

Filteri

CSS poseduje veoma bogat skup filtera, koji se mogu koristiti za primenu grafičkih efekata nad elementima. CSS filteri se primarno koriste kako bi se prilagodio prikaz slika, pozadina ili okvira. Ipak, prednjači upotreba filtera za prilagođavanje vizualnog izgleda slika, bez obzira na to da li je reč o pozadinskim slikama ili onim koje se dodaju korišćenjem `img` elementa. Pored toga, efekat različitih filtera najbolje se može videti na fotografijama, pa će upravo zbog toga u nastavku lekcije svi primeri podrazumevati korišćenje jedne fotografije koja će na stranici biti prikazana korišćenjem `img` elementa.

Za sve one koji su koristili društvenu mrežu Instagram ili su nekada obavljali osnovnu tonsku obradu fotografija, pojam filtera nije nepoznanica. Tako CSS filteri omogućavaju promenu kontrasta, osvetljenosti, intenziteta boje, nijanse i još mnogo toga.

CSS svojstvo koje omogućava definisanje filtera jeste **filter**.

Filteri se definišu korišćenjem različitih CSS funkcija (tabela 10.1).

CSS funkcija	Opis
<code>blur()</code>	definiše zamućenje
<code>brightness()</code>	koristi se za podešavanje osvetljaja
<code>contrast()</code>	koristi se za podešavanja kontrasta
<code>drop-shadow()</code>	definiše pozadinsku (<i>drop</i>) senku
<code>grayscale()</code>	obavlja konvertovanje svih boja u nijanse sive (crno-beli prikaz)
<code>hue-rotate()</code>	utiče na nijansu boja
<code>invert()</code>	vrši invertovanje svih boja
<code>opacity()</code>	dodaje providnost
<code>saturate()</code>	utiče na zasićenost boja
<code>sepia()</code>	definiše <u>sepia</u> efekat

Tabela 10.1. Različiti CSS filteri

Pre nego što budu demonstrirani različiti efekti CSS filtera, biće prikan izvorni izgled fotografije koja će poslužiti za demonstraciju efekta filtera (slika 10.1).



Slika 10.1. Slika bez filtera

blur()

Funkcijom `blur()` postiže se zamućenje elementa. Vrednost se definiše u pikselima, pri čemu više vrednosti predstavljaju jači efekat zamućenja i obrnuto (slika 10.2):

```
filter: blur(1px);
```

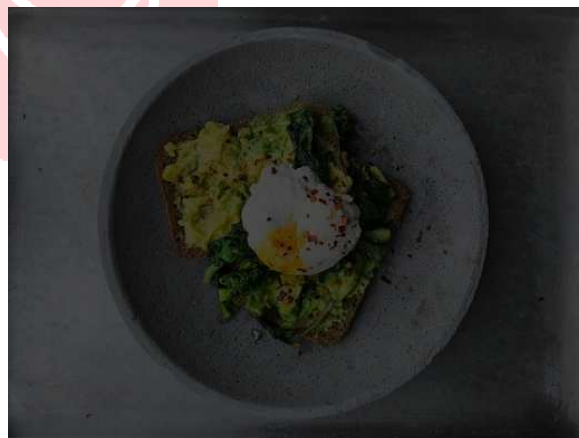


Slika 10.2. Efekat blur filtera

brightness()

Funkcijom `brightness()` može se definisati osvetljaj elementa. Vrednost ove funkcije se može iskazati na dva načina:

- upotrebom **numeričkih vrednosti** bez jedinica; 0 označava potpuno odsustvo boje; vrednost 1 je originalni osvetljaj; vrednosti preko 1 stvaraju efekat preosvetljenosti;
- upotrebom **procenata**; vrednost 0% označava potpuno odsustvo boje; vrednost 100% označava osvetljaj koji slika podrazumevano ima; vrednosti preko 100% stvaraju efekat preosvetljenosti.



