SVG

Moderni sajtovi ne mogu se zamisliti bez vizuelnih grafičkih elemenata kao što su logoi, ikonice, pozadinske slike, grafikoni, ilustracije i slično. Može se reći da na današnjim sajtovima preovlađuje grafički sadržaj, koji u većini situacija na mnogo bolji i pregledniji način može da prenese željenu poruku posetiocu nego „čist" tekst.

Na web sajtovima se direktno može prikazivati grafika u dva različita oblika, pa otuda postoji podela na sledeće dve vrste grafike:

- rasterska grafika
- vektorska grafika

U računarskoj grafici, raster je pravougaona mreža koju formira veliki broj piksela. Svaki piksel poseduje tačno utvrđenu lokaciju i boju. Skupom velikog broja piksela različitih boja, formira se rasterska slika. Popularni rasterski formati su bitmape (`.bmp`), PNG (`.png`), JPEG (`.jpg`) i GIF (`.gif`).

Za razliku od rastera, koji slike prikazuje pomoću mreže piksela i njihovih informacija, vektorski crteži funkcionišu po sasvim drugačijem principu, koji se može nazvati matematički, s obzirom na to da su vektori deo matematike. Kod vektorske grafike pojam piksela ne igra nikakvu ulogu, jer vektorska grafika može biti bilo koje veličine.

Bitno je razumeti da vektorska grafika nije zamena za rastersku. Naime, obe vrste grafike imaju svoja jasno definisana mesta primene na modernom webu. Na primer, fotografije se ne mogu prikazati u vektorskom formatu. Ipak, većina ostalih grafičkih elemenata može, pa se tako danas vektori uglavnom koriste za prikaz logoa, ikonica, različitih grafikona, dijagrama...

Zbog važnosti koju vektorska grafika ima na današnjem webu, ova i naredna lekcija kursa će biti posvećene radu sa vektorskom grafikom prilikom kreiranja modernih sajtova.

Coffee Shop web sajt

Upoznavanje vektorske grafike i različitih načina za njenu integraciju unutar web sajtova omogućiće nam da dodatno poboljšamo sajt koji razvijmo od početka ovoga kursa. Tako ćemo biti u mogućnosti da logo i sve ikonice, odnosno elemente koji su sada realizovani kao rasteri zamenimo vektorskim elementima, koje ćemo moći da stilizujemo korišćenjem CSS jezika i kojima ćemo moći dinamički da rukujemo upotrebom jezika JavaScript.

Šta je vektorska grafika?

Vektori su posebna vrsta grafike, koja se ne oslanja na pojam piksela i njihovih pozicija i boja. Prikaz vektorske grafike definiše se specifičnim sadržajem vektorskih fajlova, koji čuvaju uputstva o tome kako je potrebno da grafika definisana fajlom izgleda na displeju korisničkog uređaja. Tako se unutar fajlova vektorske grafike definišu tačke, oblici, putanje, linije i razni drugi elementi kojima se formira prikaz vektorske grafike.

Copyright © Link group

Sadržaj vektorskih fajlova ne može se direktno preneti na neki izlazni uređaj za prikaz slike (npr. displej kompjutera). Kod rasterske grafike je tako nešto moguće, zato što rasteri svoje osobine dele sa kompjuterskim displejima. I jedni i drugi se sastoje iz mreže piksela različitih boja, pa je upravo zbog toga raster veoma lako prikazati, mapiranjem piksela grafike i korisničkog displeja.

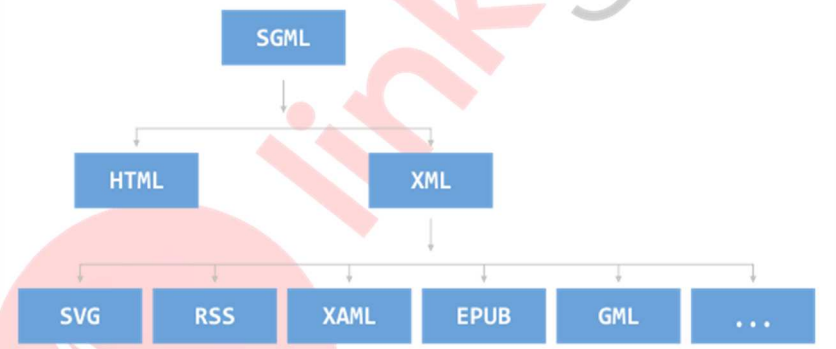
Prikaz vektora zahteva razumevanje sadržaja vektorskih fajlova, pa otuda proces prikaza vektora uvek zahteva posredovanje neke softverske komponente. Na webu, takvu komponentu predstavljaju web pregledači. Oni su zaduženi za to da razumeju sadržaj fajlova kojima se predstavlja vektorska grafika i da na osnovu onoga što u takvim fajlovima *piše* formiraju prikaz na displeju.

