**Flexbox Layout**

**00:22:36**

Jedinica: 16 od 30

**+Rezime**

Organizovanje sadržaja na stranici uvek je predstavljalo izazov prilikom kreiranja UI-ja HTML aplikacija. U prethodnoj lekciji prikazane su osnovne tehnike koje se koriste za raspoređivanje elemenata na stranici. Osnovne tehnike podrazumevaju poznavanje nekoliko različitih CSS svojstava, njihovih mogućih vrednosti i efekata koje takve vrednosti proizvode. Pristupi koji su prikazani u prethodnoj lekciji smatraju se bazičnim, te stoga funkcionišu u većini parsera (*browsera*).

Ipak, tokom vremena u kojem se web razvijao velikom brzinom, javila se potreba za još boljim i **fleksibilnijim** načinom za raspoređivanje elemenata na stranici. Tako je nastao *Flexbox Layout*.

**Šta je Flexbox Layout?**

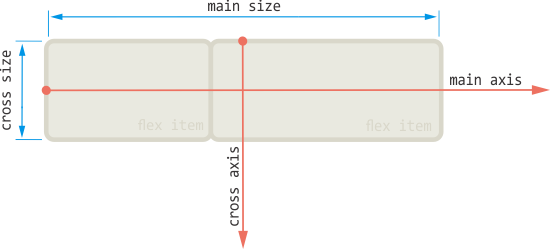
Flexbox Layout namenjen je efikasnijem raspoređivanju, centriranju i poravnavanju elemenata unutar kontejnerskih, roditeljskih elemenata, pa čak i kada veličina takvih elemenata nije poznata unapred. Osnovna ideja fleksibilnog [layouta](https://www.link-elearning.com/linkdl/opisPojma.php?id=144344) jeste omogućavanje roditeljskom elementu da kontroliše veličinu i poziciju svojih potomaka, kako bi se dostupni prostor na najbolji mogući način iskoristio. Glavni razlog ovakve potrebe jeste portabilnost, odnosno omogućavanje HTML stranici da bude adekvatno čitljiva na širokom spektru uređaja.

**Kako Flexbox Layout funkcioniše?**

U prethodnoj lekciji, kada je predstavljeno CSS svojstvo display, pomenuta je njegova vrednost flex. Ipak, *Flexbox Layout* je daleko više od puke vrednosti jednog CSS svojstva. Flexbox Layout je modul koji obuhvata kompletan skup svojstava. Neka od tih svojstava su zamišljena da se koriste na kontejnerskim, roditeljskim elementima, a druga na elementima potomcima. Iz upravo izrečene konstatacije, može se zaključiti da Flexbox Layout razlikuje dva tipa komponenata:

* flex container
* flex items

Jasno je da je *flex container* komponenta koja sadrži *flex items* elemente. Flex layout je zamišljen kao moderna alternativa raspoređivanju komponenata korišćenjem float svojstva. Ipak, i Flex layout se zasniva na tehnikama floatovanja elemenata, tako što uvek poseduje glavnu osu, duž koje se slažu elementi potomci. Način funkcionisanja Flexbox Layouta ilustrovan je slikom 16.1.



*Slika 16.1 – Način funkcionisanja Flexbox Layouta*

16.1 prikazuje flexbox kontejner, unutar koga se nalaze dva fleksibilna elementa označena kao *flex item*.

**Main axis** predstavlja osnovnu osu kontejnera, duž koje se raspoređuju elementi. Nije obavezno da glavna osa raspoređivanja bude horizontalna, kao na slici 16.1. Kao što ćete nešto kasnije videti, orijentacija glavne ose se menja korišćenjem flex-direction svojstva.

Pored osnovne ose, flexbox layout poseduje i takozvanu cross osu (engl. **cross axis**), koja je uvek normalna na glavnu osu raspoređivanja elemenata.

