

### 3. Grafică vectorială

Formatul SVG - Scalable Vector Graphics din engleză - "grafică vectorială proporțională", este un limbaj pentru descrierea de imagini 2D folosind XML. Este un standard al organizației W3C a cărui proiectare a început în anul 1999. Permite definirea imaginilor prin 3 metode: text, grafică vectorială și "bitmap-uri" (fișiere în formatul BMP).

Deși există aplicații specializate pentru crearea și editarea de SVG-uri, în acest scop poate fi folosit orice editor text. Vizualizarea unei imagini SVG poate fi realizată cu orice browser modern.

#### Detalii tehnice

În momentul de față SVG-ul are mai multe profiluri pentru a se adapta mai bine la diferite constrângeri. Astfel, profilele "SVG Tiny" și "SVG Basic" au fost create special pentru dispozitivele mobile cu resurse limitate. În același timp, profilul "SVG Print" este destinat mediilor de imprimare a documentelor.

Pentru animarea unei imagini SVG organizația W3C recomandă standardul "SMIL". Pe lângă recomandarea oficială mai există și alte soluții, precum ar fi "ECMAScript".

#### Elemente componente

Principalele elemente din componența unui fișier SVG sunt:

- *Paths*. "Căile" pot fi folosite pentru descrierea conturului unei forme. Conturul poate rămâne gol sau poate fi umplut. Căile pot fi folosite și pentru a specifica zonele de decupare;
- *Forme de bază*. Specificația SVG oferă posibilitatea folosirii următoarelor forme de bază: dreptunghi, cerc, elipsă, linii și poligoane. Acestea pot fi construite și cu ajutorul căilor care au același contur;
- *Text*. Pentru a specifica textul ce apare într-o imagine trebuie folosite elemente de tipul text;
- *Painting*. Se referă la posibilitatea de a umple formele specificate în SVG. Pentru aceasta poate fi folosită o singură culoare, o culoare cu transparență, un gradient sau un model;
- *Culoare*. Proprietatea culoare este folosită pentru specificare culorii;
- *Gradient și model*. Folosite pentru colorarea formelor specificate;
- *Decupare, mascare*. În SVG pot fi folosite zone de decupare sau de mascare;
- *Filtre*. Filtrele descriu diferite efecte aplicate imaginilor;
- *Interactivitate*. O imagine SVG are posibilitatea de a interacționa cu utilizatorul. Astfel, la apăsarea unui buton sau la folosirea mausului pot fi pornite diferite scripturi;
- *Linkuri*. Un document poate conține legături către alte pagini sau elemente din web;
- *Scripting*. Într-un SVG pot fi definite scripturi cu diverse funcții;
- *Animații*. Pentru un SVG pot fi specificate diverse tipuri de animații;

- *Font*. Nu este necesar ca utilizatorul final să aibă deja instalate diferitele seturi de caractere ("fonturi") folosite. Fonturile pot fi incluse în imagine; prin aceasta ele pot fi randate la fel în orice situație;
- *Metadata*. Pentru integrarea mai bună webul semantic oferă și opțiunea specificării de metadata (acestea sunt datele care descriu datele propriu-zise).






## Elementul HTML `<svg>`

Elementul HTML `<svg>` reprezintă un container pentru grafică vectorială SVG.

## Browser Support

În tabelul 3.1 sunt specificate versiunile de browsere începând cu care acestea suportă complet elementul `<svg>`.

Tabelul 3.1. Versiunile de browser ce suportă complet elementul `<svg>`.

Element					
<code>&lt;svg&gt;</code>	4.0	9.0	3.0	3.2	10.1

Tagul `<svg>` dispune de câteva metode pentru a desena linii (segmente), trasee (cale), dreptunghiuri, cercuri, text și imagini grafice.

## Forme SVG

Formatul SVG dispune de câteva elemente pentru descrierea unor forme predefinite care pot fi utilizate de către dezvoltator:

- Dreptunghi `<rect>`
- Cerc `<circle>`
- Elipsă `<ellipse>`
- Linie `<line>`
- Polilinie `<polyline>`
- Poligon `<polygon>`
- Cale `<path>`

## Proprietatea Stroke

Formatul oferă un spectru larg de proprietăți pentru contur:

- stroke
- stroke-width
- stroke-linecap
- stroke-dasharray

Toate proprietățile conturului pot fi aplicate oricărui tip de linie, contur al textului sau contur al figurilor.

Proprietatea stroke definește culoarea liniei, conturului textului ori a figurilor.

### *Exemplu*

```

<svg height="80" width="300">
<g fill="none">
  <line stroke="red" x1="0" y1="10" x2="200" y2="10"/>
  <line stroke="black" x1="0" y1="20" x2="200" y2="20"/>
  <line stroke="blue" x1="0" y1="30" x2="200" y2="30"/>
</g>
</svg>

```



### Proprietatea stroke-width

Proprietatea stroke-width definește grosimea liniei, conturului textului ori a conturului unui element.

