**Napredni CSS selektori**

**00:00:00**

Jedinica: 12 od 30

**+Rezime**

U prethodnoj lekciji prikazani su osnovni CSS selektori. Lekcija pred vama nastavlja upravo tamo gde se prethodna završila, tako da donosi priču o naprednijim CSS selektorima. Neki od selektora koji će biti prikazani u ovoj lekciji imaju veoma rasprostranjenu primenu, dok se neki veoma retko koriste.

U ovoj lekciji biće prikazane sledeće grupe selektora:

* selektori atributa
* pseudoklase
* pseudoelementi

**Selektori atributa**

U dosadašnjem toku kursa, prikazano je kako se korišćenjem HTML atributa id i class može obaviti selektovanje HTML elemenata koji će biti stilizovani. Iako je najčešća praksa upotreba ova dva atributa, za selektovanje HTML elemenata mogu se koristiti i ostali atributi.

Kada je u pitanju selektovanje elemenata na osnovu atributa i njihovih vrednosti, CSS dozvoljava vrlo fleksibilan scenario selektovanja. Tabela prikazuje sve moguće načine na koje je moguće selektovati element (ili elemente) na osnovu atributa.

|  |  |
| --- | --- |
| **Selektor** | **Opis** |
| [attribute] | navedeni atribut postoji na elementu |
| [attribute="value"] | navedena vrednost je puna vrednost atributa |
| [attribute\*="value"] | navedena vrednost je deo atributa i može se naći bilo gde |
| [attribute^="value"] | navedenom vrednošću započinje vrednost atributa |
| [attribute$="value"] | navedenom vrednošću se završava vrednost atributa |
| [attribute~="value"] | vrednost atributa sadrži navedenu vrednost, ali izdvojenu praznim mestima (razmacima) |
| [attribute|="value"] | vrednost atributa počinje navedenom vrednošću; pritom je vrednost atributa podeljena dash (-) karakterima, tako da se value odnosi na početak vrednosti atributa, sve do pojave prvog dash karaktera |

*Tabela 12.1 – Selektori atributa*

**Selektovanje na osnovu postojanja atributa**

CSS dozvoljava da se element selektuje na osnovu postojanja određenog atributa, ili na osnovu specifične vrednosti koju neki atribut ima. Kada se element selektuje na osnovu postojanja nekog atributa, sintaksa selektora je kao na slici 12.1.



*Slika 12.1 – Sintaksa selektora na osnovu postojanja atributa*

Jednostavno, na regularni deo selektora, bilo da je on univerzalni, tipski, class ili id, ili se zasniva na relacijama, dodaje se deo sa uglastim zagradama.

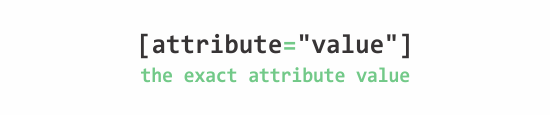
Sledeći primer ilustruje situaciju u kojoj se selektuju svi elementi koji poseduju specifični atribut:

|  |  |
| --- | --- |
| 1 | a[target] {    background-color: yellow; } |

Na prikazan način, biće selektovani svi oni a elementi koji poseduju target atribut.

**Selektovanje na osnovu tačne vrednosti atributa**

Kada je potrebno selektovati elemente na osnovu vrednosti nekog atributa, sintaksa selektora je kao na slici 12.2.



*Slika 12.2 – Sintaksa selektora na osnovu vrednosti atributa*

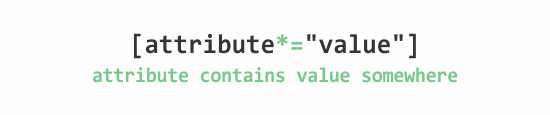
U uglastoj zagradi se, pored naziva atributa, navodi i njegova vrednost.  
Sledeći primer ilustruje upotrebu ovakve vrste selektora:

|  |  |
| --- | --- |
| 1 | a[target="\_blank"] {    background-color: yellow; } |

Na ovaj način, biće selektovani svi a elementi, čija je vrednost atributa target, *\_blank.*

**Selektovanje na osnovu dela vrednosti atributa koji može biti bilo gde**

CSS omogućava selektovanje elemenata na osnovu dela vrednosti atributa. Pritom, taj deo vrednosti se može naći bilo gde u okviru same vrednosti. Sintaksa ovakvog selektora je prikazana slikom 12.3.



*Slika 12.3 – Sintaksa selektora na osnovu dela vrednosti atributa*

Primer ovakvog selektora je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | [class\*="wrap"] {      background: yellow;  } |

Na ovaj način biće selektovani svi elementi koji u nazivu klase imaju sekvencu karaktera wrap, bilo gde. Element koji bi na ovaj način bio selektovan je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | <div class="wrapper">  …  </div> |

**Selektovanje na osnovu teksta kojim započinje vrednost atributa**

CSS omogućava selektovanje elemenata na osnovu teksta kojim započinje vrednost određenog atributa. Sintaksa ovakvog selektora prikazana je na slici 12.4.



*Slika 12.4 – Sintaksa selektora na osnovu teksta kojim započinje vrednost atributa*

Primer upotrebe ovakvog selektora jeste selektovanje svih linkova koji upućuju na jedan isti sajt, odnosno top domen, bez obzira na konkretnu stranicu takvog sajta. Takvi linkovi mogu da izgledaju ovako:

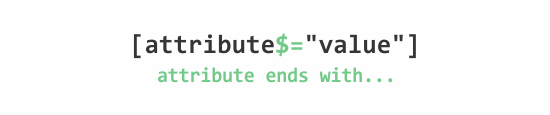
|  |  |
| --- | --- |
| 1  2  3 | <a href="http://www.link-group.eu">Link Group</a>  <a href="http://www.link-group.eu/careers">Link Careers</a>  <a href="http://www.link-group.eu/about-us-0">About Link</a> |

Sva tri prikazana linka imaju identičan top domen: [http://www.link-group.eu](http://www.link-group.eu/). Ipak, konkretne stranice se razlikuju. Ukoliko je potrebno selektovati sve linkove čiji href atribut započinje sa [http://www.link-group.eu](http://www.link-group.eu/), može se napisati selektor koji izgleda ovako:

|  |  |
| --- | --- |
| 1  2  3 | a[href^="http://www.link-group.eu"] {   color: red;  } |

**Selektovanje na osnovu teksta kojim se završava vrednost atributa**

Slično kao u prethodnom primeru, CSS omogućava da se element selektuje na osnovu teksta kojim se vrednost atributa završava. Sintaksa takvog selektora prikazana je slikom 12.5.



