

# Less

Less, odnosno **Leaner Style Sheets**, jeste još jedan pretprocesorski jezik za stilizovanje, koji se, uz Sass, svrstava u grupu najpopularnijih jezika ove vrste. Less su kreirali Alexis Sellier i Dmitry Fadeyev, 2009. godine.



Slika 25.1. Less logo

S obzirom na to da je nastao nekoliko godina nakon Sass jezika, Less je u najvećoj meri inspirisan svojim starijim bratom. Ipak, za razliku od Sassa, Less je od samog početka imao samo jednu sintaksu, koja je predstavljala nadgradnju standardnog CSS skupa funkcionalnosti. Zbog toga se može reći da je Less najbliži SCSS varijanti Sass jezika.

Mogućnosti Sass i Less jezika veoma su slične, što se može reći i za njihove sintakse. Male razlike postoje u načinu na koji se ovi jezici integrišu u projekat. U nastavku ove lekcije biće razmotrene osnovne osobine Less jezika, sa posebnim osvrtom na poređenje sa već obrađenim jezikom Sass.

## Preduslovi za korišćenje Lessa

Less je isprva bio implementiran korišćenjem Ruby jezika. To praktično znači da se Ruby koristio za prevođenje Less koda u CSS. Nešto kasnije, Ruby je zamenjen JavaScript jezikom, tako da se danas za prevođenje Less koda koristi osnovni programski jezik frontend razvoja.

S obzirom na to da se za prevođenje koda napisanog Less jezikom koristi JavaScript, postoji nekoliko načina na koje je takav posao moguće obaviti:

- **prevođenje korišćenjem specijalizovanih aplikacija**

Najjednostavniji način za prevođenje Less koda omogućavaju specijalizovani programi sa grafičkim okruženjem. Postoji veliki broj takvih programa namenjenih različitim operativnim sistemima. Može se izdvojiti program Koala, dostupan na sledećoj adresi:

<http://koala-app.com/>

Koala je jednostavna aplikacija sa grafičkim okruženjem i nju je moguće koristiti na Windows, macOS i Linux operativnim sistemima. Aplikacija je besplatna za korišćenje.

- **prevođenje od strane node.js-a**

Primarno izvršno okruženje JavaScript jezika jesu različiti web browseri unutar kojih su implementirani sistemi za prevođenje i izvršavanje JavaScript koda. Ipak, JavaScript se može izvršavati i izvan browsera, a najpopularnije takvo izvršno okruženje je node.js. Stoga je korišćenje node.js izvršnog okruženja veoma popularan pristup za prevođenje Less koda u CSS.

- **prevođenje od strane browsera**

Prevođenje Lessa u CSS može se obaviti i korišćenjem web pregledača, koji takav posao obavlja prilikom parsiranja web stranice. Ipak, ovakav pristup se preporučuje samo prilikom testiranja, zato što značajno usporava učitavanje web stranica, s obzirom na to da web pregledač samostalno mora da obavi prevod Less koda pre nego što stranicu prikaže korisniku.

## **Prevođenje Less koda korišćenjem modula za node.js**

U nastavku ove lekcije, za obavljanje prevođenja Less koda biće ilustrovan pristup koji podrazumeva korišćenje node.js izvršnog okruženja i specijalnog Less modula koji će se unutar takvog okruženja izvršavati. U realizaciji takvog posla, biće obavljeni sledeći koraci:

### **1. preuzimanje i instalacija node.js, uključujući i npm, kao i smeštanje putanje na kojoj se node.js nalazi unutar Path promenljive**

Prvi korak jeste instalacija node.js korišćenjem odgovarajućeg instalacionog fajla, koji se može preuzeti sa sledeće adrese:

<https://nodejs.org/en/download/>

Nakon preuzimanja instalacionog fajla za odgovarajući operativni sistem, neophodno je pokrenuti instalaciju i pratiti uputstva. Instalacija podrazumeva uspostavljanje node.js izvršnog okruženja, ali i specijalnog menadžera paketa koji se naziva **npm** (*node package manager*). Kao što ćete uskoro videti, npm omogućava veoma lako preuzimanje i integraciju eksternih JavaScript biblioteka – baš kakva je i Less biblioteka, koja nam je u ovom slučaju potrebna. Takođe, tokom instalacije, za nas će automatski biti obavljeno i smeštanje putanje na kojoj se nalazi node.js unutar sistemske Path promenljive.