Nešto komplikovaniji mehanizam prikaza vektorskih fajlova, sa druge strane, donosi brojne prednosti:

- fajlovi vektorske grafike zauzimaju neuporedivo manje prostora;
- zbog svoje nevezanosti za pojam piksela, vektorska grafika može biti bilo koje veličine;
- vektorska grafika nema problema sa gubitkom kvaliteta prilikom povećavanja ili smanjivanja.

Šta je SVG?

Na webu je vektorsku grafiku moguće koristiti u SVG formatu. SVG je skraćenica za pojam *Scalable Vector Graphics*, što je u suštini opisni jezik zasnovan na XML-u koji se koristi za predstavljanje vektorske grafike.



Slika 13.1. Razvojni put opisnih jezika za markiranje podataka

Sa slike 13.1. može se videti pozicija SVG-a u stablu različitih jezika koji su nastali na osnovi izvornog SGML jezika. Upravo zbog činjenice da dele zajedničkog roditelja, HTML i SVG jezici su veoma slični. Tako se može reći da je i SVG jezik oznaka i da poseduje veliki broj različitih elemenata kojima se kreiraju delovi vektorske grafike. Na primer, sledeći SVG kod ilustruje kreiranje jednog pravougaonika:

```
<rect x="10" y="10" width="200" height="100" fill="#4D4D4D"
stroke="#4F95FF" stroke-width="3" />
```

Odmah je moguće primetiti sličnosti sa HTML jezikom. Za kreiranje pravougaonika iskorišćen je element `rect`, koji je kreiran korišćenjem istoimenog samozatvarajućeg taga. To znači da, baš kao i HTML, SVG poznaje pojmove elemenata i tagova. U prikazanoj liniji SVG koda se može videti i upotreba različitih atributa, kojima se definišu osobine pravougaonika. Iz toga se može uvideti još jedna sličnost HTML i SVG jezika – postojanje atributa.

Copyright © Link group

Prednosti SVG-a u odnosu na rastersku grafiku

Već je rečeno da se vektori ne mogu koristiti za prikaz svih vrsta grafika. Na primer, fotografije se ne mogu predstaviti vektorski, zato što ne postoji model kojim bi se prikazi na fotografijama podveli pod matematička pravila vektorske grafike. Ipak, mnogi drugi elementi današnjih sajtova se mogu predstaviti korišćenjem SVG jezika, a među njima se izdvajaju logoi, ikonice, grafikoni, dijagrami...

Na primer, jedan logo na web sajtu moguće je predstaviti korišćenjem SVG-a, ali i korišćenjem rastera. Ipak, korišćenje SVG grafike donosi brojne prednosti:

- mala veličina fajla, što na webu igra veoma važnu ulogu;
- mogućnost definisanja bilo koje dimenzije;
- odsustvo pikselizacije, odnosno gubitka kvaliteta prilikom uvećavanja grafike.

Poslednju prednost najlakše je uočiti (slika 13.2).

Slika 13.2. Razlika između vektorskog (levo) i rasterskog (desno) prikaza logoa

Slika 13.2. ilustruje jedan isti logo, ali u dva različita formata. Na levoj polovini slike prikazan je vektorski logo, predstavljen SVG jezikom. Na desnoj polovni slike je isti takav logo, ali u rasterskom obliku. Uvećanjem preko svojih originalnih dimenzija, rasterski logo gubi na kvalitetu.

Dodavanje SVG-a u HTML dokument

Vektori se kreiraju korišćenjem programa za obradu i kreiranje vektorske grafike. Najpopularniji program takve vrste je svakako Adobe Illustrator. Ipak, pored njega, postoje i brojni drugi programi koji se mogu koristiti za obavljanje istog posla – Corel Draw, Inkscape, Sketch, Xara Xtreme, Skencil...

Vektorska grafika u SVG obliku se veoma retko, odnosno gotovo nikada ne kreira kao i HTML kod – direktnim pisanjem tagova, elemenata i atributa, već korišćenjem nekog od spomenutih specijalizovanih programa. Vektor koji se kreira korišćenjem takvih programa zatim se izvozi u SVG oblik.

U nastavku ove lekcije, najpre ćemo se upoznati sa različitim načinima za integraciju već pripremljene vektorske SVG grafike u HTML dokument. Nakon upoznavanja principa za integraciju, načelno ćemo se upoznati i sa strukturom SVG dokumenata.

Copyright © Link group

SVG fajl za integraciju

U prilogu ove lekcije nalazi se SVG fajl (fajl sa ekstenzijom `.svg`) koji će u narednim primerima biti iskorišćen za demonstraciju integracije SVG-a u HTML.