*Slika 12.5 – Sintaksa selektora na osnovu teksta kojim se završava vrednost atributa*

Odličan primer upotrebe ovakvog selektora jeste selektovanje download linkova na osnovu ekstenzije fajla koja stoji iza takvog linka. Primer selektora koji će selektovati samo linkove koji upućuju na pdffajlove je sledeći:

|  |  |
| --- | --- |
| 1  2  3  4 | a[href$=".pdf"] {  background: url(icon-pdf.png) left center no-repeat;  padding-left: 30px;  } |

Link koji će na ovakav način biti selektovan, može da izgleda ovako:

|  |  |
| --- | --- |
| 1 | <a href="http://www.demo.com/documents/demo.pdf">Download pdf document</a> |

**Selektovanje na osnovu dela vrednosti atributa koji je izdvojen praznim mestima**

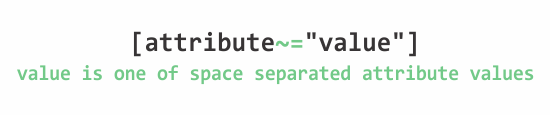
Vrlo česta praksa je postavljanje više naziva klasa na jedan element. Primer takvog elementa je sledeći:

|  |  |
| --- | --- |
| 1 | <h1 class="heading animate fade">This is document heading</h1> |

Ukoliko je potrebno selektovati element po nazivu samo jedne klase, class selektor je sasvim dovoljan, pošto omogućava selektovanje po pojedinačnim nazivima klasa. Ipak, neki drugi atributi i njihovi selektori tako nešto ne omogućavaju. Razmotrimo sledeći primer:

|  |  |
| --- | --- |
| 1 | <h1 rel="main external animated">This is heading</h1> |

Kako bi se selektovao element na osnovu samo jedne od vrednosti atributa, koja je razdvojena od susednih razmacima, potrebno je koristiti selektor čija sintaksa je prikazana slikom 12.6.



*Slika 12.6 – Sintaksa selektora na osnovu dela vrednosti atributa koji je razdvojen razmacima od ostatka vrednosti*

Tako se selektor može formirati na sledeći način:

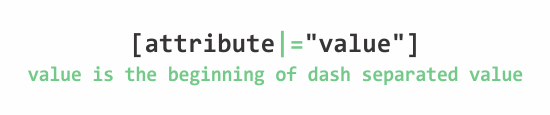
|  |  |
| --- | --- |
| 1  2  3 | h1[rel~="external"] {  color: red;  } |

**Selektovanje na osnovu početka vrednosti atributa, kada su segmenti vrednosti razdvojeni dash karakterom**

Česta je praksa da se u situacijama kada se vrednosti atributa sastoje iz više reči, takve reči razdvajaju karakterom dash (-). Primer takvih vrednosti atributa je sledeći:

|  |  |
| --- | --- |
| 1  2 | <h1 class="top-header">Welcome</h1>  <p class="top-text">Hello world!</p> |

CSS poseduje selektor koji će selektovati element na osnovu početka vrednosti atributa. Ipak, za razliku od nešto ranije pomenutog selektora slične namene, u obzir se uzima početak vrednosti, sve do karaktera dash (-). Selektor sa opisanim ponašanjem ima sintaksu prikazanu slikom 12.7.



*Slika 12.7 – Sintaksa selektora na osnovu početka vrednosti atributa, sve do prvog karaktera dash*

Primer ovakvog selektora je sledeći:

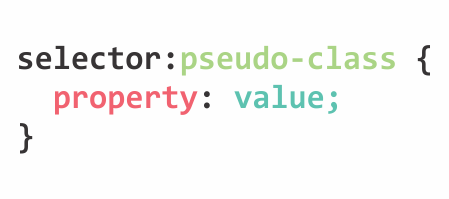
|  |  |
| --- | --- |
| 1 | [class|="top"] {    background: yellow; } |

Ovakav selektor selektovaće elemente koji imaju atribut class sa vrednošću koja počinje tekstom top. Nakon teksta top, neophodno je da usledi karakter dash (-). Takođe, ovakav selektor odnosi se i na elemente koji imaju naziv klase top.

**Pseudoklase**

Pseudoklase su selektori koji se koriste da definišu ponašanje elementa u specijalnim situacijama. Pod specijalnim situacijama misli se na prelazak strelice miša preko elementa, dobijanje fokusa, klik i slično.

Pseudoklase se karakterišu po tome što se dodaju na selektor, pri čemu pseudoklasa započinje karakterom dvotačka (:). Drugim rečima, pseudoklasa je ključna reč koja se dodaje na selektor i na taj način definiše neko specijalno stanje selektovanog elementa. Sintaksa pseudoklasa prikazana je na slici.