Kako biste se uverili da je instalacija node.js-u uspešno obavljena, unutar konzole/terminala je dovoljno uneti sledeću komandu:

```
node --version
```

Ova komanda će prikazati instaliranu verziju node.js-a – naravno, ukoliko je instalacija uspešno obavljena i putanja na kojoj se node.js nalazi smeštena unutar Path promenljive.

### **2. instalacija Less modula**

Nakon instalacije node.js izvršnog okruženja i npm menadžera paketa, može se obaviti instalacija Less node.js modula. To se obavlja upućivanjem sledeće komande korišćenjem konzole/terminala:

```
npm install -g less
```

Upućivanjem prikazane komande, Node menadžer paketa (npm) obavlja instalaciju Less modula. Provera uspešnosti ovog koraka može se obaviti upućivanjem sledeće komande:

```
lessc -version
```

Prikazana komanda ispisuje instaliranu verziju Less kompajlera.

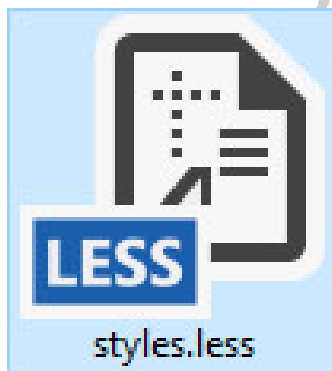
### 3. upućivanje komande za prevođenje korišćenjem konzole, odnosno terminala

Nakon instalacije Less kompajlera koji se izvršava posredstvom node.js izvršnog okruženja, prevođenje Less koda u CSS može se obaviti upućivanjem sledeće komande:

```
lessc styles.less styles.css
```

## Kreiranje Less dokumenta

Pisanje Less koda obavlja se unutar fajlova sa `.less` ekstenzijom (slika 25.2).



Slika 25.2. Less fajl

Svaki CSS kod ujedno je i potpuno validan Less kod, zato što Less predstavlja nadogradnju CSS jezika. Stoga, za početak, unutar Less dokumenta možemo napisati:

```
body {  
  font-family: sans-serif;  
}
```

Prikazanim CSS opisom, postavlja se familija fonta za kompletan HTML dokument. Ovakav kod Less kompajler će prevesti u sledeći CSS:

```
body {  
  font-family: sans-serif;  
}
```

Jasno je da u ovakvoj situaciji Less i CSS izgledaju identično, s obzirom na to da prilikom pisanja Less koda nisu iskorišćene nikakve specifične Less konstrukcije.

LESS	CSS
<pre>body {   font-family: sans-serif; }</pre>	<pre>body {   font-family: sans-serif; }</pre>

Slika 25.3. Svaki CSS kod ujedno je i potpuno validan Less kod

U nastavku ove lekcije biće ilustrovani različiti jezički elementi koji su karakteristični za Less.

## Less promenljive (variable)

Promenljive se u Lessu kreiraju korišćenjem karaktera at (@), dok se naziv i vrednost razdvajaju karakterom dve tačke (:):

```
@variablename: value;
```

Razlika u odnosu na Sass ogleda se u upotrebi karaktera @ umesto karaktera \$.

Primer kreiranja jedne Less promenljive može da izgleda ovako:

```
@primary-dark-color: #A2687E;
```

Jednom definisana promenljiva može se iskoristiti za postavljanje vrednosti nekog CSS svojstva na sledeći način:

```
body {
  font-family: sans-serif;
  color: @primary-dark-color;
}
```

LESS	CSS
<pre>@primary-dark-color: #A2687E;  body {   font-family: sans-serif;   color: @primary-dark-color; }</pre>	<pre>body {   font-family: sans-serif;   color: #A2687E; }</pre>

Slika 25.4. Primer upotrebe Less promenljivih

## Oblast važenja Less promenljivih

Oblast važenja Less promenljivih ista je kao kod Sass varijabli. Prioritet imaju promenljive definisane unutar trenutnog bloka, a ukoliko ih nema, potražuju se globalno.

LESS	CSS
<pre>@primary-dark-color: #A2687E;  body {   font-family: sans-serif;   color: @primary-dark-color; }  h1 {   @primary-dark-color: blue;   color: @primary-dark-color; }</pre>	<pre>body {   font-family: sans-serif;   color: #A2687E; } h1 {   color: blue; }</pre>

Slika 25.5. Oblast važenja Less promenljivih

Na slici 25.5. može se videti da se unutar bloka koji predstavlja CSS opis sa `h1` tipskim selektorom koristi lokalno definisana Less promenljiva. Unutar `body` bloka, koristi se globalno definisana Less promenljiva.