#### *Exemplu*

```

<svg height="80" width="300">
<g fill="none">
  <line stroke="red" stroke-width="1", x1="0" y1="10" x2="200" y2="10"/>
  <line stroke="black" stroke-width="2", x1="0" y1="20" x2="200" y2="20"/>
  <line stroke="blue" stroke-width="3", x1="0" y1="30" x2="200" y2="30"/>
</g>
</svg>

```



### Proprietatea stroke-linecap

Proprietatea stroke-linecap definește diferite tipuri de capete de contururi deschise:

#### *Exemplu*

```

<svg height="80" width="300">
<g fill="none" stroke="black" stroke-width="10">
  <line stroke-linecap="butt", x1="10" y1="20" x2="200" y2="20" />
  <line stroke-linecap="round", x1="10" y1="40" x2="200" y2="40" />
  <line stroke-linecap="square", x1="10" y1="60" x2="200" y2="60" />
</g>
</svg>

```



### Proprietatea stroke-dasharray

Proprietatea stroke-dasharray este utilizată pentru a crea linii ștrihpunct:

#### *Exemplu*

```

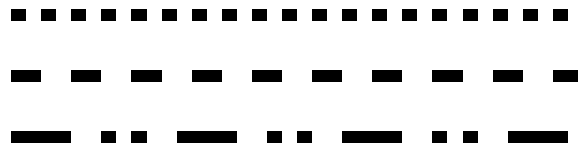
<svg height="80" width="300">
<g fill="none" stroke="black" stroke-width="4">

```

```

<line stroke-dasharray="5,5", x1="10" y1="20" x2="200" y2="20" />
  <line stroke-dasharray="10,10", x1="10" y1="40" x2="200" y2="40" />
  <line stroke-dasharray="20,10,5,5,5,10", x1="10" y1="60" x2="200" y2="60" />
</g>
</svg>

```



## Linie (segment) – <line>

Elementul <line> este utilizat pentru a desena un segment de dreaptă:

<line> definește o linie;

x1 – abscisa x a punctului de început al segmentului de dreaptă;

y1 – ordonata y a punctului de început al segmentului de dreaptă;

x2 – abscisa x a punctului de sfârșit al segmentului de dreaptă;

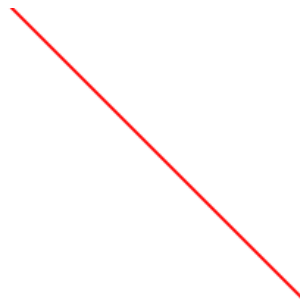
y2 – ordonata y a punctului de sfârșit al segmentului de dreaptă.

### *Exemplu*

```

<svg height="210" width="500">
<line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0); stroke-width:2" />
</svg>

```



## Polilinie – <polyline>

Elementul <polyline> este utilizat pentru a crea linii frânte construite din segmente de drepte conectate prin câteva puncte.

### *Exemplu*

```

<svg height="200" width="500">
<polyline points="20,20 40,25 60,40 80,120 120,140 200,180"
  style="fill:none;stroke:black;stroke-width:3" />
</svg>

```



Atributul *points* conține lista de puncte (perechile coordonatelor x și y) necesare pentru a desena linia frântă.

## Cerc

`<circle>` definește un cerc  
cx – coordonata pe axa x a centrului cercului;  
cy – coordonata pe axa y a centrului cercului;  
r – raza cercului.

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>
```

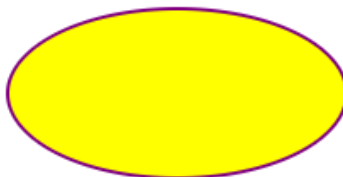
## Elipsă – `<ellipse>`

Elementul `<ellipse>` se utilizează pentru a crea o elipsă.

`<ellipse>` definește o elipsă  
cx – coordonata pe axa x a centrului elipsei;  
cy – coordonata pe axa y a centrului elipsei;  
rx – lungimea razei elipsei pe axa x (raza orizontală – lățimea);  
ry – lungimea razei elipsei pe axa y (raza verticală – înălțimea).

Elipsa reprezintă un caz mai generalizat al cercului. Diferența dintre aceste figuri constă în faptul că la elipsă lungimile razelor pe axele x și y diferă, pe când în cazul cercului acestea sînt egale.