*Slika 12.8 – Sintaksa pseudo-class selektora*

U zavisnosti od mesta upotrebe, sve pseudoklase grubo se mogu podeliti na dve kategorije:

* pseudoklase različitih stanja UI elementa
* strukturalne pseudoklase

**Pseudoklase različitih stanja UI elementa**

CSS poseduje grupu selektora koji omogućavaju da se stilizuju različita stanja u kojima se mogu naći HTML elementi. Pseudoklase ove grupe najčešće se koriste za stilizovanje linkova i input kontrola, ali i ostalih HTML elemenata. Tabela 12.2 prikazuje pseudoklase različitih stanja UI elemenata.

|  |  |
| --- | --- |
| **Pseudoklasa** | **Opis** |
| :link | selektuje linkove koji još nisu posećeni |
| :visited | selektuje linkove koji su posećeni |
| :hover | selektuje element u situaciji kada je strelica miša iznad elementa |
| :active | selektuje element koji je aktiviran od strane korisnika |
| :focus | selektuje element koji ima fokus |
| :enabled | selektuje elemente koji nisu onemogućeni |
| :disabled | selektuje onemogućene elemente |
| :checked | selektuje čekirane checkbox elemente |

*Tabela 12.2 – Pseudoklase stanja*

Veoma česta praksa jeste korišćenje pseudoklasa za stilizovane različitih stanja a elemenata. Sledeći primer ilustruje takvu situaciju:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | /\* unvisited link \*/  a:link {      color: red;  }    /\* visited link \*/  a:visited {      color: green;  }    /\* mouse over link \*/  a:hover {      color: blue;  }    /\* selected link \*/  a:active {      color: yellow;  } |

U primeru su definisani CSS opisi korišćenjem pseudoklasa :link, :visited, :hover i :active.

Pseudoklasa :link stilizuje link elemente koji još nisu posećeni. S razlogom se može postaviti pitanje opravdanosti korišćenja ove pseudoklase za selektovanje link elementa, s obzirom da identičan posao obavlja i tipski selektor a. Pa, koja je onda poenta korišćenja ove pseudoklase? Tipski selektor a selektovaće sve a elemente, bez obzira da li oni imaju definisan href atribut. Tako nešto nije slučaj prilikom korišćenja :link pseudoklase, koja selektuje samo one link elemente koji imaju definisan ovaj atribut.

Pseudoklasa :link igra značajnu ulogu i u hijerarhiji ostalih pseudoklasa za stilizovanje linkova. Naime, pseudoklase za stilizovanje linkova moraju se navesti baš onim redom kojim su navedene u prikazanom primeru. To znači da se prvo mora navesti :link selektor, zatim :visited, pa :hover i na kraju :active.

Određene pseudoklase, prikazane prethodnim primerom, nisu ograničene na linkove. Tako je vrlo česta praksa korišćenje :hover pseudoklase na bilo kojem drugom elementu. Tako nešto ilustruje sledeći primer:

|  |  |
| --- | --- |
| 1 | div:hover {    background-color: blue; } |

Prelaskom miša preko div elementa, njegova boja pozadine postaje plava.

U prethodnom primeru, korišćena je pseudoklasa :active za stilizovanje stanja u kojem je link element aktiviran od strane korisnika. Pod aktiviranjem link elementa smatra se trenutak kada korisnik izvrši klik na takav link element. Stoga link element dobija žutu boju, samo dok traje klik na njega. Drugim rečima, link u prikazanom primeru biće žute boje od trenutka kada korisnik pritisne taster miša iznad link elementa, pa sve dok ga ne otpusti.

Pseudoklasa koja se često izjednačuje sa :active pseudoklasom jeste klasa :focus. Ipak, ove dve pseudoklase imaju potpuno različitu funkciju. Pseudoklasa :focus selektuje element koji je trenutno selektovan na stranici i na taj način spreman da prihvati unos sa ulaznog uređaja računara, što je najčešće tastatura. Sledeći primer ilustruje CSS opis sa selektorom :focus:

|  |  |
| --- | --- |
| 1  2  3 | a:focus {  color: black;  } |

Kada se koristi za stilizovanje linkova, :focus je potrebno navesti pre :active pseudoklase, kako bi i efekat :active selektora došao do izražaja.

|  |
| --- |
| **Napomena**    *U prilogu lekcije nalazi se kompletan primer stilizovanja link HTML elemenata.* |

Pseudoklasa :focus se često koristi za stilizovanje input HTML kontrola. Slika 12.9 prikazuje jednu HTML formu, stilizovanu između ostalog i korišćenjem pseudoklasa.



*Slika 12.9 – Forma sa input kontrolama koje poseduju različite pseudoklase*

Na slici 12.9 se može zapaziti nekoliko stvari. Fokus poseduje e-mail input kontrola (tamnija boja okvira). Strelica miša se nalazi iznad name input kontrole. Subject input kontrola je onemogućena, što znači da nije moguće unositi tekst u nju. Tekst pored checkbox input kontrole obojen je u plavo.

|  |
| --- |
| **Napomena**    *U prilogu lekcije nalazi se i primer koji je upravo prikazan, u kome se stilizuju različita stanja input elemenata. Svakako je najbolje otvoriti HTML dokument koji se nalazi u prilogu i proučiti različita stanja u kojima se HTML elementi mogu naći i njihov izgled u takvim situacijama.* |

HTML kod kojim se kreira prikazana forma sa input elementima je sledeći:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <div id="contact-wrapper">    <form>     <input type="text" name="name" id="name" placeholder="NAME">     <input type="text" name="email" id="email" placeholder="E-MAIL">     <input disabled type="text" name="subject" id="subject" placeholder="SUBJECT (disabled)">     <input type="submit" name="submit" id="submit" value="SEND">     <input type="checkbox" name="subscribe" id="subscribe">     <label for="subscribe">YES, I WANT TO RECEIVE INFORMATION ABOUT NEW PROJECTS</label>    </form>   </div> |

Za text input kontrole definisani su sledeći CSS opisi:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | /\*STYLES FOR INPUT TEXT FIELDS\*/  input[type="text"] {  width: 100%;  padding: 10px;  border: solid 2px #c9c9c9;  margin: 10px 0px 10px 0px;  }    input[type="text"]:hover{  border: solid 2px #969696;  }    input[type="text"]:focus{  border: solid 2px #262626;  }    input[type="text"]:disabled {  background-color: rgba(201, 201, 201, 0.75);  } |

Može se primetiti da se za selektovanje koristi kombinacija selektora zasnovanih na atributima i pseudoklasama. Za text input elemente definisana su tri specijalna stanja korišćenjem pseudoklasa: hover, focus i disabled.