## Interpolacija promenljivih

Prethodni primeri ilustrovali su korišćenje Less promenljivih za definisanje vrednosti unutar CSS opisa. Ipak, Less promenljive se mogu koristiti i na mnogim drugim mestima: za definisanje selektora, svojstava, URL adresa... U takvim situacijama, vrednosti Less promenljivih su tekstualne i u većini slučajeva potrebno ih je umetnuti u već postojeći kontekst CSS koda. Takav posao se u programiranju drugačije naziva interpolacija.

### Promenljiva unutar selektora

Vrednost promenljive može se iskoristiti za kreiranje selektora. Takav primer ilustrovan je slikom 25.6.

LESS	CSS
<pre>@my-selector: box;  .#{@my-selector} {   width: 960px;   margin: 0 auto; }  .top-#{@my-selector} {   width: 960px;   height: 250px;   margin: 0 auto; }</pre>	<pre>.box {   width: 960px;   margin: 0 auto; } .top-box {   width: 960px;   height: 250px;   margin: 0 auto; }</pre>

Slika 25.6. Primer upotrebe Less promenljive za formulaciju selektora

Primer ilustruje kreiranje jedne Less promenljive sa nazivom `@my-selector` i vrednošću `box`. Vrednost takve promenljive upotrebljena je za formulisanje dva selektora. Interpolacija, odnosno umetanje tekstualne vrednosti promenljive, obavljena je korišćenjem vitičastih zagrada kojima prethodi karakter `@`.

#### Napomena

U starijim verzijama Less jezika, tekstualna vrednost promenljivih smeštala se između dvostrukih ili jednostrukih navodnika ispred kojih se navodio karakter tilda (`~`):

```
@my-selector: ~"box";
```

ili

```
@my-selector: ~'box';
```

U nastavku ove lekcije koristiće se noviji pristup, koji podrazumeva definisanje tekstualnih vrednosti bez navodnika.

#### Vrednost promenljive unutar naziva CSS svojstva

Less promenljive se mogu koristiti i za definisanje naziva CSS svojstva. Procedura je ista kao i prilikom definisanja selektora (slika 25.7).

LESS	CSS
<pre>@w: width;  .box {   @w: 960px;   margin: 0 auto; }</pre>	<pre>.box {   width: 960px;   margin: 0 auto; }</pre>

Slika 25.7. Primer upotrebe Less promenljive za formulaciju naziva CSS svojstva

U prikazanom primeru, vrednost promenljive je iskorišćena za definisanje naziva CSS svojstva `width`. Interpolacija je obavljena na isti način kao i prilikom definisanja naziva selektora, odnosno korišćenjem konstrukcije `@{variable_name}`, gde je `variable_name` naziv Less promenljive.

#### Vrednost promenljive za kreiranje medija upita

Interpolacija promenljivih u Less jeziku može biti posebno korisna prilikom definisanja medija upita. Tako ona dozvoljava da se pojedinačni uslovi definišu na jednom mestu, a zatim koriste proizvoljan broj puta u značajno kompaktnijem obliku (slika 25.8).

LESS	CSS
<pre> @min1280: (min-width: 1280px);  .box {   @media @min1280 {     width: 1024px;   } }  #header {   @media @min1280 {     height: 60px;   } } </pre>	<pre> @media (min-width: 1280px) {   .box {     width: 1024px;   } }  @media (min-width: 1280px) {   #header {     height: 60px;   } } </pre>

Slika 25.8. Primer upotrebe Less promenljive za formulisanje medija upita

Primer ilustruje kreiranje jedne Less promenljive sa nazivom `@min1280` i vrednošću `(min-width: 1280px)`. Ovako definisana promenljiva iskorišćena je prilikom kreiranja dva medija upita unutar bloka `.box` i bloka `header`.

### Pitanje

Less promenljive započinju karakterom:

- @
- &
- \$
- ~

### Objašnjenje:

*Promenljive se u Lessu kreiraju korišćenjem karaktera at (@), dok se naziv i vrednost razdvajaju karakterom dve tačke (:).*

## Gnežđenje

Baš kao Sass, i Less omogućava gnežđenje stilizacije, čime se izbegava korišćenje određenih selektora zasnovanih na relacijama i dobija mnogo pregledniji kod za stilizovanje. Primer Less gnežđenja ilustrovan je slikom 25.9.