**Main size** i **Cross size** na slici označavaju širinu ili visinu, duž kojih je moguće ređati elemente.

**Implementiranje layouta korišćenjem Flexboxa**

Kreiranje *Flexbox Layouta* započinje kreiranjem kontejnera. On može izgledati ovako:

|  |  |
| --- | --- |
| 1  2 | <div id="flexbox1">  </div> |

Ovo je jedan najobičniji div element sa id vrednošću *flexbox1*. On se ni po čemu ne razlikuje od bilo kojeg drugog div elementa. Kako bi se od prikazanog div elementa napravio flexbox kontejner, dovoljno je definisati CSS display svojstvo sa vrednošću flex.

|  |  |
| --- | --- |
| 1  2  3  4 | #flexbox1 {  display: flex;  border: 1px solid black;  } |

Na ovaj način, div element *flexbox1* proglašen je kontejnerom za flexbox item elemente.

|  |
| --- |
| **Napomena**    *Podrška za flexbox raspoređivanje elemenata je vrlo šarolika i može se reći da samo najnovije verzije svih browsera podržavaju ovakav način raspoređivanja. Starije verzije browsera na različite načine omogućavaju podršku za flexbox, dok neki browseri takvu podršku ne nude uopšte. U najlošijem položaju je, tradicionalno, Internet Explorer, koji do verzije 11 ne podržava u potpunosti flexbox raspoređivanje elemenata. Internet Explorer 10 podržava flexbox samo parcijalno, i to korišćenjem sledeće sintakse:*    display: -ms-flexbox;    *Starije verzije Chrome, Safari i Opera web browsera flexbox podržavaju korišćenjem sledeće sintakse:*    display: -webkit-flexbox;  *Starije verzije Mozilla browsera flexbox podržavaju na sledeći način:*    display: -moz-flexbox;    *Na kraju, kao što je rečeno, najnovije verzije većine browsera podržavaju sintaksu koja je deo specifikacije, i ona će biti i korišćena u ovom kursu:*    display: flex; |

Kontejneru se elementi mogu dodati na sledeći način:

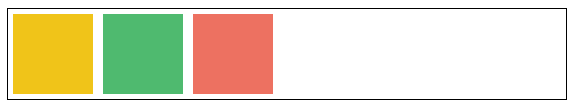
|  |  |
| --- | --- |
| 1  2  3  4  5 | <div id="flexbox1">  <div></div>  <div></div>  <div></div>  </div> |

Kako bi efekat flexbox layouta mogao biti vidljiv, elementi unutar *flexbox1*kontejnera biće stilizovani na sledeći način:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | #flexbox1 > div {  width: 80px;  height: 80px;  margin: 5px;  }    #flexbox1 > div:nth-child(1) {  background-color: #F0C419;  }    #flexbox1 > div:nth-child(2) {  background-color: #4FBA6F;  }    #flexbox1 > div:nth-child(3) {  background-color: #ED7161;  } |

|  |
| --- |
| **Napomena**    *U primeru CSS opisa, koriste se child selektori i pseudoklase, pojmovi koji su obrađeni u lekciji u kojoj je bilo reči o CSS selektorima.* |

Svim direktnim div potomcima elementa flexbox1 postavljene su visina, širina i margine. Takođe, svakom od div elemenata unutar flexbox kontejnera definisana je jedinstvena boja pozadine. Kod koji je do sada napisan sasvim je dovoljan da bi se video efekat flexbox raspoređivanja elemenata. To ilustruje slika 16.2.



*Slika 16.2 – Elementi raspoređeni korišćenjem flexboxa*

Podrazumevano, elementi unutar flexbox kontejnera raspoređuju se po horizontalnoj flex liniji, sleva nadesno, što dokazuje i slika 16.2. Za kontrolu ponašanja flexbox kontejnera i elemenata koji se unutar njega nalaze postoji veliki broj svojstava, od kojih se neka mogu definisati na kontejnerskom elementu, a neka na fleksibilnim elementima potomcima. U nastavku lekcije biće prikazani efekti koji se proizvode takvim svojstvima.

Svojstva koja je moguće definisati na roditeljskom flexbox elementu su sledeća:

flex-direction

justify-content

align-items

flex-wrap

align-content

flex-flow

Svojstva koja se mogu definisati na fleksibilnim elementima, potomcima su sledeća:

order

flex-basis

flex-grow

flex-shrink

flex

align-self

Sva navedena svojstva biće detaljno obrađena u nastavku lekcije.