Kao i kod linkova, pseudoklasa :hover selektovaće element u situacijama kada se strelica miša nalazi iznad elementa. Upravo zbog toga se boja okvira input text kontrole menja kada se preko nje pređe strelicom miša.

Pseudoklasa :focus selektuje kontrolu koja ima fokus na stranici. Kontrola može da dobije fokus klikom, ili korišćenjem tastature (npr. tastera tab). Na taj način će input kontrola koja je u fokusu imati tamniju boju okvira.

Na kraju, korišćenjem :disabled pseudoklase definisana je boja pozadine za onemogućene kontrole. Onemogućene kontrole su one koje imaju postavljen atribut disabled. S obzirom da je u HTML kodu input text kontrola za unos subjecta onemogućena korišćenjem pomenutog atributa, ovako definisana stilizacija primenjivaće se na ovoj kontroli. Upravo zbog toga, input kontrola za unos subjecta ima tamniju boju pozadine od ostalih.

I za input submit element definisana je stilizacija korišćenjem nekoliko pseudoklasa:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | /\*STYLES FOR INPUT SUBMIT ELEMENT\*/  input[type="submit"] {  padding: 10px;  border: solid 2px #c9c9c9;  background-color: rgba(201, 201, 201, 0);  display: block;  margin-bottom: 10px;  }    input[type="submit"]:hover {  border: solid 2px #969696;  }    input[type="submit"]:focus{  border: solid 2px #262626;  }    input[type="submit"]:active {  border: solid 2px #41B9E1;  } |

Za *hvatanje* specijalnih stanja submit input kontrole, prilikom definisanja CSS opisa, iskorišćene su :hover, :focus i :active pseudoklase.

Pseudoklasa :hover ima identičan efekat kao i u svim prethodnim primerima, dok upotreba :focus i :active pseudoklasa može biti zbunjujuća. Kada input submit element dobije fokus, aktivira se CSS opis sa selektorom :focus. Ipak, ukoliko dobijanje fokusa za sobom povlači i aktiviranje submit input elementa, prednost će imati CSS opis sa :active pseudoklasom, jer je naveden nakon CSS opisa sa pseudoklasom :focus. Ukoliko korisnik dodeli fokus input submit elementu korišćenjem tastature, takav element dobiće okvire tamnije boje. Ukoliko korisnik izvrši klik na submit input element, aktiviraće CSS opis sa :active pseudoklasom.

Na kraju, u primeru stilizovanja specijalnih stanja input kontrola, naveden je i jedan CSS opis koji se tiče input checkbox kontrole. To je sledeći opis:

|  |  |
| --- | --- |
| 1  2  3  4 | /\*STYLES FOR INPUT CHECKBOX ELEMENT\*/  input[type="checkbox"]:checked + label {  color: #41B9E1;  } |

Opis sadrži pseudoklasu :checked, čime se CSS opis primenjuje samo u situaciji kada je checkbox element čekiran. Pritom, u ovom primeru odlazimo i korak dalje, pa vršimo stilizaciju label elementa, onda kada je checkbox element čekiran. Drugim rečima, CSS omogućava da se prilikom ulaska elementa u neko od specijalnih stanja, stilizuje neki drugi HTML element. Još jednostavnije rečeno: kada se checkbox kontrola čekira, boja teksta label elementa pored boji se u plavo.

|  |
| --- |
| **Napomena**    *U prilogu lekcije nalazi se kompletan primer stilizovanja input HTML elemenata forme.* |

**Koja pseudoklasa se koristi kako bi se uhvatilo stanje prelaska strelice miša preko elementa:1**

 :hover

 :focus

 :disabled

 :active

**Strukturalne pseudoklase**

CSS omogućava selektovanje elemenata na osnovu strukture koju oni međusobno grade. Reč je o veoma korisnoj grupi strukturalnih pseudoklasa, koje su prikazane tabelom 12.3.

|  |  |
| --- | --- |
| **Pseudoklasa** | **Opis** |
| :first-child | samo elementi koji su prvi potomci svog roditelja |
| :last-child | samo elementi koji su poslednji potomci svoj roditelja |
| :first-of-type | samo elementi koji su prvi potomci određenog tipa |
| :last-of-type | samo elementi koji su poslednji potomci određenog tipa |
| :only-child | samo elementi koji su jedini potomci svog roditelja |
| :only-of-type | samo elementi koji su jedini potomci svoje vrste |
| :nth-child(n) | selektuje sve elemente koji su n-ti potomci svog roditelja |
| :nth-last-child(n) | selektuje sve elemente koji su n-ti potomci svog roditelja od pozadi |
| :nth-of-type(n) | selektuje sve elemente koji su n-ti potomci svog tipa |
| :nth-last-of-type(n) | selektuje sve elemente koji su n-ti potomci svog tipa od pozadi |

*Tabela 12.3 – Strukturalne pseudoklase*

**:first-child**

Za seletovanje elemenata koji su isključivo prvi potomci svog roditelja, može se koristiti pseudoklasa :first-child. Slika 12.10 ilustruje upotrebu ovog selektora.



*Slika 12.10 – Selektovanje paragrafa koji su prvi potomci svog roditelja*

Kao što se sa slike 12.10 može videti, selektuju se svi paragrafi koji su prvi potomci svog roditelja. To je u prikazanom primeru samo *Paragraf 3.*

**:last-child**

Suprotan efekat upravo prikazanom selektoru, postiže se upotrebom pseudoklase: :last-child. Slika 12.11 ilustruje upotrebu ovog selektora.



*Slika 12.11 – Selektovanje paragrafa koji su poslednji potomci svog roditelja*

Kao što se na slici 12.11 može videti, selektuju se samo paragrafi koji su poslednji potomci svog roditelja. Paragraf 3 je ujedno i prvi i poslednji potomak svog roditelja, s obzirom da je jedini element unutar *second-div* div elementa. *Paragraf 5* je poslednji potomak body elementa.

**:first-of-type**

Već je prikazano kako pseudoklasa :first-child omogućava selektovanje elemenata koji su prvi potomci svog roditelja. Ukoliko je potrebno selektovati elemente koji su prvi potomci određenog tipa, koristi se pseudoklasa :first-of-type. Primer na slici 12.12 ilustruje upotrebu ovog selektora.