Korišćenje elementa img za integraciju SVG-a

Najjednostavniji način za integraciju SVG fajla u HTML dokument jeste korišćenje `img` elementa:

```
<img src="img/logo.svg" alt="coffee shop logo">
```

Za vrednost `src` atributa `img` elementa postavljena je putanja na kojoj se nalazi SVG fajl. Veličina ovako dodate SVG vektorske grafike zavisiće od dimenzija koje su definisane prilikom kreiranja SVG fajla unutar nekog od programa za kreiranje vektorske grafike (*nešto kasnije će biti prikazano kako se definisanje veličine postiže unutar samog SVG koda*). Veličina ovako uključene SVG grafike može se menjati kao i veličina bilo koje druge slike – upotrebom CSS svojstava `width` i `height`. Ipak, za razliku od rasterskih slika, ovakva vektorska grafika se može povećavati bez bojazni da će doći do gubitka kvaliteta prikaza.

SVG kao pozadina elementa

SVG grafika se može postaviti za pozadinu bilo kog HTML elementa korišćenjem CSS svojstava `background` ili `background-image`. HTML element za čiju pozadinu će biti postavljen SVG će izgledati ovako:

```
<div id="my-div"></div>
```

Stilizacija za postavljanje SVG-a kao pozadine:

```
#my-div{
    width: 300px;
    height: 300px;
    background: url(img/logo.svg);
    }
```

SVG koji je iskorišćen na ovaj način imaće sve osobine klasičnih pozadinskih slika. Drugim rečima, on će inicijalno imati veličinu koja je definisana prilikom kreiranja SVG dokumenta. Takođe, SVG grafika će biti ponovljena ukoliko je manja od elementa za čiju pozadinu se postavlja. Za kontrolisanje SVG pozadinske grafike moguće je koristiti poznati skup CSS svojstava:

- `background-repeat` – za kontrolu ponavljanja
- `background-position` – za kontrolu pozicije
- `background-size` – za kontrolisanje veličine

Copyright © Link group

Korišćenje elementa object za integraciju SVG-a

Jedna od osnovnih prednosti SVG grafike, pored već pobrojanih, jeste mogućnost stilizacije korišćenjem CSS-a i manipulacije korišćenjem JavaScripta. Prethodna dva pristupa za integraciju SVG grafike u HTML poseduju jedan veliki nedostatak. Sadržajem grafike koja se na prikazane načine integriše u HTML nije moguće rukovati korišćenjem JavaScript jezika. Upravo zbog toga, postoji još jedan način za integraciju SVG-a u HTML, koji podrazumeva korišćenje HTML elementa `object`:

```
<object type="image/svg+xml" data="img/logo.svg"></object>
```

Na `object` elementu je potrebno definisati dva atributa:

- `type` – definiše tip objekta koji se uključuje u dokument; vrednost je potrebno postaviti na `image/svg+xml`
- `data` – definiše putanju na kojoj se nalazi `.svg` fajl

Element `object` omogućava da se definiše i rezervni mehanizam za prikaz grafike, koji će se aktivirati ukoliko web pregledač ne podržava `object` element:

```
<object type="image/svg+xml" data="img/logo.svg">
<img src="img/logo.png" alt="logo">
</object>
```

Sada je unutar `object` elementa definisan i jedan `img` element, kojim će, u slučaju da pregledač ne poznaje `object` element, biti prikazana rasterska varijanta logoa.

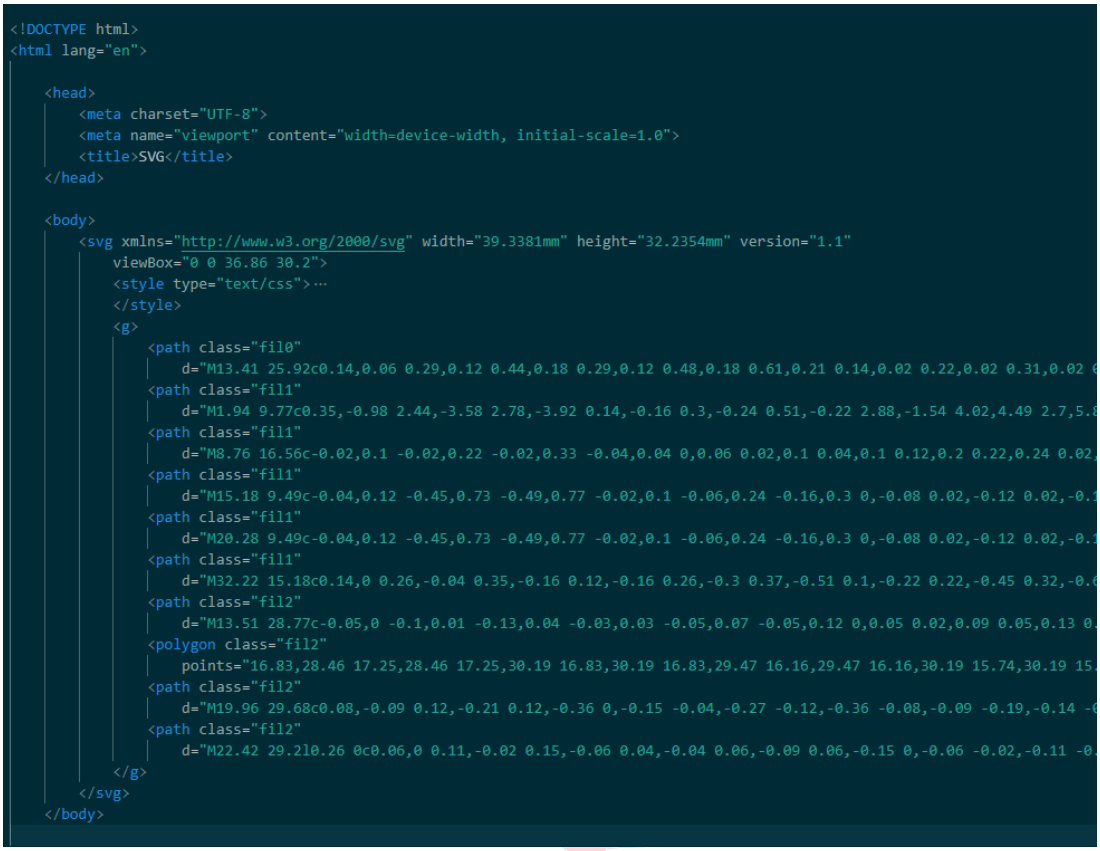
Nešto kasnije biće prikazano kako se SVG grafici uključenoj korišćenjem `object` elementa može pristupiti programabilno.