**Kako bi se neki element učinio fleksibilnim kontejnerom, potrebno je postaviti vrednost display svojstva na:**

 flex

 flexbox

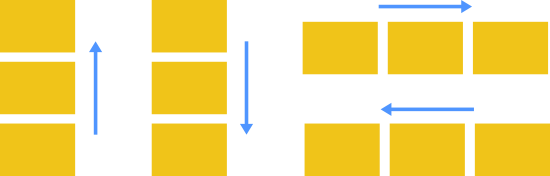
 inline-flex

 flexibile

**Flex Direction**

Nešto ranije je prikazano kako roditeljski flexbox element ređa svoje potomke duž glavne ose. Glavna osa je podrazumevano horizontalna, što znači da se elementi ređaju u liniji, sleva nadesno.

Promenu orijentacije glavne ose flexbox kontejnera moguće je postići korišćenjem svojstva flex-direction.



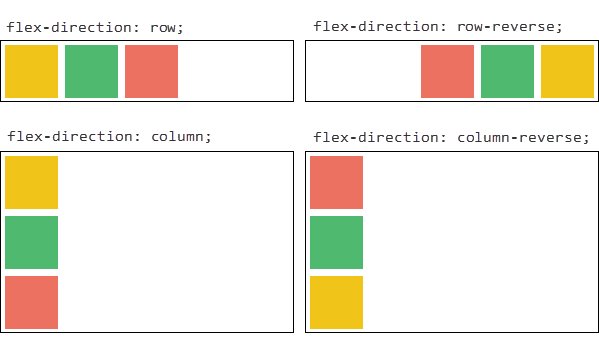
*Slika 16.3 – Efekti flex-direction svojstva*

flex-direction svojstvo može imati vrednosti prikazane tabelom 16.1.

|  |  |
| --- | --- |
| **Vrednost** | **Opis** |
| row | fleksibilni elementi se raspoređuju horizontalno, u redovima; ovo je podrazumevana vrednost |
| row-reverse | identičan efekat kao i row, samo što se elementi ređaju zdesna nalevo |
| column | fleksibilni elementi se raspoređuju vertikalno, u kolonama |
| column-reverse | identičan efekat kao i column, s tim što se elementi raspoređuju obrnutim redosledom, odozdo nagore |

*Tabela 16.1 – Vrednost flex-direction svojstva*

Efekti koje proizvode različite vrednost flex-direction svojstva prikazani su slikom 16.4.



*Slika 16.4 – Efekti flex-direction svojstva*

**Justify content**

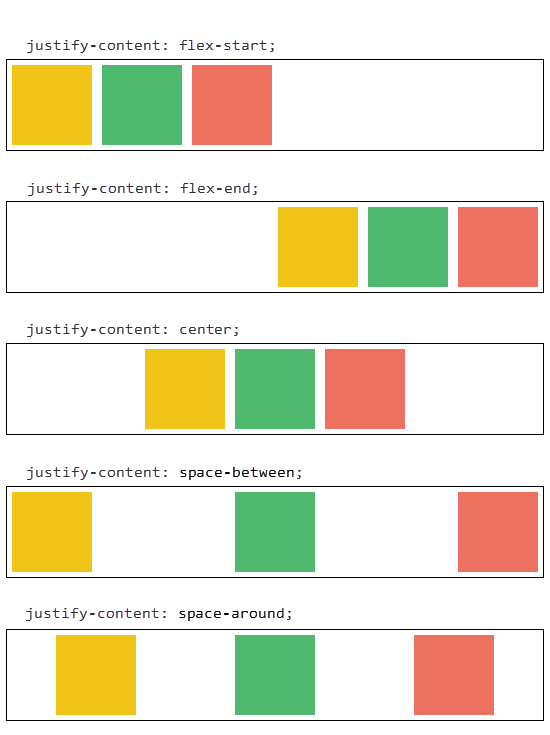
Svojstvom justify-content moguće je uticati na poravnanje elemenata unutar flexbox kontejnera kada oni ne zauzimaju kompletan dostupan prostor duž glavne ose. Ovo je veoma bitan detalj, koji je potrebno zapamtiti odmah na početku. Svojstvo justify-content odnosi se na poravnanje sadržaja duž glavne ose. Da li će to poravnanje biti po vertikalnoj ili horizontalnoj osi, zavisi od toga koja osa se smatra glavnom. Kao što je već rečeno, definisanje glavne ose se postiže svojstvom flex-direction.