*Slika 12.12 – Selektovanje paragrafa koji su prvi potomci svog roditelja, svog tipa*

Na slici 12.12 se može videti da se selektuju svi paragraf elementi koji su prvi potomci svog tipa unutar svojih roditelja. Unutar body elementa, prvi potomak p tipa je *Paragraf 0*. Unutar first-divelementa, prvi potomak p tipa je *Paragraf 1*. Na kraju, *Paragraf 3* je prvi i poslednji i jedini potomak second-div elementa.

**:last-of-type**

Suprotan efekat upravo prikazanom selektoru ima pseudoklasa :last-of-type. Efekat ove pseudoklase prikazan je slikom 12.13.



*Slika 12.13 – Selektovanje paragrafa koji su poslednji potomci svog roditelja, svog tipa*

Ovoga puta selektuju se svi p elementi koji su poslednji potomci svog tipa. Unutar body elementa, poslednji paragraf element je *Paragraf 5*. Unutar second-div elementa, prvi i poslednji paragraf je *Paragraf 3*, dok je unutar first-div elementa, poslednji paragraf element *Paragraf 2*. Na primeru *Paragraf 2* elementa, može se videti razlika između :last-child i :last-of-type selektora.

**:only-child**

Selektor :only-child selektuje element ukoliko je on jedini potomak svog roditelja. Sledeći primer, prikazan slikom 12.14, ilustruje takvu situaciju.



*Slika 12.14 – Selektovanje paragrafa koji su jedini potomci svog roditelja*

Slika 12.14 prikazuje upotrebu :only-child selektora, za selektovanje svih onih paragraf elemenata koji su jedini potomci svog roditelja. U prikazanom primeru, takve osobine ima samo paragraf 3, s obzirom da jedino *second-div* poseduje samo jednog potomka.

**:only-of-type**

Slično prethodnom selektoru, :only-of-type pseudoklasa omogućava da se selektuje element ukoliko je on jedini potomak svog tipa. Primer je ilustrovan slikom 12.15.



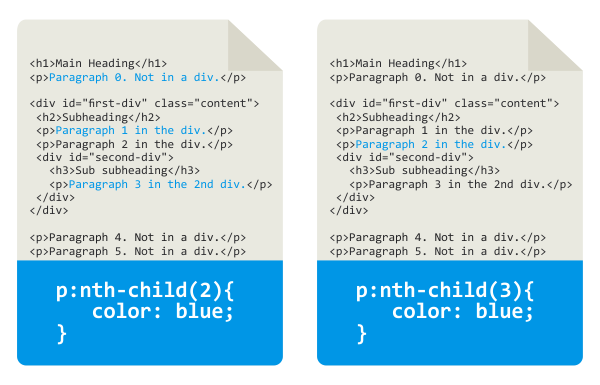
*Slika 12.15 – Selektovanje paragrafa koji su jedini potomci svog roditelja, svog tipa*

Svi kontejnerski elementi sadrže više od jednog paragrafa. Jedino second-div element poseduje tačno jedan paragraf element. Pritom, za razliku od prethodnog primera, *Paragraf 3* nije jedini potomak svog roditelja, ali jeste jedini potomak svoje vrste, te stoga biva selektovan pseudoklasom :only-of-type. Da je kojim slučajem u ovom primeru iskorišćen selektor :only-child, ne bi bio selektovan nijedan element.

**:nth-child(n)**

Do sada su prikazani selektori koji su omogućavali selektovanje prvih, poslednjih i jedinih elemenata u strukturi HTML dokumenta. CSS nije ograničen na selektovanje samo prvih i poslednjih elemenata, već poseduje skup selektora, koji omogućavaju da se elementi pogode po proizvoljnoj logici. Jedan od takvih selektora je i :nth-child(n).

Pseudoklasa :nth-child(n) omogućava selektovanje elementa koji je n-ti potomak svog roditelja. Primer na slici 12.16 ilustruje upotrebu ovog selektora.



*Slika 12.16 – Selektovanje paragrafa koji su drugi i treći potomci svojih roditelja*

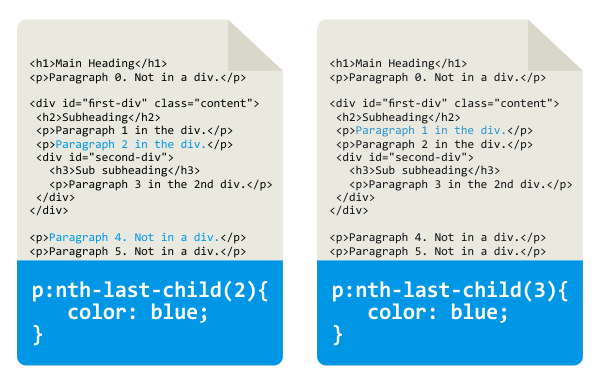
U primeru sa leve strane slike 12.16, selektuje se svaki onaj paragraf koji je u hijerarhiji drugi potomak svog roditelja. *Paragraf 0* to jeste, takođe *Paragraf 1 i Paragraf 3.*

Na desnoj polovini slike, kao vrednost promenljive n postavljeno je 3, što znači da će biti selektovan samo onaj paragraf koji je treći potomak svog roditelja. Jedini paragraf koji zadovoljava takav kriterijum je *Paragraf 2*, s obzirom da je treći potomak, odmah iza *Subheading i Paragraf 1*elemenata.