```
<svg height="140" width="500">  
  <ellipse cx="200" cy="80" rx="100" ry="50"  
  style="fill:yellow;stroke:purple;stroke-width:2" />  
</svg>
```



## Dreptunghi - `<rect>`

Elementul <rect> este utilizat pentru a crea dreptunghiuri și figuri derivate a acestuia:

<rect> definește un dreptunghi  
x – coordonata pe axa x a colțului din partea stânga de sus a dreptunghiului;  
y – coordonata pe axa y a colțului din partea stânga de sus a dreptunghiului;  
rx – raza pe axa x (pentru rotunjirea vârfurilor elementului);  
ry – raza pe axa y (pentru rotunjirea vârfurilor elementului);  
width – lățimea dreptunghiului;  
height – înălțimea dreptunghiului.

**Exemplu** <svg width="400" height="180">  
<rect x="50" y="20" rx="20" ry="20" width="150" height="150"  
style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />  
</svg>

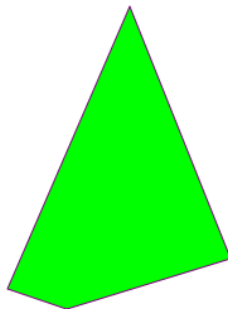


## Poligon – <polygon>

Elementul <polygon> este utilizat pentru a crea elemente grafice care trebuie să conțină cel puțin 3 laturi. Poligoanele sunt create din linii drepte, iar figura reprezintă un contur închis.

<polygon> definește un poligon care conține cel puțin trei laturi  
*points* – conține coordonatele x și y a fiecărui punct din mulțimea punctelor care constituie vârfurile poligonului. Numărul total al punctelor trebuie să fie par.  
*fill-rule* – reprezintă un atribut de prezentare care definește algoritmul utilizat pentru reprezentarea interiorului figurii (metoda de umplere/colorare) și poate lua valorile *evenodd* sau *nonzero*.

<svg height="250" width="500">  
<polygon points="220,10 300,210 170,250 123,234"  
style="fill:lime;stroke:purple;stroke-width:1" />  
</svg>



## Path – <path>

Elementul <path> este utilizat pentru a defini o cale.

<path> definește o cale  
d – parametrul path data reprezintă un set de comenzi care definește calea;  
pathLength – dacă este prezent, calea va fi scalată astfel încât lungimea calculată a căii care trece prin puncte să fie egală cu această valoare;  
transform – lista de transformări.

Parametrul path data poate avea valorile:

M = moveto

L = lineto

H = horizontal lineto

V = vertical lineto

C = curveto

S = smooth curveto

Q = quadratic Bézier curve

T = smooth quadratic Bézier curveto

A = elliptical Arc

Z = closepath

Toate comenzile de mai sus pot fi scrise și cu minuscule. Utilizarea majusculilor semnifică poziționate absolut, utilizarea minusculilor semnifică poziționate relativă.

## Curbele Bézier

Curbele Bézier sunt utilizate pentru a modela linii netede care pot fi scalate la infinit fără a-și pierde din calitate. Pentru a reprezenta o curbă Bézier utilizatorul definește două puncte care indică cele două capete ale curbei și unul sau două puncte de control. O curbă Bézier cu un punct de control poartă numele de curbă Bézier pătratică, iar cele cu două puncte de control – curbe Bézier cubice.

În exemplul de mai jos este prezentată crearea unei curbe Bézier pătratice, unde punctele A și C reprezintă capetele iar punctul B reprezintă punctul de control:

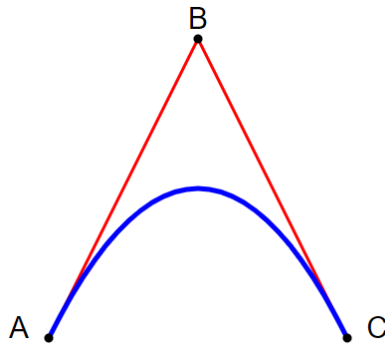
### *Exemplu*

```
<svg height="400" width="450">  
<path id="lineAB" d="M 100 350 l 150 -300" stroke="red"  
stroke-width="3" fill="none" />  
<path id="lineBC" d="M 250 50 l 150 300" stroke="red"  
stroke-width="3" fill="none" />  
<path d="M 100 350 q 150 -300 300 0" stroke="blue"  
stroke-width="5" fill="none" />  
<!-- Mark relevant points -->  
<g stroke="black" stroke-width="3" fill="black">  
<circle id="pointA" cx="100" cy="350" r="3" />  
<circle id="pointB" cx="250" cy="50" r="3" />  
<circle id="pointC" cx="400" cy="350" r="3" />  
</g>  
<!-- Label the points -->  
<g font-size="30" font-family="sans-serif" fill="black" stroke="none"  
text-anchor="middle">  
<text x="100" y="350" dx="-30">A</text>
```

```

<text x="250" y="50" dy="-10">B</text>
<text x="400" y="350" dx="30">C</text>
</g>
</svg>

```



## Text – <text>

Elementul <text> este utilizat pentru a defini un text.

<text> definește un text

x – lista de coordonate pe axa x a caracterelor. Elementul  $n$  din lista de coordonate de pe axa x corespunde caracterului cu numărul de ordine  $n$  din text. Dacă există caractere suplimentare cărora nu le corespunde valori pentru coordonata de pe axa x, acestea sunt plasate după ultimul caracter. Valoare implicită – 0.

y – lista de coordonate pe axa y a caracterelor.

dx – lista valorilor pe axa x cu care se deplasează caracterele relativ la poziția absolută a ultimului caracter desenat;

dy – lista valorilor pe axa y cu care se deplasează caracterele relativ la poziția absolută a ultimului caracter desenat;

rotate – lista unghiurilor de rotație. Valoarea  $n$  din lista de rotație se aplică caracterului  $n$  din textul afișat. Caractere suplimentare nu se rotesc.

### *Exemplu*