Svojstvo justify-content može imati jednu od vrednosti prikazanih tabelom 16.2.

|  |  |
| --- | --- |
| **Vrednost** | **Opis** |
| flex-start | elementi se ređaju od početka glavne ose |
| flex-end | elementi se ređaju od kraja glavne ose |
| center | elementi su centrirani duž glavne ose |
| space-between | elementi su jednako raspoređeni duž glavne ose, tako što su prvi i poslednji element prilepljeni za ivice |
| space-around | elementi su jednako raspoređeni duž glavne ose, sa jednakim razmakom između svih elemenata |

*Tabela 16.2 – Vrednost justify-content svojstva*

Efekti koje proizvode različite vrednosti justify-content svojstva prikazani su slikom 16.5. Imajte na umu da flex-direction ima podrazumevanu vrednost row.



*Slika 16.5 – Efekti justify-content svojstva*

**Align items**

Baš kao što svojstvo justify-content poravnava elemente duž glavne ose, svojstvo align-item to čini po sporednoj cross osi. Naravno, primenljivo je na kontejner bez obzira na vrednost flex-direction svojstva. Preduslov da bi se izvršilo bilo koje poravnanje jeste to da postoji slobodan prostor unutar kontejnera koji elementi ne zauzimaju, po osi po kojoj se želi obaviti poravnanje.

Svojstvo align-items može imati jednu od vrednosti prikazanih tabelom 16.3.

|  |  |
| --- | --- |
| **Vrednost** | **Opis** |
| flex-start | elementi se poravnavaju u odnosu na početak cross ose |
| flex-end | elementi se poravnavaju u odnosu na kraj cross ose |
| center | elementi su centrirani po cross osi |
| baseline | elementi se poravnavaju po osnovnoj liniji sadržaja elemenata |
| stretch | elementi se razvlače kako bi popunili dostupan prostor unutar kontejnera po cross osi; ovo je podrazumevana vrednost |

*Tabela 16.3 – Vrednost align-items svojstva*

Efekti koje različite vrednosti align-items svojstva imaju na poravnanje elemenata prikazani su na slici 16.6.

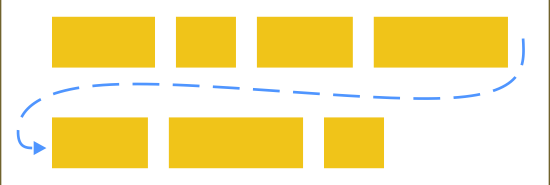


*Slika 16.6 – Efekti align-items svojstva*

Ovoga puta, div elementima je dodat sadržaj u obliku single karaktera A, B i C. Razlog je efekat koji osnovna linija sadržaja ima na poravnanje prilikom korišćenja baseline vrednosti. Da kojim slučajem sadržaj elemenata nije naveden, vrednosti baseline i flex-start bi imale identičan efekat.

**Flex wrap**

Podrazumevano, flex kontejner pokušava da sve elemente smesti u jednu liniju, bilo da je reč o horizontalnoj (red) ili vertikalnoj liniji (kolona). Ovakvo podrazumevano ponašanje je moguće promeniti korišćenjem svojstva flex-wrap i tako elemente rasporediti u više linija.



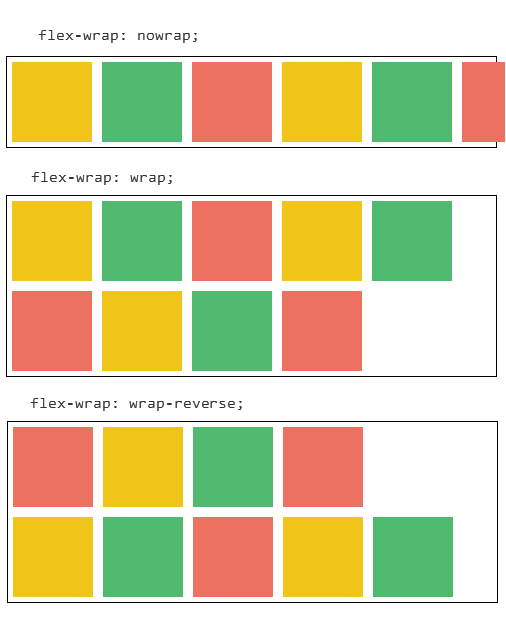
*Slika 16.7 – Efekat flex-wrap svojstva*

