, II, III, IV, V…
* lower-alpha: დაბალი რეგისტრის ლათინური ასოები,მაგალითად, a, b, c..., z
* upper-alpha: მაღალი რეგისტრის ლათინური ასოები,მაგალითად, A, B, C, … Z
* georgian: ქართული ანბანის ასოები, ა,ბ,გ,დ,...ჰ

დანომრილ სიებს start ატრიბუტის მეშვეობით შეიძლება მივუთითოთ, თუ საიდან დაიწყება ნუმერაცია. მაგალითად:

| * <h2>list-style-type = decimal</h2> * <ol style="list-style-type:decimal;" start="3"> * <li>iPhone 6S</li> * <li>Galaxy S7</li> * <li>Nexus 5X</li> * <li>Lumia 950</li> * </ol> * <h2>list-style-type = upper-roman</h2> * <ol style="list-style-type:upper-roman;" start="10"> * <li>iPhone 6S Plus</li> * <li>Galaxy S7 Edge</li> * <li>Nexus 6P</li> * <li>Lumia 950 XL</li> * </ol> * <h2>list-style-type = lower-alpha</h2> * <ol style="list-style-type:lower-alpha;"> * <li>LG G 5</li> * <li>Huawei P8</li> * <li>Asus ZenFone 2</li> * </ol> * <h2>list-style-type = georgian</h2> * <ol style="list-style-type:georgian;"> * <li>ერთი</li> * <li>ორი</li> * <li>სამი</li> * <li>ოთხი</li> * <li>ხუთი</li> * </ol> |
| --- |

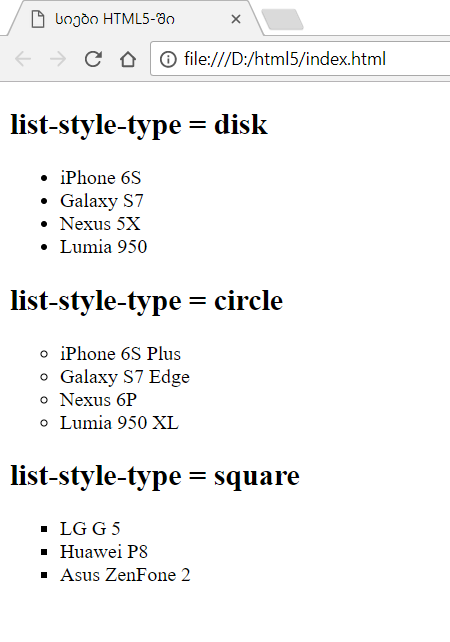


უნომრო სიებისთვის list-style-type-მა შეიძლება მიიღოს შემდეგი მნიშვნელობები:

* disk: შავი წრე
* circle: ცარიელი წრე
* square: შავი კვადრატი

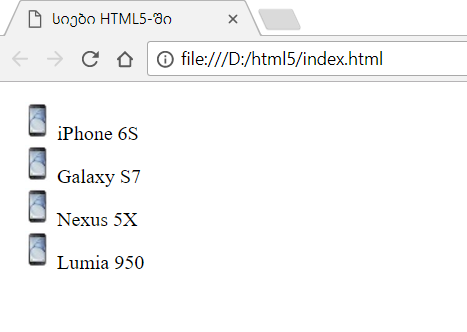
მაგალითად:

| * <h2>list-style-type = disk</h2> * <ul style="list-style-type:disk;"> * <li>iPhone 6S</li> * <li>Galaxy S7</li> * <li>Nexus 5X</li> * <li>Lumia 950</li> * </ul> * <h2>list-style-type = circle</h2> * <ul style="list-style-type:circle;"> * <li>iPhone 6S Plus</li> * <li>Galaxy S7 Edge</li> * <li>Nexus 6P</li> * <li>Lumia 950 XL</li> * </ul> * <h2>list-style-type = square</h2> * <ul style="list-style-type:square;"> * <li>LG G 5</li> * <li>Huawei P8</li> * <li>Asus ZenFone 2</li> * </ul> |
| --- |



კიდევ ერთ საინტერესო შესაძლებლობას სიებისთვის გვაძლევს სტილი list-style-image. მისი მეშვეობით შესაძლებელია სიის გვერდით გამოვიტანოთ სურათი. მაგალითად:

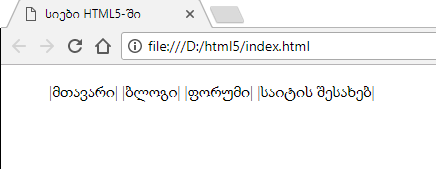
| * <ul style="list-style-image:url(phone.jpeg);"> * <li>iPhone 6S</li> * <li>Galaxy S7</li> * <li>Nexus 5X</li> * <li>Lumia 950</li> * </ul> |
| --- |



ჰორიზონტალური სია

ჰორიზონტალური სიის მისაღებად სიის ყველა ელემენტისთვის უნდა დავაყენოთ სტილი display:inline:

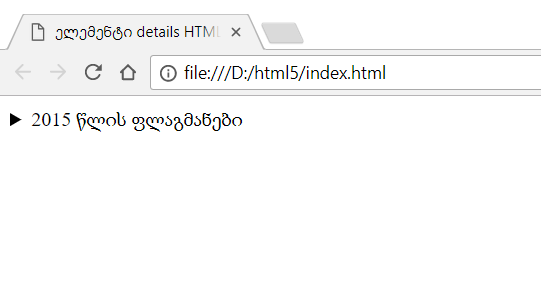
| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>სიები HTML5-ში</title> * <style> * ul#menu li { * display:inline; * } * </style> * </head> * <body> * <ul id="menu"> * <li>|მთავარი|</li> * <li>|ბლოგი|</li> * <li>|ფორუმი|</li> * <li>|საიტის შესახებ|</li> * </ul> * </body> * </body * </html> |
| --- |



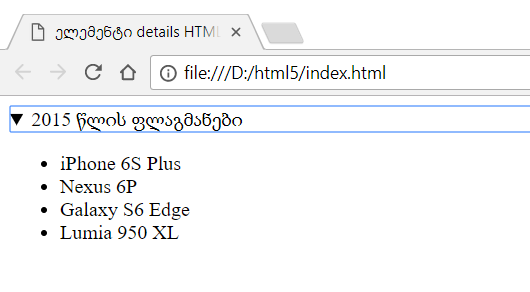
ელემენტი details

ელემენტი details საშუალებას იძლევა შეიქმნას ჩამოშლადი ბლოკი, რომელიც თავდაპირველად დამალულია. ის შეიცავს ელემენტს summary, რომელიც ყოველთვის ჩანს, როგორც სათაური, ხოლო დანარჩენი ნაწილი გამოჩნდება ისარზე ან სათაურზე დაჭერის შემთხვევაში. მაგალითად:

| * <!DOCTYPE html> * <html> * <head> * <meta charset="utf-8"> * <title>ელემენტი details HTML5-ში</title> * </head> * <body> * <details> * <summary>2015 წლის ფლაგმანები</summary> * <ul> * <li>iPhone 6S Plus</li> * <li>Nexus 6P</li> * <li>Galaxy S6 Edge</li> * <li>Lumia 950 XL</li> * </ul> * </details> * </body> * </html> |
| --- |



სათურზე ან ისარზე დაჭერისას გაიშლება დამალული ნაწილი:



უნდა გავითვალისწინოთ, რომ ამ დროისთვის ამ ელემენტის მხარდაჭერა ყველა ბრაუზერს არა აქვს.