LESS	CSS
<pre> article {   h1 {     color: red;   }    h2 {     color: blue;   }    p {     color: green;   } } </pre>	<pre> article h1 {   color: red; }  article h2 {   color: blue; }  article p {   color: green; } </pre>

Slika 25.9. Primer gnežđenja u Lessu

Less omogućava i gnežđenje medija upita. Na taj način je stilizaciju za jedan element moguće objediniti unutar jednog bloka, kombinovanjem osnovne stilizacije, ali i one koja će se aktivirati u posebnim situacijama. Takav primer ilustrovan je slikom 25.10.

LESS	CSS
<pre> .box {   width: 960px;    @media (min-width: 1280px) {     width: 800px;   }    @media (min-width: 768px) {     width: 600px;   } } </pre>	<pre> .box {   width: 960px; }  @media (min-width: 1280px) {   .box {     width: 800px;   } }  @media (min-width: 768px) {   .box {     width: 600px;   } } </pre>

Slika 25.10. Primer gnežđenja medija upita u Lessu

Na levoj strani slike 25.10 može se videti da su medija upiti smešteni unutar CSS opisa sa selektorom na koji se odnose. Zbog toga se unutar medija upita ne navodi nikakav selektor. Prilikom prevođenja, Less kompajler od ovakvog opisa formira tri zasebna bloka, što se može videti na desnoj polovini slike 25.10.

Moguće je otići i korak dalje, te obaviti gnežđenje medija upita na više nivoa, baš kao na slici 25.11.



LESS	<pre> .box {   background-image: url("tokyo-1024w.jpg");    @media (min-width: 768px) {     background-image: url("tokyo-768w.jpg");   }    @media (min-width: 414px) {     background-image: url("tokyo-414w.jpg");      @media (min-resolution: 2dppx) {       background-image: url("tokyo-414w@2.jpg");     }      @media (min-resolution: 3dppx) {       background-image: url("tokyo-414w@3.jpg");     }   } } </pre>
CSS	<pre> .box {   background-image: url("tokyo-1024w.jpg"); } @media (min-width: 768px) {   .box {     background-image: url("tokyo-768w.jpg");   } } @media (min-width: 414px) {   .box {     background-image: url("tokyo-414w.jpg");   } } @media (min-width: 414px) and (min-resolution: 2dppx) {   .box {     background-image: url("tokyo-414w@2.jpg");   } } @media (min-width: 414px) and (min-resolution: 3dppx) {   .box {     background-image: url("tokyo-414w@3.jpg");   } } </pre>

Slika 25.11. Primer gnežđenja medija upita u više nivoa

Na slici 25.11, posebnu pažnju je potrebno obratiti na drugi medija upit unutar Less koda. Unutar njega nalazi se jedna deklaracija i još dva medija upita. Ovakvim gnežđenjem medija upita dolazi do nadovezivanja uslova, i to logičkim operatorom **AND**, što se na slici 25.11. može videti analizom CSS koda koji je dobijen prevođenjem Lessa.

## Mixini

Pojam mixina u Lessu poseduje nešto drugačije značenje nego u jeziku Sass. Tako Less mixini omogućavaju uključivanje CSS deklaracija iz jednog CSS opisa u neki drugi. Na primer, pogledajte sledeći CSS opis:

```

.foo {
  color: blue;
}

```

Upravo je prikazan CSS opis sa jednom deklaracijom kojom se postavlja boja teksta na plavu. Ovakva deklaracija se pomoću Lessa veoma lako može ugraditi u neki drugi CSS opis, na sledeći način:

```
#text-1 {
  .foo();
  font-size: 16px;
}

#text-2 {
  .foo();
  font-size: 18px;
}
```

Efekat prikazanog primera ilustrovan je slikom 25.12.

LESS	CSS
<pre>.foo {   color: blue; }  #text-1 {   .foo();   font-size: 16px; }  #text-2 {   .foo();   font-size: 18px; }</pre>	<pre>.foo {   color: blue; } #text-1 {   color: blue;   font-size: 16px; } #text-2 {   color: blue;   font-size: 18px; }</pre>

Slika 25.12. Primer mixina u Lessu

Sa slike 25.12. može se videti da se Less mixini *pozivaju* korišćenjem zagrada nakon kojih se navodi karakter tačka-zapeta (;). Efekat Less mixina jeste dodavanje deklaracija koje se nalaze unutar navedenog CSS opisa, što se može videti na desnoj polovini slike 25.12.