Svojstvo flex-wrap može imati vrednosti prikazane tabelom.

|  |  |
| --- | --- |
| **Vrednost** | **Opis** |
| nowrap | fleksibilni elementi se raspoređuju u jednoj liniji; podrazumevana vrednost |
| wrap | fleksibilni elementi se raspoređuju u više linija |
| wrap-reverse | fleksibilni elementi se raspoređuju u više linija, obrnutim redosledom od opcije wrap |

*Tabela 16.4 – Vrednost flex-wrap svojstva*

Efekti koji se postižu različitim vrednostima flex-wrap svojstva prikazani su slikom 16.8.



*Slika 16.8 – Efekat flex-wrap svojstva*

**Align content**

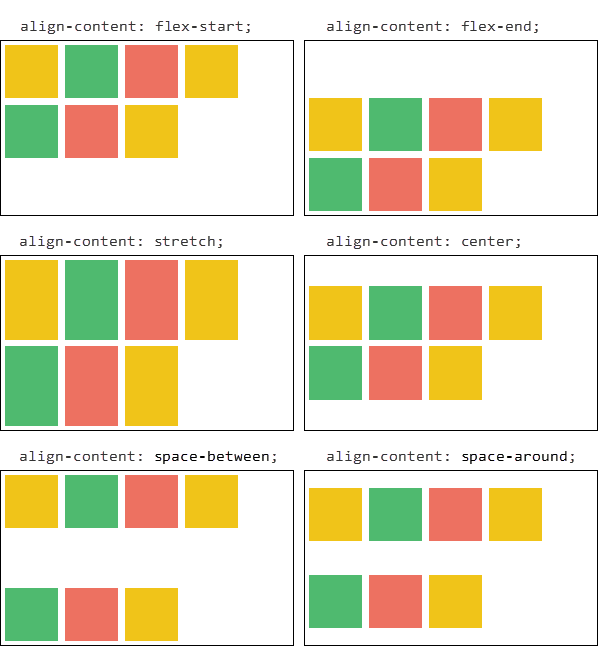
Svojstvo align-content utiče na ponašanje upravo opisanog flex-wrap svojstva. Slično svojstvu align-items, svojstvo align-content poravnava višelinijski fleksibilni sadržaj. Upravo zbog toga, ovo svojstvo ima efekta samo kada flex-wrap ima vrednost različitu od nowrap.

Svojstvo align-content može imati jednu od vrednosti predstavljenih tabelom.

|  |  |
| --- | --- |
| **Vrednost** | **Opis** |
| stretch | sadržaj linija se proširuje kako bi se zauzeo kompletno dostupan prostor; ovo je podrazumevana vrednost |
| flex-start | linije sa sadržajem se poravnavaju po početnoj liniji kontejnera |
| flex-end | linije sa sadržajem se poravnavaju po krajnjoj liniji kontejnera |
| center | linije sa sadržajem se poravnavaju po sredini kontejnera |
| space-between | linije sa sadržajem se raspoređuju ravnomerno unutar kontejnera, pri čemu se prva linija smešta na početak, a poslednja na kraj kontejnera |
| space-around | linije sa sadržajem se ravnomerno raspoređuju sa jednakim praznim prostorom pre i posle |

*Tabela 16.5 – Vrednost align-content svojstva*

Efekti koje proizvode različite vrednosti align-contet svojstva prikazani su slikom 16.9.