|  |
| --- |
| **Napomena**    Promenljiva n kod selektora :nth-child(n)(ali i svih drugih selektora koji poseduju promenljivu u svojoj sintaksi, a čiji opis sledi), nije ograničena na celobrojnu vrednost. Naime, vrednost promenljive n može biti ceo broj, ključna reč ili formula. CSS pseudoklase mogu da na mestu promenljive prihvate dve ključne reči: odd i even. Na taj način omogućava se selektovanje svih elemenata koji imaju neparan, odnosno paran indeks, respektivno. Slika 12.17 to ilustruje.    https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_12_17.jpg  *Slika 12.17 – Selektovanje svih parnih i neparnih paragrafa*    Pored celih brojeva, ključnih reči even i odd, pseudoklase na mestu promenljive mogu da prihvate i formulu. Formula ima sledeći oblik (an + b), gde je a množilac, n brojač, a b offset. Brojač počinje od 0. Slika 12.18 ilustruje definisanje izraza korišćenjem navedene formule na mestu promenljive.    https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_12_18.jpg  *Slika 12.18 – Korišćenje formula za selektovanje unutar pseudoklase*    U primeru koji je prikazan na levoj strani slike 12.18, definisan je sledeći izraz:    (2n+1)  Na ovaj način, biće selektvani svi p elementi koji imaju sledeće redne brojeve u hijerarhiji unutar svojih roditeljskih elemenata:    (2\*0+1)=1 (2\*1+1)=3 (2\*2+1)=5 (2\*3+1)=7 ...  Početna vrednost brojača je nula, a u svakoj iteraciji, brojač se uvećava za jedan. Tako se dobijaju rezultati prikazani iznad. Može se primetiti da se korišćenjem formule (2n+1) postiže efekat identičan ključnoj reči odd, jer se selektuju svi neparni elementi.    S desne strane slike 12.18, definisana je pseudoklasa :nth-child sa formulom (2n+2). Na ovaj način, biće selektovani svi p elementi sa sledećim rednim brojevima:    (2\*0+2)=2 (2\*1+2)=4 (2\*2+2)=6 (2\*3+2)=8 ...  Može se primetiti da formula (2n+2) ima identičan efekat ključnoj reči even, jer selektuje sve parne elemente. |

**:nth-last-child(n)**

Selektor :nth-last-child(n) ima identičan efekat prethodno opisanom, s tim što elemente broji od pozadi. Primeri upotrebe ovog selektora prikazani su slikom 12.19.



*Slika 12.19 – Selektovanje paragrafa koji su drugi i treći elementi od pozadi unutar svojih roditelja*

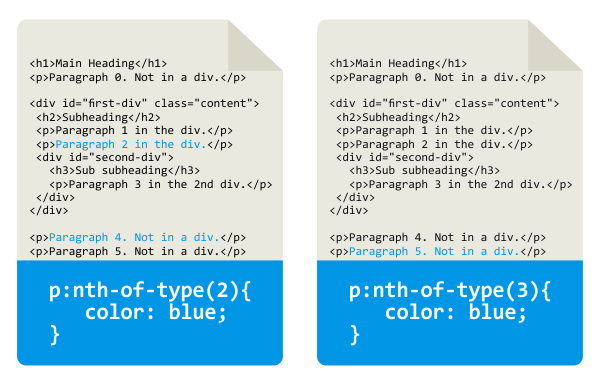
U primeru sa leve strane slike 12.19, selektuje se svaki onaj paragraf koji je po redosledu drugi potomak od pozadi, unutar svog roditeljskog kontejnera. *Paragraf 4* to svakako jeste, s obzirom da pre njega, od pozadi, postoji samo*Paragraf 5* element. Takođe, naveden uslov zadovoljava i element *Paragraf 2*, s obzirom da unutar*first-div elementa*, pre njega, od pozadi, postoji samo *second-div element*.

Slična je situacija i u primeru sa desne strane slike 12.19. Selektuje se svaki paragraf element koji je treći potomak svog roditelja od pozadi. Jedini paragraf element sa takvim osobinama je Paragraf 1, s obzirom da, u hijerarhiji elementa koji su potomci first-div elementa, pre njega, postoje second-div i Paragraf 2.

Pseudoklasa :nth-last-child(n) takođe može da prihvati celobrojnu vrednost, ključnu reč even ili odd ili formulu, baš kao što je bio slučaj i sa selektorom :nth-child(n).

**:nth-of-type(n)**

Selektor :nth-of-type(n) je praktično identičan selektoru :nth-child(n) s tim što selektuje sve n-te elemente, ali isključivo određenog tipa. Slika 12.20 ilustruje upotrebu ovog selektora.



*Slika 12.20 – Selektovanje paragrafa koji su drugi i treći elementi svog tipa, unutar svojih roditelja*

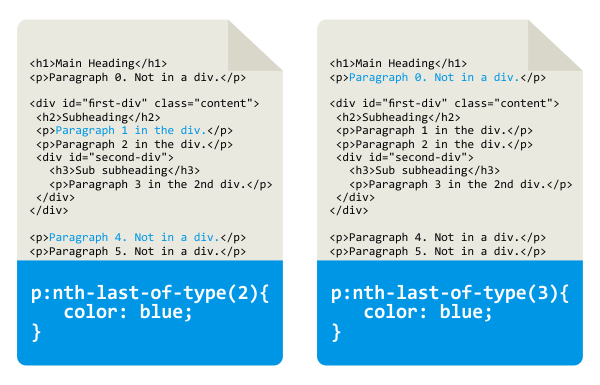
Selektor sa leve strane slike 12.20, selektuje sve paragraf elemente koji su drugi paragraf potomci svog roditelja. *Paragraf 2* to jeste, s obzirom da je *Paragraf 1* prvi potomak.

Sa desne strane slike 12.20, vrši se selektovanje svih onih paragrafa koji su treći paragraf potomci unutar svog roditelja. Jedini element koji zadovoljava takav kriterijum je *Paragraf 5* element. Pre njega unutar istog roditelja nalaze se *Paragraf 0* i *Paragraf 4* elementi, tako da je jasno da je *Paragraf 5*treći p element unutar svog roditelja.

|  |
| --- |
| **Napomena**    *Selektor :nth-of-type(n), baš kao i prethodno prikazani selektori, može da prihvati parametar u obliku celobrojne vrednosti, ključne reči ili formule.* |

**:nth-last-of-type(n)**

Selektor :nth-last-of-type(n) selektuje sve one elemente koji su n-ti potomci svoje vrste, ali gledano od pozadi. Slika 12.21. ilustruje upotrebu ovog selektora.