Smeštanje SVG koda direktno unutar HTML dokumenta

Najprimitivniji način za uključivanje SVG grafike u HTML dokument jeste smeštanje SVG koda direktno unutar HTML dokumenta. Iako je reč o najprimitivnijem pristupu, to je mehanizam za integraciju koji omogućava najveću slobodu pri rukovanju sadržajem SVG grafike.

Direktno smeštanje SVG koda unutar HTML dokumenta podrazumeva kopiranje kompletnog tekstualnog sadržaja iz fajla sa ekstenzijom `.svg` unutar HTML dokumenta. Nakon dodavanja SVG koda, HTML dokument može izgledati kao na slici 13.3.

Copyright © Link group



Slika 13.3. SVG kod u HTML dokumentu

Kompletan kod koji se nalazio unutar zasebnog fajla sa ekstenzijom `.svg` sada je smešten direktno unutar `body` elementa HTML dokumenta.

Osnovna struktura i sintaksa SVG jezika

S obzirom na to da je u prethodnom primeru prvi put obavljena direktna integracija SVG koda u HTML dokument, pravi je trenutak da se načelno upoznamo sa strukturom vektorske grafike u SVG obliku. Kao što je rečeno, iako se veoma retko praktikuje direktno pisanje SVG koda, nije zgoreg poznavati osnovnu strukturu SVG grafike.

Osnovni okružujući element koji se koristi za crtanje SVG grafike je `svg`. Ovaj element se može doživeti kao kontejner za razne grafičke elemente koji se mogu iscrtati na stranici, tako da se može povući izvesna paralela između `html` i `svg` elemenata. Na isti način na koji `html` element predstavlja kontejner za sve one tekstualne i semantičke celine stranice, element `svg` je kontejner za elemente vektorske grafike. Osnovni okružujući element SVG grafike u prikazanom primeru izgleda ovako:

```
        <svg xmlns="http://www.w3.org/2000/svg" width="39.3381mm"
height="32.2354mm" version="1.1" viewBox="0 0 36.86 30.2">

            ...

        </svg>
```

Copyright © Link group

Element svg na sebi poseduje veliki broj atributa:

- xmlns - XML prostor imena
- version – verzija jezika
- width – širina SVG grafike
- height – visina SVG grafike
- viewbox – dimenzije i pozicija prostora za crtanje grafike

Sada prvi put možemo da vidimo gde je to bila definisana veličina SVG grafike koju smo u prethodnim primerima na nekoliko različitih načina prikazivali u HTML dokumentu. Veličina SVG grafike unutar HTML dokumenta definisana je atributima widht i height. Ipak, za uspešno razumevanje prikaza SVG grafike neophodno je razumeti i ulogu viewbox atributa. Ovaj atribut poseduje četiri pojedinačne vrednosti:

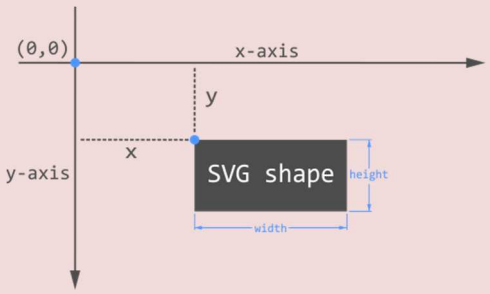
- x – x koordinata referentne tačke
- y – y koordinata referente tačke
- width – širina prostora za crtanje
- height – visina prostora za crtanje

Dimenzionisanje i pozicioniranje svih elemenata unutar SVG-a obavlja se u odnosu na vrednosti definisane viewbox atributom. Atributi width i height koriste se samo da definišu veličinu SVG grafike na stranici. Tako oni nemaju nikakav značaj za elemente koji se nalaze unutar svg elementa, što nije slučaj sa vrednostima viewbox atributa.