```

<svg height="200" width="200">
<text font-size="50" x="0" y="0" fill="red" transform="translate(50,50) rotate(40)">
SVG</text>
</svg>

```

SVG

## Diferența dintre SVG și Canvas

SVG un limbaj de descriere a graficii 2D in XML.

Canvasul permite generarea dinamică a graficii 2D de tip rastru utilizând limbajul JavaScript.



SVG este bazat pe XML care presupune faptul că fiecare element este accesibil cu ajutorul metodelor SVG DOM. Acest fapt oferă posibilitatea atașării evenimentelor JavaScript fiecărui element.

În SVG, fiecare figură reprezintă un obiect iar dacă atributele unui astfel de obiect SVG se modifică, browserul automat redesenează figura.

Canvasul se desenează pixel cu pixel. În canvas odată ce figura este desenată de către browser, acesta pierde legătura cu aceasta. Dacă poziția unei figuri din scenă trebuie modificată, în acest caz trebuie redesenată întreaga scenă.

Tabelul 3.2. Comparații dintre Canvas și SVG

Canvas	SVG
<ul style="list-style-type: none"> <li>– Dependent de rezoluție</li> <li>– Nu suportă evenimentele</li> <li>– Redare proastă a textului</li> <li>– Posibilitatea exportării graficii în formatele .png sau .jpg</li> <li>– Potrivit pentru jocurile cu grafică avansată</li> </ul>	<ul style="list-style-type: none"> <li>– Independent de rezoluție</li> <li>– Suportă evenimentele</li> <li>– Potrivit pentru aplicații cu zone mari de redare (Google Maps)</li> <li>– Redare lentă în scene complexe (utilizare excesivă DOM se răsfrânge asupra performanței)</li> <li>– Nu se potrivește pentru jocuri.</li> </ul>

### Integrarea graficii vectoriale în p5.js Web Editor

Pentru a integra codul `<svg>` în limbajul HTML cu ajutorul editorului web p5.js este nevoie de logarea cu ajutorul contului *Google Account* sau *GitHub* pe platforma respectivă. Pentru a vizualiza fișierele din componența proiectului trebuie accesat butonul cu pictograma de săgeată din partea stângă a ferestrei (fig.3.1).