*Slika 12.21 – Selektovanje paragrafa koji su drugi i treći elementi od pozadi, svog tipa, unutar svojih roditelja*

Na levoj polovini slike 12.21 ilustrovano je selektovanje svakog paragraf elementa koji je drugi od pozadi paragraf element unutar svog roditelja. *Paragraf 4* je drugi od pozadi paragraf, jer pre njega postoji *Paragraf 5*. Takođe, *Paragraf 1* je drugi od pozadi unutar svog roditelja, jer pre njega postoji samo *Paragraf 2* element.

Sa desne strane slike 12.21, u primeru je povećana vrednost parametra na 3. Na taj način selektuje se svaki paragraf element koji je treći potomak svoje vrste od pozadi. Takav kriterijum ispunjava jedino element Paragraf 0. Pre njega, unutar istog kontejnera, postoje p elementi *Paragraf 5* i *Paragraf 4.*

|  |
| --- |
| **Napomena**    *Selektor :nth-last-of-type(n), baš kao i prethodno prikazani selektori, može da prihvati parametar u obliku celobrojne vrednosti, ključne reči ili formule*. |

**Pseudoelementi**

U dosadašnjem toku lekcije, prikazane su različite pseudoklase. Pseudoklase se koriste za selektovanje *regularnih* HTML elemenata koji se mogu naći u specijalnim stanjima ili koji zadovoljavaju određene uslove. Kaže se *regularni* HTML elementi, zato što takvi elementi postoje u HTML strukturi dokumenta.

Za razliku od pseudoklasa, pseudoelementi kreiraju nove elemente koji nisu navedeni u HTML strukturi dokumenta. Na taj način, moguće je postići različite zanimljive efekte kada je u pitanju prezentaciona logika HTML dokumenta. S pojavom CSS3 verzije jezika, razlika između pseudoklasa i pseudoelemenata još je izraženija i jasnija, pošto se za pisanje pseudoelemenata koriste dva karaktera dvotačka (::). U CSS3 jeziku postoje sledeći pseudoelementi:

* ::after (:after)
* ::before (:before)
* ::first-letter (:first-letter)
* ::first-line (:first-line)
* ::selection

**::after (:after)**

U CSS-u, pseudoelement ::after selektuje poslednjeg virtualnog potomka nad selektovanim elementima. Drugim rečima, pseudoelement ::afterdodaje određeni sadržaj nakon selektovanog elementa. Sintaksa ovog pseudoelementa je sledeća:

|  |  |
| --- | --- |
| 1  2  3  4  5 | /\* CSS3 syntax \*/  ::after { style properties }    /\* CSS2 syntax \*/  :after { style properties } |

Može se primetiti da je razlika u sintaksi između CSS2 i CSS3 jezika isključivo u upotrebi duplog karaktera dvotačka.

|  |
| --- |
| **Napomena**    *Stariji browseri, pogotovo Internet Explorer ispod verzije 9, podržavaju isključivo staru sintaksu za pisanje pseudoelemenata.* |

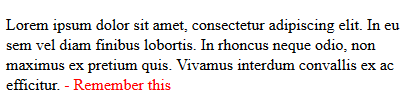
Sledeći primer će ilustrovati upotrebu ::after pseudoelementa. HTML kod je:

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non maximus ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>

Da bi se neki sadržaj dodao nakon prikazanog paragrafa, moguće je napisati:

|  |  |
| --- | --- |
| 1  2  3  4 | p::after {  content: " - Remember this";  color: red;  } |

Na ovaj način, iskorišćen je pseudoelement ::after za dodavanje sadržaja nakon svih paragrafa na stranici. Sadržaj koji se na ovaj način dodaje definiše se korišćenjem svojstva content. Takođe, dodati sadržaj je moguće stilizovati po želji, a u primeru je boja teksta postavljena na crvenu. Prikazani CSS opis stvoriće efekat kao na slici 12.22.



*Slika 12.22 – Sadržaj koji je dodat nakon paragraf elementa korišćenjem ::after pseudoelementa*

|  |  |  |
| --- | --- | --- |
| **Primer upotrebe ::after pseudoelementa**  *Pseudoelement ::after često se praktično koristi za kreiranje tooltipa. Sledeći primer će to ilustrovati. HTML struktura sastoji se iz jednog paragrafa. Paragraf izgleda ovako:*    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non <span data-descr="Lorem ipsum dolor sit amet">maximus</span> ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>    *Može se primetiti da se unutar paragrafa nalazi jedan span element, koji obuhvata reč maximus. Span element poseduje i jedan*[*custom data atribut*](https://www.link-elearning.com/linkdl/opisPojma.php?id=144347)*. Vrednost ovoga atributa biće iskorišćena kao tekst tooltipa. Tooltip je potrebno da se pojavi prilikom prelaska strelice miša preko span elementa. Da bi se sve navedeno postiglo, potrebno je definisati sledeći CSS:*     |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | span[data-descr] {  position: relative;  text-decoration: underline;  color: #00F;  cursor: help;  }  span[data-descr]:hover::after {  content: attr(data-descr);  position: absolute;  left: 0;  top: 24px;  min-width: 200px;  border: 1px #aaaaaa solid;  background-color: #4D4D4D;  padding: 12px;  color: white;  font-size: 14px;  z-index: 1;  } |   *Prvo su stilizovani svi span elementi sa atributom data-descr, a zatim je ::after pseudoelement dodat na span samo u specijalnoj situaciji kada dođe do hovera. Na taj način, tooltip koji je definisan kroz ::after pseudoelement pojaviće se samo u situacijama kada se strelica miša nađe iznad span elementa koji ima atribut data-descr. Slika 12.23 ilustruje efekat prikazanog koda za kreiranje tooltipa.*    *https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_12_23.jpg*    *Slika 12.23 – Kreiranje tooltipa korišćenjem ::after pseudoelementa*    *U prilogu ove lekcije nalazi se html fajl sa kompletnim primerom kreiranja tooltipa*. |

**::before (:before)**

Pseudoelement ::before dodaje određeni sadržaj pre selektovanog elementa. Drugim rečima, pseudoelement ::before ima efekat identičan pseudoelementu ::after, s tim što je razlika u poziciji na koju se dodaje sadržaj.