Prve dve vrednosti viewbox atributa (x i y) definišu koordinate referentne tačke za crtanje. Referentna tačka za crtanje je uvek u gornjem levom uglu svg elementa. Takva tačka inicijalno ima koordinate (0, 0), a korišćenjem vrednosti x i y viewbox atributa, takve početne koordinate je moguće menjati.

SVG koordinatni sistem

Svi oblici crtaju se unutar granica, odnosno unutar koordinatnog sistema SVG elementa. Koordinatni sistem poseduje dve ose, a jedan SVG oblik unutar takvog sistema prikazan je na slici 13.4.



Slika 13.4. Koordinatni sistem SVG-a

Na slici 13.4. prikazan je pravougaonik unutar SVG koordinatnog sistema. Na pravougaoniku je obeležena krajnja gornja leva tačka i prikazane su njene koordinate. Koordinate tačaka biće intenzivno korišćene u nastavku lekcije, gde će biti predstavljeni osnovni oblici SVG jezika.

Copyright © Link group

Ukoliko se x i y vrednosti unutar viewbox atributa postave na 15 15, tačka u gornjem levom uglu više neće imati koordinate (0, 0), već (15, 15).

Korišćenjem drugog para vrednosti viewbox atributa (width height), definiše se veličina ukupnog prostora unutar koga će moći da se crtaju vektorski elementi. Veličine elemenata koji se crtaju direktno su zavisne od ovih vrednosti. Naime, povećavanjem ovih vrednosti elementi koji se crtaju će postajati sitniji, a smanjivanjem ovih vrednosti oni će postajati krupniji.

Unutar svg elementa u prikazanom primeru su definisana dva direktna potomka:

- style – element unutar koga se definiše stilizacija koja će biti primenjena na SVG elementima
- g – element koji se koristi za grupisanje većeg broja drugih elemenata; može se uporediti sa div elementom u HTML jeziku

Sam SVG jezik definiše brojne elemente za crtanje različitih grafičkih oblika:

- krug (circle)
- elipsa (ellipse)
- linija (line)
- izlomljena linija (polyline)
- poligon (polygon)
- putanja (path)

U zagradama su navedeni nazivi elemenata koji se koriste za crtanje odgovarajućih oblika. U nastavku će biti prikazano crtanje upravo navedenih SVG oblika. Crtanja će biti obavljana unutar ovakvog svg elementa:

```
<svg xmlns="http://www.w3.org/2000/svg" width="560" height="200"
version="1.1" viewBox="0 0 560 200">
        ...
</svg>
```

Pravougaonik (rect)

Element rect koristi se crtanje pravougaonika. Ovaj element poznaje šest različitih atributa kojima se kontrolišu pozicija, veličina i oblik pravougaonika. Ti atributi prikazani su u tabeli 13.1.

Atribut	Opis
x	x-koordinata gornje leve tačke
y	y-koordinata gornje leve tačke
width	širina pravougaonika
height	visina pravougaonika
rx	x-radijus ivica pravougaonika
ry	y radijus ivica pravougaonika

Tabela 13.1. Atributi rect SVG oblika

Copyright © Link group

Uzimajući sve atribute u obzir, sledeći primer ilustruje kreiranje dva pravougaonika:

```
<rect x="15" y="15" width="100" height="60"/>
<rect x="130" y="15" rx="10" ry="10" width="100" height="60"/>
```

Prikazani kod proizvodi efekat kao na slici 13.5.

Slika 13.5. Pravougaonici kreirani korišćenjem rect SVG oblika

Krug (circle)

Element `circle` koristi se za crtanje kruga. Ovaj element poznaje tri atributa, prikazana tabelom 13.2.

Atribut	Opis
r	poluprečnik kruga
cx	x-koordinata centra kruga
cy	y-koordinata centra kruga

Tabela 13.2. Atributi circle SVG oblika

Sledeći primer prikazuje kod za crtanje kruga:

```
<circle cx="45" cy="45" r="45"/>
```

Prikazani kod proizvodi efekat kao na slici 13.6.

Slika 13.6. Krug kreiran korišćenjem circle SVG elementa

Copyright © Link group

Elipsa (ellipse)

Elipsa je geometrijski oblik koji je u osnovu krug kod koga se nezavisno mogu podešavati x i y poluprečnici. Element za crtanje elipse je `ellipse` i on omogućava definisanje atributa prikazanih u tabeli 13.3.

Atribut	Opis
`rx`	x-poluprečnik elipse
`ry`	y-poluprečnik elipse
`cx`	x-koordinata centra elipse
`cy`	y-koordinata centra elipse

Tabela 13.3. Atributi ellipse SVG elementa

Sledeći primer ilustruje kreiranje elipse:

```
<ellipse cx="50" cy="30" rx="50" ry="30" />
```

Prikazani kod proizvodi efekat kao na slici 13.7.