Slika 10.3. filter: brightness(0.3); ili filter: brightness(30%);

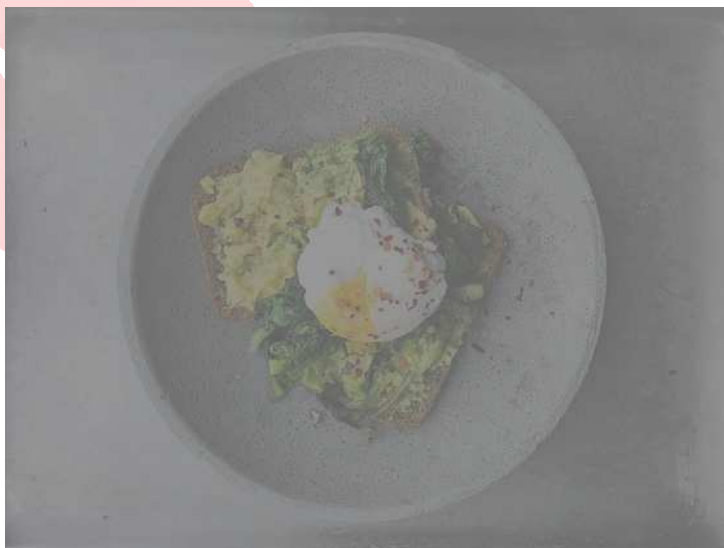


Slika 10.4. *filter: brightness(1.3); ili filter: brightness(130%);*

contrast()

Funkcija `contrast()` omogućava uticaj na kontrast elementa. Definisanje vrednosti ove CSS funkcije obavlja se na identičan način kao i kod funkcije `brightness()`:

- upotrebom **numeričkih vrednosti** bez jedinica; 0 označava potpuno odsustvo kontrasta, što će stvoriti potpuno taman prikaz; vrednost 1 predstavlja originalni kontrast; vrednostima preko 1 postiže se povećanje kontrasta;
- upotrebom **procenata**; vrednost 0% označava potpuno odsustvo kontrasta; vrednost 100% predstavlja originalni kontrast; vrednostima preko 100% postiže se povećanje kontrasta.



Slika 10.5. *filter: contrast(20%); ili filter: contrast(0.2);*



Slika 10.6. filter: contrast(120%); ili filter: contrast(1.2);

drop-shadow()

U jednoj od prethodnih lekcija već je bilo reči o različitim načinima za kreiranje senki korišćenjem CSS-a. Još jedan od načina za dobijanje senki jeste korišćenje posebne filter funkcije – `drop-shadow()`. Ova funkcija prihvata gotovo identične vrednosti kao i svojstvo `box-shadow`:

drop-shadow(offset-x offset-y blur-radius color)

Korišćenje ovog filtera ilustrovano je slikom 10.7.



Slika 10.7. filter: drop-shadow(16px 16px 10px gray);

grayscale()

CSS filter koji se može koristiti za dobijanje crno-belih prikaza je `grayscale()`. Vrednost se može definisati dvojako:

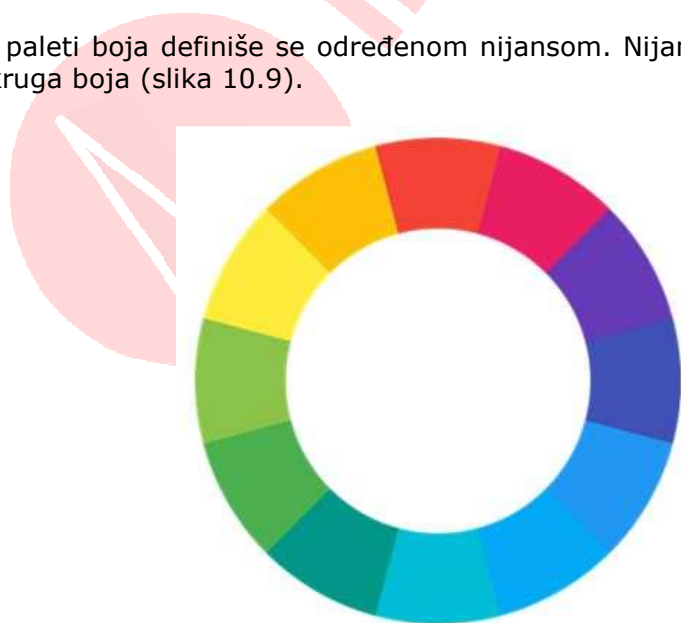
- upotrebom **numeričkih vrednosti** bez jedinica; 1 označava najjači intenzitet filtera, dok vrednost 0 ne stvara nikakve vizualne izmene na elementu;
- upotrebom **procenata**; 100% označava najjači intenzitet filtera, dok vrednost 0% ne stvara nikakve vizualne izmene na elementu.