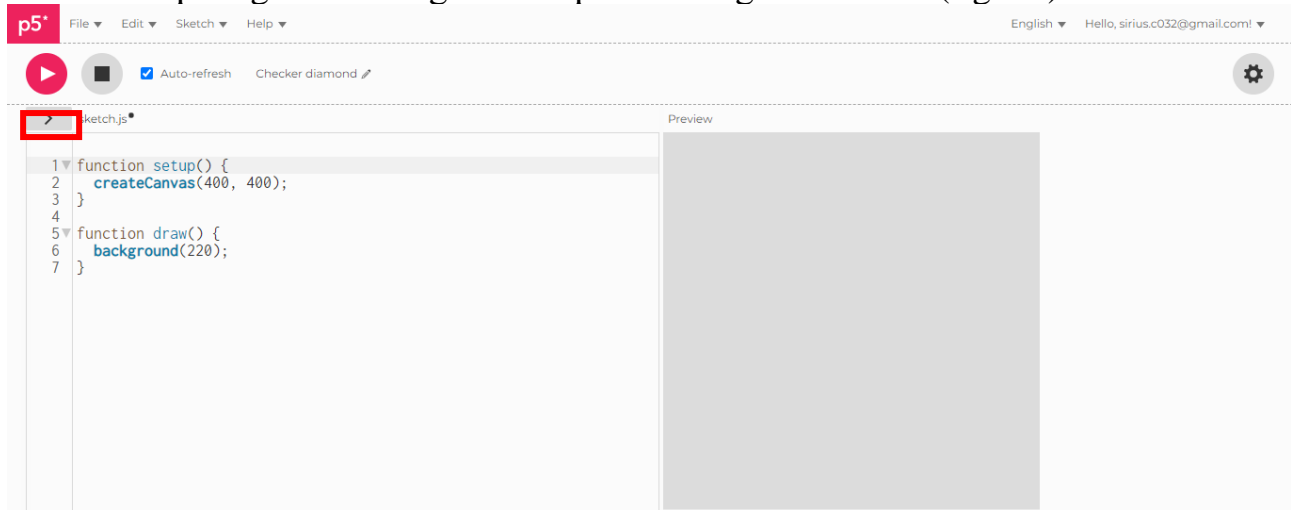


Fig. 3.1. Accesarea ferestrei Sketch Files a editorului p5.js Web Editor.

În urma acțiunii date se deschide fereastra Sketch Files în care se pot vedea fișierele componente ale proiectului: *index.html*, *sketch.js* și *style.css*. Fișierul *index.html* conține codul HTML pentru marcarea interfeței componentelor paginii web a proiectului (fig. 3.2).

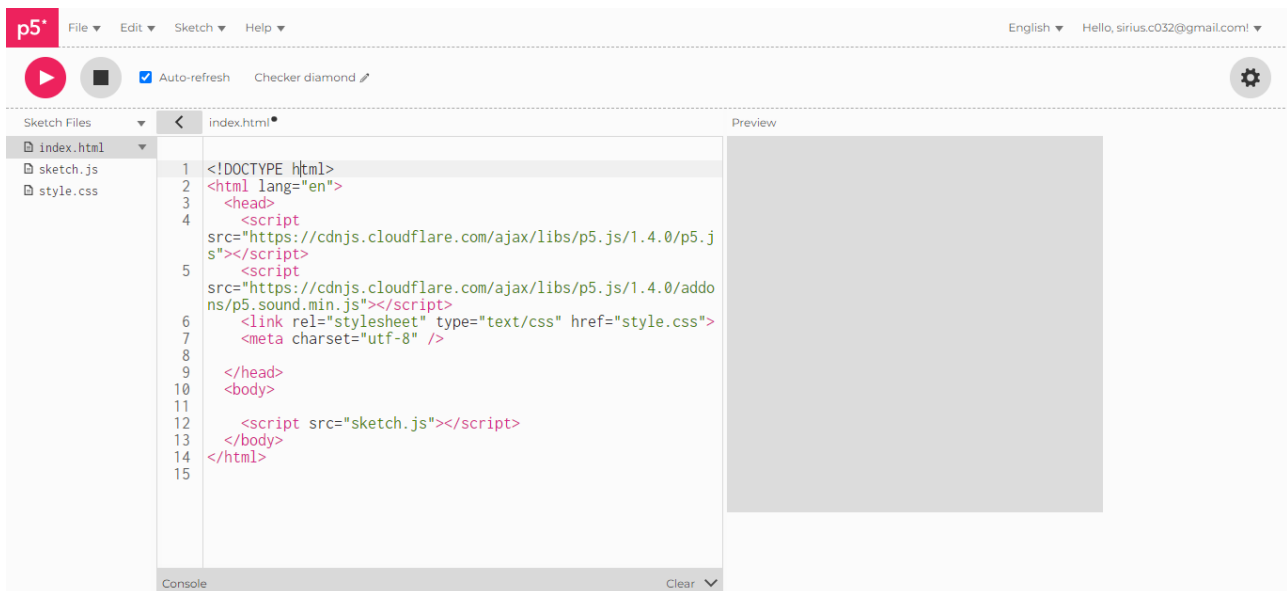


Fig. 3.2. Conținutul fișierului index.html al proiectului.

Se poate observa că în fișierul *index.html* este inclusă biblioteca *p5.js* care conține toate funcțiile grafice care ușurează lucrul cu grafica 2D și 3D în browser, fișierul *p5.sound.min.js* care reprezintă o bibliotecă de sunet care extinde funcționalitățile de baza ale bibliotecii *p5* cu funcționalitate Web Audio, inclusiv intrare audio, redare, analiză și sinteză, fișierul *style.css* în care pot fi adăugate stilurile pentru componentele HTML din cadrul proiectului, (fig. 3.3), fișierul I care reprezintă programul sursă a scenei editat în limbajul JavaScript, (fig. 3.1).

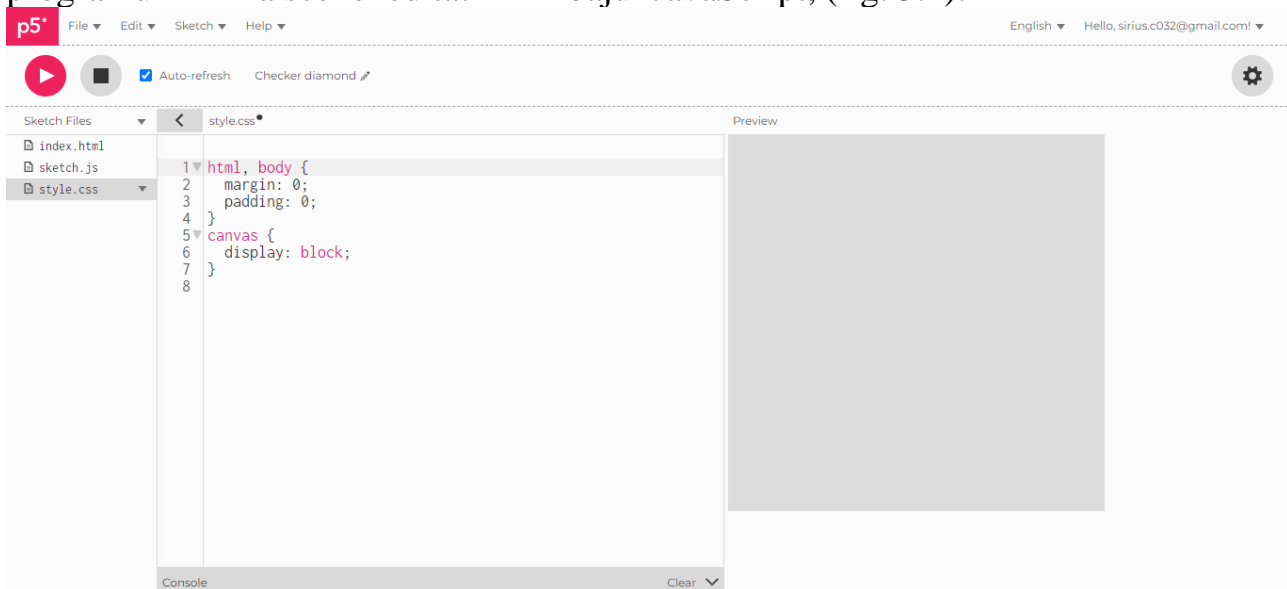


Fig. 3.3. Conținutul fișierului style.css al proiectului

Pentru a integra codul `<svg>` în proiectul realizat cu ajutorul editorului *p5.js* Web Editor este nevoie de a edita fișierul *index.html* și anume înlocuirea rândului `<script src="sketch.js"></script>` din interiorul tag-ului `<body>` cu codul `<svg>` necesar, (fig. 3.4).

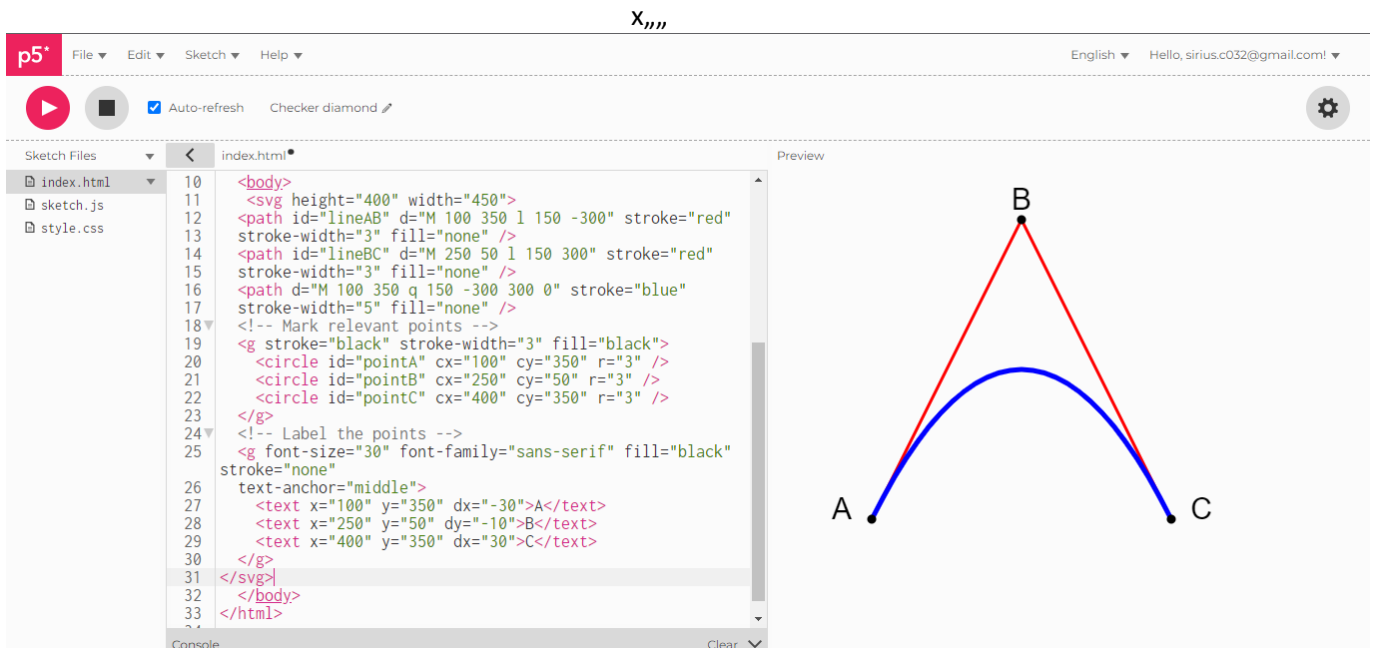


Fig. 3.4. Rezultatul rulării programului