Slika 13.7. Elipsa kreirana korišćenjem ellipse SVG elementa

Pitanje

Za crtanje kruga korišćenjem SVG-a koristi se element:

a) ellipse
b) circle
c) rect
d) square

Objašnjenje:

Element circle koristi se za crtanje kruga.

Copyright © Link group

Linija (line)

Za kreiranje prave linije koristi se element `line`. Linija se definiše početnom i krajnjom tačkom, pa otuda i atributi koje ovaj element poseduje (tabela 13.4).

Atribut	Opis
x1	x-koordinata početna tačke
y1	y-koordinata početna tačke
x2	x-koordinata krajnje tačke
y2	y-koordinata krajnje tačke

Tabela 13.4. Atributi line SVG oblika

Primer koji ilustruje kreiranje jedne linije je sledeći:

```
<line x1="10" y1="10" x2="200" y2="10" stroke-width="3"
stroke="#FF7F7F"/>
```

Prikazani kod proizvodi efekat kao na slici 13.8.

Slika 13.8. Linija kreirana korišćenjem line SVG elementa

U kodu za kreiranje linije, definisana je njena debljina, korišćenjem atributa `stroke-width`, i njena boja, korišćenjem atributa `stroke`.

Izlomljena linija (polyline)

Izlomljene linije su grupe međusobno povezanih pravih linija. Definišu se nizom koordinata tačaka koje grade takvu liniju. Za kreiranje ovakvih linija koristi se element `polyline`, koji poznaje samo jedan atribut, za čiju vrednost se postavljaju koordinate svih tačaka linije.

Atribut	Opis
points	niz tačaka razdvojenih zarezom, koje se definišu x i y koordinatama

Tabela 13.5. Atribut polyline SVG elementa

Sledeći primer ilustruje kreiranje jedne izlomljene linije:

```
<polyline points="30 55,60 25,100 65,140 20,185 70,230 15,280 75, 340
10, 400 85, 460 5" stroke-width="3" stroke="#FF7F7F" fill="none"/>
```

Izlomljene linije spajanjem pravih linija grade geometrijski oblik sa ispunom, tako da je neophodno postaviti `fill` takvog oblika na none, kako bi se videla samo linija. Prikazani kod proizvodi liniju kao na slici 13.9.

Copyright © Link group

Slika 13.9. Izlomljena linija kreirana korišćenjem polyline SVG elementa

Poligon (polygon)

Poligoni su oblici koji se sastoje iz pravih, međusobno povezanih linija. Kao i izlomljene linije, definišu se skupom tačaka određenih x i y koordinatama, ali se početna i krajnja tačka automatski spajaju i tako grade zatvoren oblik. Poligoni se crtaju korišćenjem `polygon` elementa, a skup tačaka se definiše unutar `points` atributa.

Atribut	Opis
`points`	niz tačaka razdvojenih zarezom, koje se definišu x i y koordinatama

Tabela 13.6. Atribut polygon SVG elementa

Sledeći primer ilustruje kod za kreiranje jednog poligona:

```
<polygon  points="20 65,60 5,100 65" stroke-width="2" stroke="#4F95FF"
fill="#4D4D4D"/>
```

Korišćenjem atributa `points`, definisane su koordinate tri tačke. Prva i poslednja tačka biće automatski spojene, pošto je reč o `polygon` obliku. Spajanjem prve i poslednje tačke, dobija se oblik koji je moguće bojiti. To je učinjeno korišćenjem atributa `fill`, a sama boja je definisana u heksadecimalnom obliku.

Prikazani kod proizvodi efekat kao na slici 13.10.

Slika 13.10. Trougao kreiran korišćenjem polygon SVG oblika

Copyright © Link group

Putanja (path)

Putanja je najmoćniji oblik u SVG sistemu. Pomoću njega je moguće crtati praktično sve oblike, od pravougaonika preko krugova i elipsi pa sve do izlomljenih linija i poligona. Međutim, tu nije kraj, pa se korišćenjem putanje mogu crtati Bezjeove krive, kvadratne krive i slično.

Putanja se crta korišćenjem elementa `path`, koji poznaje samo jedan atribut, prikazan tabelom 13.7.

Atribut	Opis
d	niz tačaka razdvojenih zarezom, koje se definišu x i y koordinatama

Tabela 13.7. Atribut path SVG oblika

Vrednost atributa `d` sastoji se iz niza naredbi i parametara. Naredba započinje velikim slovom abecede, nakon čega slede `x` i `y` koordinate. Sledeći primer ilustruje kod za kreiranje jedne Bezjeove krive:

```
<path d="M10 10 C 20 20, 40 20, 50 10" stroke="#4F95FF"
fill="transparent"/>
```

Prikazani kod proizvodi efekat kao na slici 13.11.