Upravo prikazani primer podrazumevao je jedan mixin koji je zapravo potpuno ravnopravan CSS opis. To se može zaključiti i po tome što je on sastavni deo CSS koda koji se dobija nakon prevođenja. Ukoliko je potrebno dobiti mixin čija će jedina uloga biti integracija unutar nekih drugih opisa, takav mixin se može formirati na sledeći način:

```
.foo() {
  color: blue;
}
```

Sada su nakon selektora dodate zagrade. Na ovaj način je rečeno da ovakav opis neće biti deo prevedenog CSS koda, već će isključivo moći da se upotrebi unutar nekog drugog opisa.

## Mixini kao mape

Less mixini mogu se koristiti na još jedan zanimljiv način. Naime, oni omogućavaju definisanje grupa srodnih vrednosti koje se drugačije nazivaju mape:

```
#sizes() {  
  mobile : 320px;  
  tablet : 768px;  
  desktop: 1280px;  
}
```

Ovakav mixin može se doživeti kao skup vrednosti koje se mogu koristiti nezavisno. Na primer:

```
.box {  
  @media (min-width: #sizes[mobile]) {  
    width: 300px;  
  }  
}
```

Za definisanje vrednosti unutar uslova medija upita sada je iskorišćena vrednost proizvoljnog svojstva iz mixina. Ona je unutar medija upita smeštena korišćenjem naziva mixina i uglastih zagrada – []. Kompletan primer je ilustrovan slikom 25.13.

LESS	CSS
<pre>#sizes() {   mobile : 320px;   tablet : 768px;   desktop: 1280px; }  .box {   @media (min-width: #sizes[mobile]) {     width: 300px;   } }</pre>	<pre>@media (min-width: 320px) {   .box {     width: 300px;   } }</pre>

Slika 25.13. Primer korišćenja mixina kao mapa

## Operatori

Less dozvoljava korišćenje operatora, i to prilikom definisanja vrednosti CSS svojstava ili Less promenljivih. Operatore je moguće koristiti nad numeričkim vrednostima i bojama. Slika 25.14. ilustruje nekoliko različitih upotreba Less operatora.

LESS	CSS
<pre>@base_width: 600px / 960px; @base_height: 40%;  .box {   width: @base_width * 100%;   height: @base_height * 2; }</pre>	<pre>.box {   width: 62.5%;   height: 80%; }</pre>

Slika 25.14. Primer korišćenja operatora u Lessu

Na slici 25.14. može se videti da je jedan od operatora prvo iskorišćen prilikom definisanja vrednosti promenljive `@base_width`. Reč je o operatoru za deljenje (/). Prilikom evaluacije vrednosti ove promenljive, Less kompajler je izračunao vrednosti izraza koji se na ovakav način dobija.

Na slici 25.14. prikazana je upotreba operatora i prilikom definisanja vrednosti CSS svojstava `width` i `height`, gde je upotrebljen operator množenja (\*).

Less je u određenoj meri sposoban da obavi konverziju vrednosti, ukoliko je reč o srodnim tipovima. Tako, na primer, `5cm + 10mm` postaje `6cm`. Ipak, Less ne može da obavi konverziju između tipova koji nisu srodni (na primer `px` i `cm`, `px` i `%` itd.).

#### Napomena

Izrazi sa relativnim vrednostima (procenti) koji se formiraju korišćenjem operatora bilo kojeg pretprocesorskog jezika (Sass, Less...) imaju jedan veoma značajan nedostatak. U vanila CSS-u (*čist CSS se veoma često naziva vanila CSS*), vrednost izražena u procentima odnosi se na neku drugu veličinu definisanu na nekom od roditeljskih elemenata. Takva relativna vrednost se u CSS-u proračunava od strane browsera, tokom izgradnje stranice. S obzirom na to da se kod pretprocesorskih jezika za stilizovanje prevodi izvan web pregledača, oni ni na koji način ne mogu znati proračunate vrednosti koje su u CSS-u definisane relativnim jedinicama. Stoga upotreba procentualnih vrednosti unutar CSS funkcije `calc()` predstavlja veoma moćan alat, koji nikada neće biti na raspolaganju pretprocesorskim jezicima. Tako se u vanila CSS-u može napisati ovako nešto:

```
width: calc(100% - 80px);
```

Ovakva linija koda imaće potpuno drugačiji efekat od veoma slične linije Less koda:

```
width: 100% - 80px;
```

U čistom CSS-u, vrednost od 100% biće tretirana na značajno drugačiji način, tako što će prvo biti proračunata njena stvarna vrednost u zavisnosti od roditeljske vrednosti na koju se odnosi, a tek onda će biti obavljeno oduzimanje. Sa druge strane, pretprocesorski jezici će ovakav izraz jednostavno pretvoriti u vrednosti od 20%.