Slika 10.8. `filter: grayscale(1);` ili `filter: grayscale(100%);`

hue-rotate()

Svaka boja u paleti boja definiše se određenom nijansom. Nijansa se veoma često izražava korišćenjem kruga boja (slika 10.9).



Slika 10.9. Točak sa nijansama boja

Upravo zbog ove činjenice, funkcija `hue-rotate()` prihvata vrednost izraženu u procentima:

```
filter: hue-rotate(180deg);
```

Efekat ovakvog filtera ilustrovan je slikom 10.10.



Slika 10.10. Efekat `hue-rotate()` filtera

invert()

CSS omogućava invertovanje svih boja nekog prikaza. Kako bi se obavilo invertovanje jedne boje, neophodno je vrednost svakog od njena tri kanala (R, G, B) oduzeti od vrednosti 255.

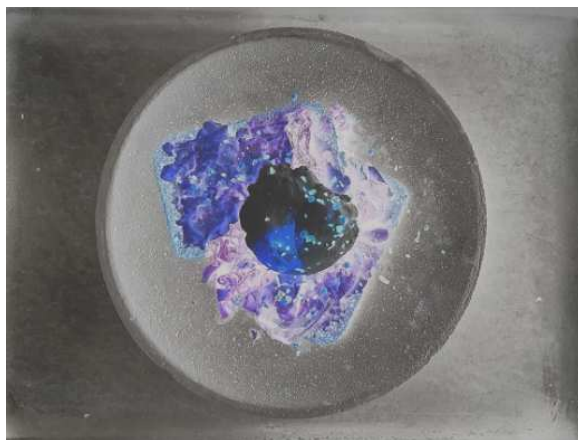
Funkcija `invert()` može da prihvati dva tipa vrednosti:

- numeričke vrednosti bez jedinica; vrednost 1 označava potpuno invertovanje; vrednost 0 ne menja prikaz; sve vrednosti između definišu odgovarajući stepen *invert* efekta;
- procentualne vrednosti; vrednost 100% označava potpuno invertovanje; vrednost 0% ne menja prikaz; sve vrednosti između definišu odgovarajući stepen *invert* efekta.

Primer upotrebe `invert()` funkcije može da izgleda ovako:

```
filter: invert(80%);
```

Prikazana linija proizvodi efekat kao na slici 10.11.



Slika 10.11. Efekat invert () filtera

opacity()

CSS funkcija `opacity()` koristi se za definisanje providnosti pozadine elementa. Kao i veliki broj već prikazanih funkcija, može da prihvati dve vrste vrednosti:

- numeričke vrednosti bez jedinica; vrednost 0 označava potpunu providnost; vrednost 1 označava odsustvo providnosti; sve vrednosti između definišu odgovarajući stepen providnosti;
- procentualne vrednosti; vrednost 0% označava potpunu providnost; vrednost 100% označava odsustvo providnosti; sve vrednosti između definišu odgovarajući stepen providnosti.

Primer korišćenja `opacity()` filtera je sledeći:

```
filter: opacity(40%);
```

Efekat `opacity()` filtera ilustrovan je slikom 10.12.