Slika 13.11. Bezjeova kriva kreirana korišćenjem path SVG oblika

Stilizovanje SVG-a

Stilizovanje SVG-a obavlja se korišćenjem CSS jezika. Upravo zbog toga se vrlo često kaže da je CSS jezik za stilizovanje HTML, ali i nekih drugih dokumenata. U grupu nekih drugih dokumenata koje je moguće stilizovati korišćenjem CSS-a svrstava se i SVG.

Postoje dva osnovna izazova prilikom definisanja CSS stilizacije SVG elemenata:

gde pisati CSS opise

Ukoliko se za uključivanje SVG grafike koristi `img` element, pozadinska slika ili `object` element, CSS stilizaciju elemenata je neophodno definisati u samom `.svg` dokumentu. Upravo je to obavljeno u primeru `.svg` dokumenta iz ove lekcije. Unutar takvog dokumenta je definisan element `style` sa stilizacijom.

Kada se SVG postavi direktno unutar HTML dokumenta, tada je CSS opise za stilizovanje SVG-a moguće pisati unutar `style` elementa HTML dokumenta.

Copyright © Link group

specifična CSS svojstva koja se koriste za stilizovanje SVG elemenata

Mnoga CSS svojstva su zajednička za HTML i SVG elemente. Ipak, postoje izvesne razlike. Najznačajnija CSS svojstva koja su specifična za SVG su:

- `fill` – koristi se za postavljanje boje pozadine SVG elemenata
- `fill-opacity` – intenzitet neprovidnosti pozadine SVG elemenata; vrednost 0 označava odsustvo neprovidnosti, odnosno potpunu providnost, a vrednost 1 potpunu neprovidnost
- `stroke` – boja okvira SVG elemenata
- `stroke-width` – debljina okvira SVG elemenata

Ova specifična CSS svojstva mogu se koristiti i unutar CSS opisa, ali i kao atributi na SVG elementima:

```
<rect x="15" y="15" width="300" height="100" fill="burlywood" stroke-
width="3" stroke="blueviolet" />
```

Na kraju, SVG elementi na sebi mogu imati `id` i `class` atribute, koji se zatim mogu koristiti za selektovanje takvih elemenata prilikom stilizovanja.

```
        <svg xmlns="http://www.w3.org/2000/svg" width="560" height="200"
version="1.1" viewBox="0 0 560 200">
            <rect id="my-rect" x="15" y="15" width="300" height="100" />
        </svg>
```

Na `rect` element je postavljen `id` sa vrednošću `my-rect`. Tako se ovaj element može stilizovati na sledeći način:

```
#my-rect {
    fill: burlywood;
    stroke-width: 3;
    stroke: blueviolet;
}
```

Kontrolisanje SVG-a korišćenjem JavaScripta

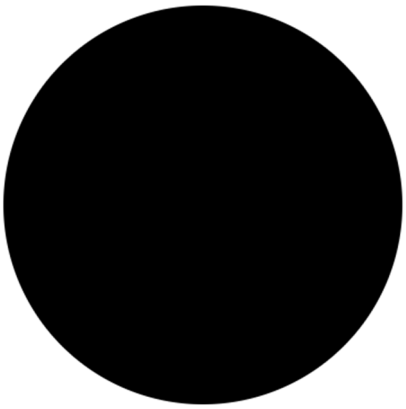
Vrhunac manipulacije SVG elementima ogleda se u njihovom kontrolisanju putem JavaScript jezika. Za početak, bitno je reći da se SVG elementima koji su u HTML uključeni korišćenjem `img` elementa ili `background` CSS svojstva ne može rukovati korišćenjem JavaScript jezika. Stoga, za postizanje programabilne kontrole, neophodno je SVG uključiti korišćenjem `object` elementa ili direktno postaviti u HTML dokument.

Najlakše je programabilno kontrolisati SVG elemente koji se nalaze unutar HTML dokumenta. Takvi elementi su direktno dostupni JavaScript programskom kodu, korišćenjem DOM funkcionalnosti koje su prikazane u prethodnim lekcijama ovog kursa. U nastavku će prvo biti prikazan jedan primer programabilnog rukovanja SVG elementima koji se nalaze direktno unutar HTML dokumenta:

Copyright © Link group

```
        <svg xmlns="http://www.w3.org/2000/svg" width="200"
height="200" version="1.1" viewBox="0 0 200 200">
            <circle id="circle1" cx="90" cy="90" r="90" />
        </svg>

        <script>

            let circle1 = document.getElementById("circle1");

            circle1.addEventListener("click", function(){

                let x = Math.floor(Math.random() * 256);
                let y = Math.floor(Math.random() * 256);
                let z = Math.floor(Math.random() * 256);
                let fill = "rgb(" + x + "," + y + "," + z + ")";

                this.style.fill = fill;

            });

        </script>
```