*Slika 16.9 – Efekat align-contet svojstva*

|  |
| --- |
| **Napomena**    *Svojstvo align-content nema efekta ukoliko unutar kontejnera postoji samo jedna linija elemenata.* |

**Flex flow**

U dosadašnjem toku lekcije prikazana su svojstva flex-direction  i flex-wrap. Ova svojstva mogu se definisati i objedinjeno, korišćenjem svojstva flex-flow. Tako je sintaksa ovog svojstva sledeća:

|  |  |
| --- | --- |
| 1 | flex-flow: <‘flex-direction’> || <‘flex-wrap’> |

Podrazumevana vrednost svojstva flex-flow sastoji se iz podrazumevanih vrednosti pojedinačnih svojstava, tako da izgleda ovako:

|  |  |
| --- | --- |
| 1 | flex-flow: row nowrap; |

**Order**

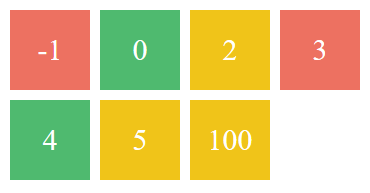
Sva do sada prikazana svojstva odnosila su se isključivo na kontejnerski element Flexbox layouta. Pored takvih svojstava, flexbox sistem poznaje i svojstva koja se definišu na fleksibilnim elementima potomcima, te se na taj način utiče na njihovo razmeštanje unutar kontejnera. Prvo takvo svojstvo koje će biti predstavljeno u ovoj lekciji jeste svojstvo order.

Svojstvom order utiče se na redosled elemenata unutar kontejnera. Podrazumevano, elementi se unutar kontejnera ređaju po redosledu po kojem su navedeni u HTML dokumentu. To je na lak način moguće promeniti, definisanjem vrednosti svojstva order na pojedinačnim fleksibilnim elementima.

Sledeći primer ilustruje definisanje redosleda elementa:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <div id="flexbox1">   <div style="order: 5;">5</div>   <div style="order: 4;">4</div>   <div style="order: 3;">3</div>   <div style="order: 2;">2</div> <div>0</div>   <div style="order: -1;">-1</div>   <div style="order: 100;">100</div>  </div> |

Na elementima koji se nalaze unutar flexbox kontejnera definisane su vrednosti svojstva order. To je učinjeno na svim, osim na jednom div elementu, koji u takvoj situaciji ima podrazumevanu vrednosti order svojstva. Podrazumevana vrednost order svojstva je nula. Elementima je dodat i sadržaj koji će ilustrovati vrednost njihovog order svojstva. Prikazani kod proizvešće efekat kao na slici 16.10.



*Slika 16.10 – Efekat order svojstva*

Analizom slike 16.10, lako se zaključuje da se elementi sa manjom vrednošću order svojstva renderuju pre onih sa većom.

**Flex basis**

Inicijalna veličina koju je potrebno da fleksibilni element poseduje unutar kontejnera definiše se korišćenjem svojstva flex-basis. Ovo svojstvo može da prihvati vrednost izraženu u različitim apsolutnim ili relativnim jedinicama. Sledeći primer ilustruje upotrebu flex-basis svojstva.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <div id="flexbox1">   <div style="flex-basis: 5em;">5em</div>   <div style="flex-basis: 10em;">10em</div>   <div style="flex-basis: 5em;">5em</div>  </div> |

U prikazanom primeru navedene su vrednosti korišćenjem relativnih em jedinica. Efekat će biti kao na slici 16.11.



*Slika 16.11 – Efekat flex-basis svojstva*

Sa slike 16.11 se može videti da elementi imaju inicijalnu veličinu koja je zadata korišćenjem svojstva flex-basis.

Pored vrednosti u relativnim i apsolutnim jedinicama, svojstvo flex-basis može imati vrednost izraženu ključnom rečju auto. To je inače podrazumevana vrednost, po kojoj se za utvrđivanje veličine elementa uzima u obzir veličina koja je definisana korišćenjem CSS width ili heightsvojstava. Ukoliko element ne poseduje eksplicitno definisanu veličinu ni korišćenjem ovih svojstava, njegova veličina će biti taman tolika da obuhvati pripadajući sadržaj.