## Importovanje

Less i Sass jezici veoma su slični kada je reč o postizanju modularizacije koda za stilizovanje. I jedan i drugi nadograđuju osnovnu at-rule izjavu CSS jezika – `@import`. Stoga, kako bi se stilizacija napisana CSS ili Less jezikom uključila u neki drugi Less fajl, dovoljno je napisati sledeće:

```
@import "common";
```

Na ovaj način, stilizacija sadržana unutar fajla `common.less` biće uključena u fajl sa ovakvom linijom. Bitno je primetiti da je prilikom navođenja naziva fajla moguće izostaviti ekstenziju ukoliko je reč o `.less` fajlu. U slučaju definisanja `.css` fajla, Less će takvu liniju ostaviti u svom izvornom obliku, pa će samo importovanje obaviti browser (slika 25.15).



Slika 25.15. Primer importovanja u Lessu

Analizom slike 25.15. može se razumeti način na koji funkcioniše Less importovanje. Unutar fajla `styles.less` obavljeno je importovanje dva fajla: `common.less` i `common.css`. Pri tome je prilikom importovanja Less fajla njegova ekstenzija izostavljena, zato što njeno navođenje nije obavezno. Bitno je primetiti kako izgleda finalni CSS kod koji je dobijen prevođenjem Less koda. Less prevodilac ignoriše liniju za importovanje fajla sa ekstenzijom `.css`, pa je ona u svom originalnom obliku prebačena u CSS fajl.

## Rezime

- Less, odnosno **Leaner Style Sheets**, jeste pretprocesorski jezik za stilizovanje koji se, uz Sass, svrstava u grupu najpopularnijih jezika ove vrste.
- Less je implementiran korišćenjem JavaScript jezika, tako da se prevođenje obavlja korišćenjem browsera ili node.js izvršnog okruženja.
- Pisanje Less koda obavlja se unutar fajlova sa `.less` ekstenzijom.
- Svaki CSS kod ujedno je i potpuno validan Less kod, zato što Less predstavlja nadogradnju CSS jezika.
- Promenljive se u Lessu kreiraju korišćenjem karaktera at (`@`), dok se naziv i vrednost razdvajaju karakterom dve tačke (`:`).
- Less promenljive poseduju oblast važenja, pa se tako promenljiva prvo potražuje unutar tekućeg bloka, a zatim globalno.
- Pojam interpolacije promenljivih predstavlja umetanje njihove vrednosti u već postojeći tekstualni sadržaj; na taj način se vrednosti Less promenljivih mogu koristiti za definisanje selektora, naziva CSS svojstava, medija upita...

- Interpolacija vrednosti promenljive unutar selektora i naziva CSS svojstava obavlja se korišćenjem konstrukcije `@{variable_name}`, gde je `variable_name` naziv Less promenljive.
- Less omogućava gnežđenje stilizacije, čime se izbegava korišćenje određenih selektora zasnovanih na relacijama i dobija mnogo pregledniji kod za stilizovanje.
- Less mixini omogućavaju uključivanje CSS deklaracija iz jednog CSS opisa u neki drugi; bilo koji CSS opis može postati mixin.
- Less mixini *pozivaju* se korišćenjem zagrada nakon kojih se navodi karakter tačka i zapeta (;).
- Less mixini omogućavaju definisanje grupa srodnih vrednosti koje se drugačije nazivaju mape.
- Less dozvoljava korišćenje operatora, i to prilikom definisanja vrednosti CSS svojstava ili Less promenljivih.
- Less omogućava importovanje stilizacije definisane unutar nekih drugih Less fajlova, i to korišćenjem `@import` direktive; pri tome nije neophodno navoditi `.less` ekstenziju.
- Less prevodilac ignoriše importovanje fajlova sa `.css` ekstenzijom, pa se takve linije u CSS prevode u izvornom obliku.