U prikazanom primeru, direktno unutar HTML dokumenta je postavljen SVG kod sa jednim `circle` elementom. Odmah ispod `svg` elementa, definisan je `script` element sa JavaScript kodom kojim se vrši dinamičko postavljanje boje pozadine nacrtanog kruga. Do reference na SVG element došlo se korišćenjem već poznatih pristupa, kao da je reč o bilo kojem HTML elementu. Zatim je obavljena pretplata na `click` događaj na SVG element. Prilikom klika, aktivira se anonimna funkcija u kojoj se prvo vrši dinamičko generisanje jedne RGB boje. Prilikom svakog novog klika, generiše se drugačija boja, upotrebom metode `random()`. S obzirom na to da metoda `random()` vraća vrednost između 0 i 1, dobijena vrednost se množi sa 256, što je maksimalna vrednost boje svakog kanala pojedinačno. Na kraju je dobijena vrednost zaokružena na prvi manji ceo broj i to je pristup koji je iskorišćen za dobijanje svakog od tri kanala boje. Na kraju je dobijena boja, postavljena kao pozadinska za `circle` element, upotrebom svojstava `style` i `fill`.



Animacija 13.1. Primer dinamičke manipulacije SVG elementom

Copyright © Link group

Kada se SVG grafika unutar HTML-a dodaje korišćenjem elementa `object`, dinamičku manipulaciju je nešto teže postići. Naime, kada se koristi `object` element, SVG kod se ne nalazi direktno unutar HTML dokumenta, pa nije moguće koristiti upravo prikazani pristup. Pre nego što se upotrebi logika koja je već prikazana, neophodno je na neki način doći do reference na SVG kod koji se nalazi u zasebnom fajlu. To je moguće postići na sledeći način:

```
        <object id="svg-object" type="image/svg+xml"
data="img/circle.svg"></object>

        <script>

            let svgObject = document.getElementById("svg-object");

            let svgDoc = svgObject.contentDocument;
            let circle1 = svgDoc.getElementById("circle");

            circle1.addEventListener("click", function () {

                let x = Math.floor(Math.random() * 256);
                let y = Math.floor(Math.random() * 256);
                let z = Math.floor(Math.random() * 256);
                let fill = "rgb(" + x + "," + y + "," + z + ")";

                this.style.fill = fill;

            });
        </script>
```

SVG se sada ne nalazi direktno unutar HTML dokumenta, već unutar zasebnog fajla koji je u HTML uključen korišćenjem `object` elementa. Kako bi se došlo do SVG sadržaja, prvo je obavljeno selektovanje `object` elementa. Nad DOM referencom `object` elementa iskorišćeno je svojstvo `contentDocument`, koje sadrži kompletan unutrašnji sadržaj dokumenta koji je uključen korišćenjem `object` elementa. Ostatak koda je identičan prethodnom primeru.

Napomena

U većini modernih web pregledača ovaj poslednji primer neće funkcionisati ukoliko se HTML dokument i `.svg` fajl nalaze u *lokalu*. Naime, web pregledači ne dozvoljavaju stranicama da preko `file:` protokola pristupaju eksternim fajlovima, što je u ovom slučaju fajl sa `.svg` ekstenzijom. Stoga će u takvim situacijama svojstvo `contentDocument` uvek imati vrednost `null`. Rešenje se ogleda u postavljanju fajlova primera na neki server, udaljeni ili lokalni (WAMP, XAMPP...).

Rezime

- Vektori su posebna vrsta grafike, koja se ne oslanja na pojam piksela i njihovih pozicija i boja, već se prikaz definiše specifičnim sadržajem vektorskih fajlova koji čuvaju uputstva o tome kako je potrebno da grafika definisana fajlom izgleda na displeju korisničkog uređaja.
- Vektori se kreiraju korišćenjem programa za obradu i kreiranje vektorske grafike.

Copyright © Link group

- SVG je skraćenica za pojam *Scalable Vector Graphics*, što je opisni jezik zasnovan na XML-u koji se koristi za predstavljanje vektorske grafike.
- Prednosti SVG grafike u odnosu na rastere su mala veličina fajla, mogućnost definisanja bilo koje dimenzije i odsustvo pikselizacije.
- Najjednostavniji način za integraciju SVG fajla u HTML dokument jeste korišćenje `img` elementa.
- SVG grafika se može postaviti za pozadinu bilo kog HTML elementa korišćenjem CSS svojstava `background` ili `background-image`.
- Integracija SVG-a u HTML može se obaviti i korišćenjem elementa `object`.
- Najprimitivniji način za uključivanje SVG grafike u HTML dokument jeste smeštanje SVG koda direktno unutar HTML dokumenta.
- SVG jezik poznaje pojmove tagova, elemenata i atributa za definisanje vektorske grafike.
- Stilizovanje SVG-a se može obaviti korišćenjem CSS jezika; CSS opisi se mogu smestiti direktno unutar SVG fajla ili unutar HTML dokumenta ukoliko se SVG kod nalazi u HTML dokumentu.
- Programabilno kontrolisanje SVG elemenata se može obaviti korišćenjem funkcionalnosti DOM API-a.

Copyright © Link group