**Flex grow**

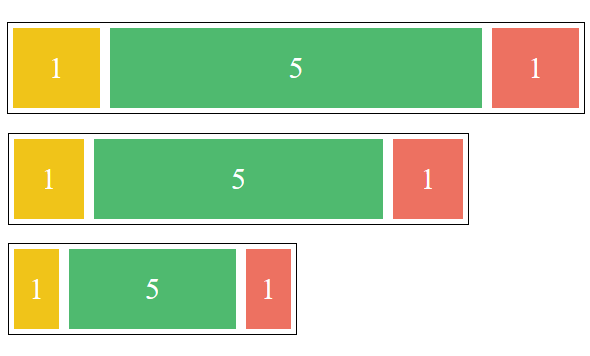
Upravo je prikazano kako se definiše početna veličina fleksibilnih elemenata duž glavne ose. Postavlja se pitanje šta će se dogoditi sa veličinom elemenata u situacijama kada je dostupan prostor kontejnera veći ili manji od onog koji je potreban za prikaz elemenata sa inicijalnom veličinom.

Svojstvo flex-grow utiče na mogućnost elementa da povećava svoju veličinu, ukoliko postoji prostor koji je potrebno popuniti. Pomenuti dostupan prostor može se videti na slici 16.11, sa desne strane.

Svojstvo flex-grow prihvata celobrojnu, pozitivnu vrednost, koja predstavlja proporciju koju element treba da zauzme unutar svog roditeljskog kontejnera. Podrazumevana vrednost je nula, što znači da element neće povećavati svoju veličinu. Sledeći primer ilustruje upotrebu svojstva flex-grow:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <div id="flexbox1">   <div style="flex-grow: 1;">1</div>   <div style="flex-grow: 5;">5</div>   <div style="flex-grow: 1;">1</div>  </div> |

Na elementima su definisane vrednosti flex-grow svojstava. Prvi i poslednji element imaju vrednosti 1, dok središnji element ima vrednost 5. Na ovaj način je definisano da će središnji element biti uvek 5 puta veći od prvog i poslednjeg. Takvo ponašanje ilustruje slika 16.12.



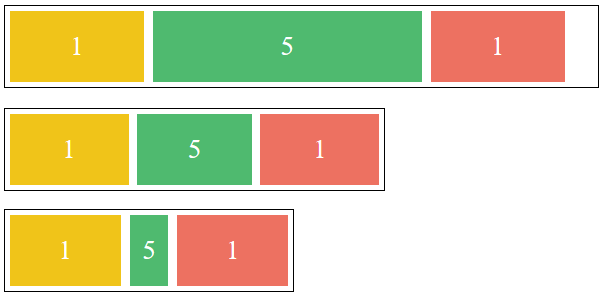
*Slika 16.12 – Efekat flex-grow svojstva*

**Flex shrink**

Podrazumevano, kada se veličina kontejnera smanji, smanjuju se i veličine njegovih elemenata. Svojstvo flex-shrink definiše koliko će se element smanjivati u odnosi na ostale elemente flexbox kontejnera. Sledeći primer ilustruje upotrebu ovoga svojstva:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <div id="flexbox1">   <div style="flex-basis: 5em; flex-shrink: 1;">1</div>   <div style="flex-basis: 10em; flex-shrink: 5;">5</div>   <div style="flex-basis: 5em; flex-shrink: 1;">1</div>  </div> |

U primeru je definisano da će se središnji div element smanjiti (ukoliko je to moguće) pet puta više od elemenata koji kao vrednost svojstva flex-shrink imaju jedinicu. Efekat je ilustrovan slikom 16.13.



*Slika 16.13 – Efekat flex-shrink svojstva*

**Flex**

S obzirom da su prethodna tri navedena svojstva, flex-basis, flex-grow i flex-shrinkpovezana, CSS poznaje svojstvo flex, kojim je moguće na lak način definisati vrednosti sva tri pomenuta svojstva. Njegova sintaksa je sledeća:

|  |  |
| --- | --- |
| 1 | flex: flex-grow flex-shrink flex-basis; |

Sledeći primer ilustruje definisanje vrednosti flex svojstva:

|  |  |
| --- | --- |
| 1 | flex: 2 2 10%; |

Svojstvo flex je moguće navesti sa jednom, dve ili tri vrednosti. Različiti scenariji postavljaju vrednosti različitih svojstava, a sledeći primer ilustruje različite situacije setovanja vrednosti:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | /\* 0 0 auto \*/  flex: none;    /\* One value, unitless number: flex-grow \*/  flex: 2;    /\* One value, width/height: flex-basis \*/  flex: 10em;  flex: 30px;  flex: auto;  flex: content;    /\* Two values: flex-grow | flex-basis \*/  flex: 1 30px;    /\* Two values: flex-grow | flex-shrink \*/  flex: 2 2;    /\* Three values: flex-grow | flex-shrink | flex-basis \*/  flex: 2 2 10%; |