Vizualizarea rezultatului execuției programului realizat poate fi observată în partea dreaptă a editorului.

## Software pentru grafica Vectorială

### Software Grafică Vectorială Plătit

Adobe Illustrator (Mac, Win)

[Adobe Illustrator](#) este software-ul profesional considerat un standard pentru editarea grafică în format vectorial. Vă permite să creați logo-uri, sigle, desene și ilustrații pentru print, web, video și mobil.

Affinity Designer (Mac, Win)

[Affinity Designer](#) - software de editare grafică în format vectorial, rapid, omogen și precis.

Graphic (Mac)

[Graphic](#) software de editare grafică pentru design și ilustrații. Creați ilustrații tehnice detaliate sau desene artistice impresionante. [Autodesk Graphic Ghidul Utilizatorului - Mac](#)

Corel Draw (Win)

[Corel Draw](#) este un software de editare grafică în format vectorial folosit în designul grafic. Capabil de interoperare limitată cu formatele de fișiere grafice din Adobe Illustrator.

Sketch (Win, Mac)

[Sketch](#) este un software de editare grafică cu un set de instrumente de design concepute pentru a vă ajuta să creați grafică în format vectorial mai ales pentru aplicații web și aplicații mobile.

### Software Grafică Vectorială Gratuit

Inkscape (Mac, Win, Linux)

[Inkscape](#) este un software de editare grafică pentru Windows, Mac OS X și Linux. Este gratuit și open source.

Gravit Designer (Mac, Win, Linux, Chrome OS, Browser)

[Gravit Designer](#) este o aplicație online gratuită pentru design și editare vectorială. Editați vectorial, online, direct în browser: [Gravit Designer Online](#)

Vectr (Mac, Win, Linux, Chrome OS, Browser)

[Vectr](#) este un software de editare grafică gratuit, folosit pentru a crea grafică vectorială ușor și intuitiv. Este un instrument simplu și puternic pentru web și platforme desktop.