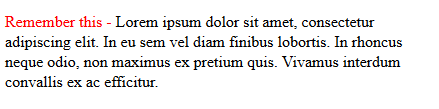
Sledeći primer ilustrovaće upotrebu pseudoelementa ::before. HTML kod je sledeći:

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non maximus ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>

Za dodavanje sadržaja pre prikazanog paragrafa, korišćenjem pseudoelementa moguće je napisati:

|  |  |
| --- | --- |
| 1  2  3  4 | p::before {  content: "Remember this - ";  color: red;  } |

Sve je praktično identično primeru koji ilustruje upotrebu pseudoelementa ::after. Prikazani kod proizvodi efekat kao na slici 12.24.



*Slika 12.24 – Sadržaj koji je dodat pre paragraf elementa korišćenjem ::before pseudoelementa*

**::first-letter (:first-letter)**

Pseudoelement ::first-letter može se koristiti za stilizovanje prvog slova selektovanih elemenata. Sintaksa ovog selektora je sledeća:

|  |  |
| --- | --- |
| 1  2  3  4  5 | /\* CSS3 syntax \*/  ::first-letter { style properties }    /\* CSS2 syntax \*/  :first-letter { style properties } |

Kao i kod ostalih pseudoelemenata, razlika između CSS2 i CSS3 sintakse se ogleda u dve dvotačke za CSS3 sintaksu. Sledeći primer ilustrovaće upotrebu ::first-letter selektora. HTML kod je sledeći:

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non maximus ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>

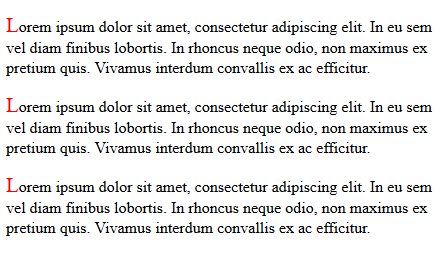
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non maximus ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non maximus ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>

Kako bi se početno slovo svakog paragrafa istaklo povećanjem veličine slova i promenom boje, moguće je napisati nešto ovako:

|  |  |
| --- | --- |
| 1  2  3  4 | p::first-letter {  color: red;  font-size: 130%;  } |

Efekat koji ovakav CSS opis proizvodi prikazan je slikom 12.25.



*Slika 12.25 – Stilizovanje prvog slova svakog paragrafa, korišćenjem pseudoelementa ::first-letter*

**::first-line (:first-line)**

Za stilizovanje prve linije teksta nekog elementa moguće je koristiti CSS selektor ::first-line. Reč je o pseudoelementu koji omogućava da se selektuje samo prva linija teksta nekog elementa. Višelinijski tekst moguć je samo kod block elemenata, odnosno kod elemenata čija je vrednost display svojstva block, inline-block, table-cell ili table-caption. Na inline elementima, pseudoelement ::first-line nema efekta.

Sintaksa pseudoelementa je sledeća:

|  |  |
| --- | --- |
| 1  2  3  4  5 | /\* CSS3 syntax \*/  ::first-line { style properties }    /\* CSS2 syntax \*/  :first-line { style properties } |

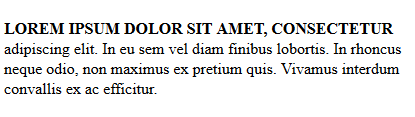
Pseudoelement dozvoljava stilizovanje teksta, tako da omogućava definisanje samo svojstava koja se tiču stilizovanja teksta. Sledeći primer ilustruje upotrebu ::first-line selektora. HTML kod je sledeći:

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non maximus ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>

HTML kod se sastoji iz jednog paragraf elementa. Koliko teksta će se naći u jednoj liniji na stranici, zavisi od puno faktora. Ipak, na sledeći način, biće stilizovana prva linija teksta prikazanog paragrafa, bez obzira na količinu teksta koja se unutar nje može naći:

|  |  |
| --- | --- |
| 1  2  3  4 | p::first-line {  text-transform: uppercase;  font-weight: bold;  } |

Prikazani CSS opis proizvodi efekat kao na slici 12.26.



*Slika 12.26 – Stilizovanje prve linije teksta svakog paragrafa, korišćenjem pseudoelementa ::first-line*

Bez obzira na veličinu prve linije, sav tekst unutar nje, biće stilizovan na način na koji je to definisano unutar opisa sa pseudoelementom ::first-line. Pritom, stilizacija će se dinamički prilagođavati promeni veličine prve linije.

**::selection**

Pseudoelementi omogućavaju stilizovanje dela teksta koji je selektovan. To je moguće postići zahvaljujući pseudoelementu ::selection. Ovaj pseudoelement omogućava definisanje CSS svojstava koja se tiču stilizovanja teksta. Tako je moguće definisati sledeća svojstva selektovanog teksta:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | Color  backround-color  cursor  outline  text-decoration  text-emphasis-color  text-shadow |

Sintaksa ::selection pseudoklase je sledeća:

|  |  |
| --- | --- |
| 1  2 | /\* Firefox syntax \*/  ::-moz-selection { style properties }::selection { style properties } |

Mozilla Firefox zahteva drugačiju sintaksu za korišćenje ovog pseudoelementa, dok ostali browseri poseduju izjednačenu sintaksu.

Sledi primer koji ilustruje korišćenje ::selection pseudoelementa. HTML kod je sledeći:

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu sem vel diam finibus lobortis. In rhoncus neque odio, non maximus ex pretium quis. Vivamus interdum convallis ex ac efficitur.</p>

Stilizovanje selektovanog teksta se postiže na sledeći način:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | ::-moz-selection { /\* Code for Firefox \*/  color: white;  background: coral;  }    ::selection {  color: white;  background: coral;  } |

Efekat koji se na ovakav način postiže je prikazan slikom 12.27.