**Align self**

Nešto ranije opisano je svojstvo align-items, kojim se postiže poravnavanje elementa po cross osi, flexbox kontejnera. Svojstvom se poravnavaju svi elementi kontejnera, a ukoliko je potrebno definisati način poravnavanja za samo jedan element, može se koristiti svojstvo align-self, koje se definiše na konkretnom elementu. Na taj način, moguće je pregaziti način poravnanja koji je definisan za ceo kontejner i definisati specifično poravnanje na nivou jednog elementa. Svojstvo align-self može imati vrednosti identične onima koje može imati svojstvo align-items.

|  |
| --- |
| **Napomena**    *Vrednost baseline na samo jednom elementu ima efekat identičan onom koji proizvodi vrednost flex-start. Zato je za testiranje efekta baseline vrednosti, neophodno postaviti takvo poravnanje na najmanje dva elementa.* |

**Primer**

**Kreiranje layouta korišćenjem Flexboxa**

*Naredni primer prikazaće kreiranje standardnog layouta web strane korišćenjem flexbox pristupa.*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | <!DOCTYPE html>  <html>  <head>   <title>Flexbox Layout Demo</title>   <style>    body{     font-size: 100%;     color: white;    }    .wrapper {     display: -webkit-box;     display: -moz-box;     display: -ms-flexbox;     display: -webkit-flex;     display: flex;     -webkit-flex-flow: row wrap;     flex-flow: row wrap;     font-weight: bold;     text-align: center;     margin: 0;    }    header, footer {     flex-basis: 100%;     /\*shorthand\*/     /\*flex: 1 100%;\*/    }    .aside {     flex-basis: auto;     flex-grow: 1;     /\*shorthand\*/     /\*flex: 1 auto; \*/    }    .main {     flex-grow: 5;     flex-basis: 0%;     /\*shorthand\*/     /\*flex: 5;\*/    }    .header, .footer, .aside, .main{     font-size: 1.2em;     padding: 10px;    }    header{     background-color: #929296;    }    .main{     background-color: #ABCADB;     text-align: left;    }    .main > h2{     text-align: center;    }    footer{     background-color: #5F5F64;    }    .aside-1{     background-color: #A4A098;    }    .aside-2{     background-color: #ADACC2;    }   </style>  </head>  <body class="wrapper">   <header class="header">    <h2>Header</h2>   </header>   <aside class="aside aside-1">    <h2>Aside 1</h2>   </aside>   <div class="main">    <h2>Main</h2>    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque pretium varius varius. Duis volutpat elementum neque, sed congue neque luctus ut. Nunc ligula lacus, hendrerit eget urna finibus, tristique ullamcorper nulla. Pellentesque congue laoreet eros. Nunc sit amet ipsum id ex rhoncus ornare quis eu mi. Morbi ac finibus odio. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nunc mattis tempus tempor. Etiam at neque sed nisl ornare semper pharetra non sapien. Praesent pellentesque, ipsum a fermentum fermentum, sapien dui pharetra felis, in dictum arcu urna in nibh. Aliquam pulvinar pellentesque nisl, pharetra ultricies mauris auctor vel. Aliquam erat volutpat. Vestibulum suscipit ut ligula finibus semper. In hac habitasse platea dictumst. Fusce ante nisi, auctor vel fermentum sed, tincidunt non ex. </p>   </div>   <aside class="aside aside-2">    <h2>Aside 2</h2>   </aside>   <footer class="footer">    <h2>Footer</h2>   </footer>  </body>  </html> |

*Efekat koji se na ovakav način postiže prikazan je na slici 16.14.*



*Slika 16.14 – Efekat flex-shrink svojstva*

Kod kompletnog primera nalazi se u prilogu lekcije.