### **Program de grafică vectorială *Vectr***

Utilizați acest editor vectorial gratuit pe un computer obișnuit sau browser pentru a crea vectori și alte elemente grafice. Este cu adevărat un software multiplatformă și ușor de utilizat: aplicațiile pentru Mac, Windows, Linux sau Chromebook vă sunt disponibile.

Vectr este un instrument la îndemână și este utilizat pe scară largă pentru adnotarea, editarea imaginilor și desenarea de machete și diagrame. În plus, poate fi folosit pentru:

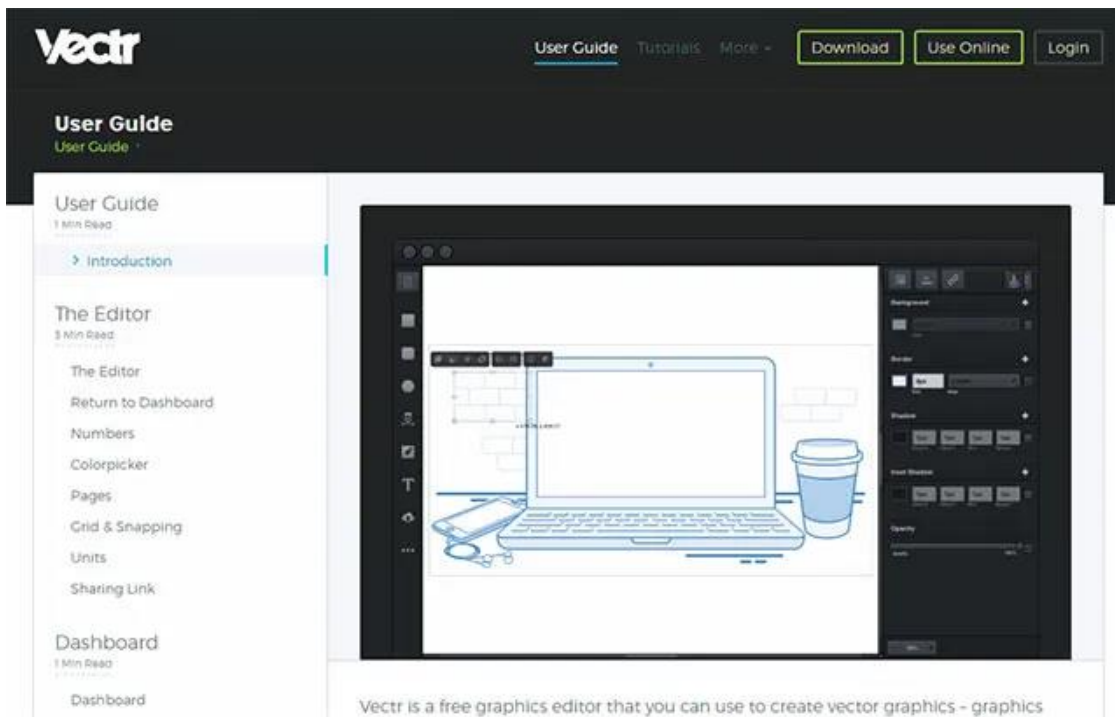
- logos;
- filigranele;
- și planuri de site-uri web;
- bannere de social media;
- icoane;
- și aproape orice alt tip de artă 2D.

Programul vă permite să creați desene vectoriale scalabile utilizând funcții simple și clare. Nu este dificil de învățat, este perfect pentru cei care abia încep să se familiarizeze cu vectorul și care au suficiente capacități de bază de ilustrare.

### **Cum funcționează *Vectr***

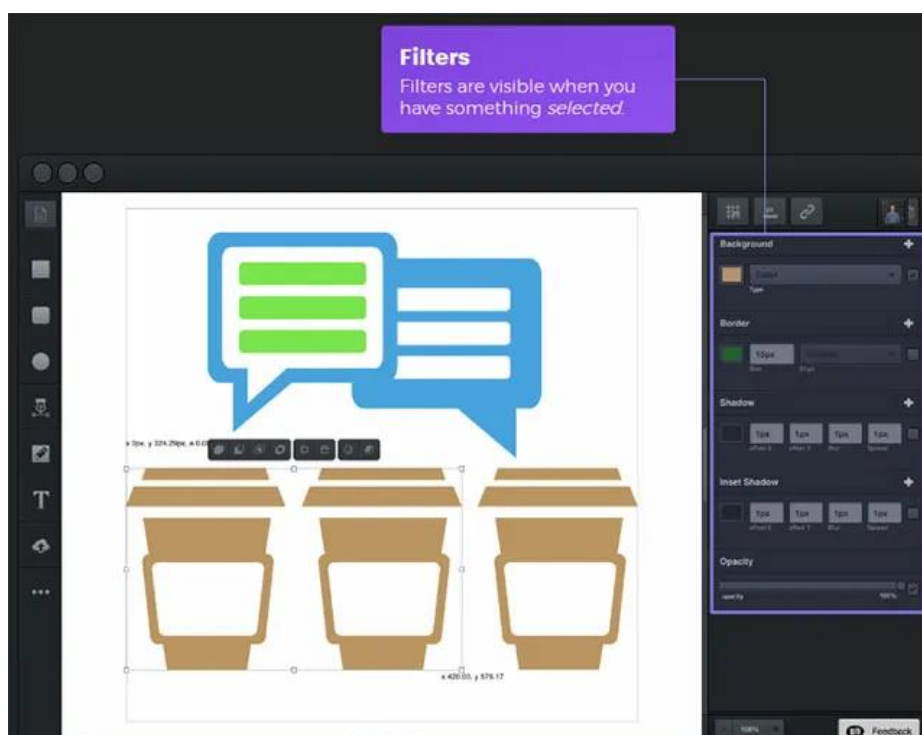
Când începeți să vă adânciți în specificul acestui software, primul lucru pe care îl veți întâlni este un set întreg de tutoriale interactive. Cu Vectr, veți învăța cum să creați și să editați căi, să rotiți și să sculați obiecte, să folosiți instrumente în spațiul de lucru și să gestionați straturile. Există multe ghiduri video pe Youtube.

<https://www.youtube.com/watch?v=CnSRzM91FYY>



Pentru a începe, pur și simplu încărcați sau glisați și fixați o imagine în Vectr. Este permis să importați fișiere în format EPS, AI, SVG, PNG sau JPEG - acest lucru facilitează foarte mult procesul de comutare de la alte alternative.