Slika 10.12. Efekat opacity() filtera

saturate()

Intenzitet boje na nekoj fotografiji drugačije se naziva zasićenost (*saturation*). Za uticanje na zasićenost boja, CSS poseduje istoimenu funkciju – `saturate()`. Ova funkcija može da prihvati sledeće vrednosti:

- numeričke vrednosti bez jedinica; vrednost 0 označava najmanju zasićenost, odnosno odsustvo boje; vrednost 1 definiše izvornu zasićenost, tako da ova vrednost ne proizvodi razliku u odnosu na originalni prikaz; vrednosti preko 1 linearno pojačavaju efekat zasićenosti;
- procentualne vrednosti; vrednost 0% označava najmanju zasićenost, odnosno odsustvo boje; vrednost od 100% definiše izvornu zasićenost, tako da ova vrednost ne proizvodi razliku u odnosu na originalni prikaz; vrednosti preko 100% linearno pojačavaju efekat zasićenosti.

Slike 10.13. i 10.14. ilustruju efekat `saturate()` filtera.



Slika 10.13. filter: saturate(200%); ili filter: saturate(2);



Slika 10.14. filter: saturate(0%); ili filter: saturate(0);

sepia()

Sepia je jedan od veoma popularnih tonskih efekata koji se primenjuju nad fotografijama. Originalno se odnosi na proces koji se primenjivao nad crno-belim fotografijama, kako bi one dobile toplije tonove i postale otpornije i dugovečnije. U modernoj, digitalnoj fotografiji, ovaj efekat je doživeo pravu renesansu, a koristi se kako bi se dobio efekat starih fotografija. Inače, efekat *sepia* je dobio naziv po crvenobraon pigmentu koji je karakterističan za morsku životinju sipu.

Sepia efekat se u CSS-u može postići upotrebom funkcije `sepia()`. Ova funkcija prihvata dvojako definisanje vrednosti:

- numeričke vrednosti bez jedinica; vrednost 1 označava maksimalni sepia efekat; vrednost 0 označava potpuno odsustvo efekta; vrednosti između 0 i 1 definišu proporcionalnu količinu sepije;
- procentualne vrednost; vrednost 100% označava maksimalni sepia efekat; vrednost 0% označava potpuno odsustvo efekta; vrednosti između 0 i 1 definišu proporcionalnu količinu sepije.

Primer korišćenja sepia filtera ilustrovan je slikom 10.15.



Slika 10.15. filter: sepia(50%); ili filter: sepia(0.5);

Pitanje

Efekat zamućenosti može se dobiti korišćenjem filtera:

- **blur()**
- `fade()`
- `opacity()`
- `shadow()`

Objašnjenje:

Funkcijom `blur()` postiže se zamućenje elementa.

Rezime

- CSS filteri se primarno koriste kako bi se prilagodio prikaz slika, pozadina ili okvira.
- CSS filteri omogućavaju promenu kontrasta, osvetljenosti, intenziteta boje, nijanse i još mnogo toga.
- CSS svojstvo koje omogućava definisanje filtera jeste `filter`.
- Funkcijom `blur()` postiže se zamućenje elementa.
- Funkcijom `brightness()` može se definisati osvetljaj elementa.
- Funkcija `contrast()` omogućava uticaj na kontrast elementa.
- Filteri se mogu koristiti i za definisanje senke elementa, upotrebom CSS funkcije `drop-shadow()`.
- CSS filter koji se može koristiti za dobijanje crno-belih prikaza je `grayscale()`.
- Korišćenjem CSS funkcije `hue-rotate()` moguće je uticati na nijansu slike.
- CSS omogućava invertovanje svih boja nekog prikaza korišćenjem funkcije `invert()`.
- CSS funkcija `opacity()` koristi se za definisanje providnosti pozadine elementa.
- Intenzitet boje na nekoj fotografiji drugačije se naziva zasićenost; za uticanje na zasićenost boja, CSS poseduje istoimenu funkciju – `saturate()`.
- Sepia efekat se u CSS-u može postići upotrebom funkcije `sepia()`.