Interfața este atent gândită și organizată convenabil, astfel încât începătorii nu vor avea dificultăți cu funcții inutile. Filele Pagini și Straturi din colțul din stânga sus vă oferă control asupra procesului de creare a proiectelor. Panoul filtrelor este situat în partea dreaptă a interfeței. Puteți face ilustrațiile și mai interesante, de exemplu, schimbând unghiul, adăugând umbre, margini sau un efect de estompare.



La sfârșitul proiectării, programul vă permite să exportați desenele în PNG, JPEG sau SVG. Ca o notă laterală, SVG este singurul format vector scalabil care poate fi deschis în alte aplicații, în timp ce bitmap PNG și JPEG sunt mai potrivite pentru web.

Este foarte interesant faptul că o mulțime de informații utile și sfaturi practice în secțiunea de instruire sunt dedicate începătorilor. Echipa de dezvoltare a editorului de vector Vectr gratuit oferă instrucțiuni pas cu pas pentru rezolvarea celor mai populare probleme de proiectare: crearea de icoane, sigle, tipografie, meniuri, colaje, infografii și multe altele.

### **Lucrarea de laborator 3**

Tema: Studierea primitivelor grafice vectoriale

*Scopul lucrării:* Obținerea cunoștințelor practice în sinteza scenelor grafice vectoriale 2D, utilizând primitivele grafice simple SVG.

*Sarcina lucrării:*

1. Elaborați un program pentru sinteza unei scene 2D statice utilizând elementele grafice necesare cum ar fi: `<rect>`, `<circle>`, `<ellipse>`, `<line>`, `<polyline>`, `<polygon>`, `<path>`, împreună cu atributele corespunzătoare, scena trebuie să conțină un element `<text>` plasat în colțul dreapta jos a ecranului care ar indica numele, prenumele și grupa studentului.

2. Elaborați un program care creează versiunea vectorială a personajul desenat în lucrarea de laborator nr.1. Variantele sunt indicate în tabelul 1.2. E permis de realizat în editor grafic vectorial.