Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

**Кафедра CАП**



Звіт

з лабораторної роботи №4

На тему: «Робота з графікою»

Дисципліна: «Технології веб-розробки та дизайну»

Варіант 1

Виконала:

ст. гр. ПП-24

Бенькалович М.М.

Прийняла:

Стефанович Т.О.

**Львів** **2024**

**Мета:**

Ознайомитися з графіою

<https://spacenamee.github.io/benkalovych.io/>

1. Ознайомитися з теоретичними матеріалами щодо комп’ютерної графіки, графічних веб-форматів та їх особливостями. Розуміти відмінності між растровою та векторною графікою
2. Здійснити певні дії в наведених редакторах:
   * **Відкрити існуюче зображення**, внести зміни (додати текст чи інше зображення, зменшити розмір, повернути елемент тощо).



Рис.1 Початкове фото з Photopea



Рис.2 Редаговане фото з Photopea

* + **Створити нове полотно** певного розміру, зберегти файл. Розробити тематичне зображення: логотип, банер, аватар, мем, листівку, фотоколаж.

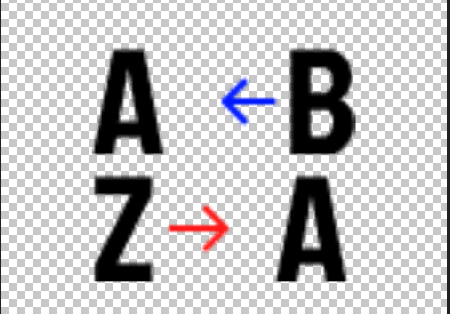
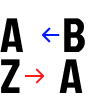
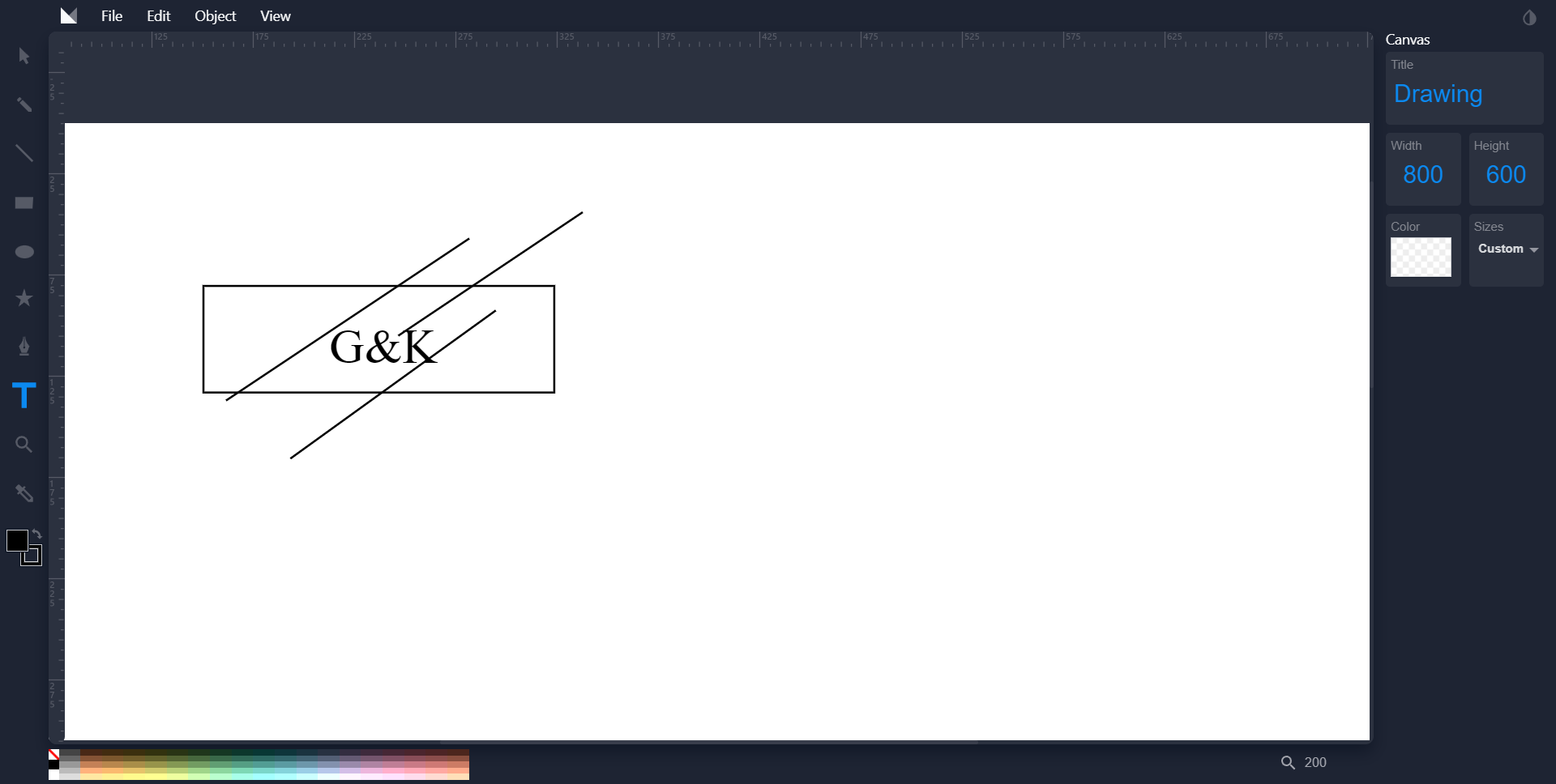
 

Рис.3 Логотип з Figma

* + **Створити SVG зображення** простої фігури з колірними елементами для подальшего розгляду його XML-коду.



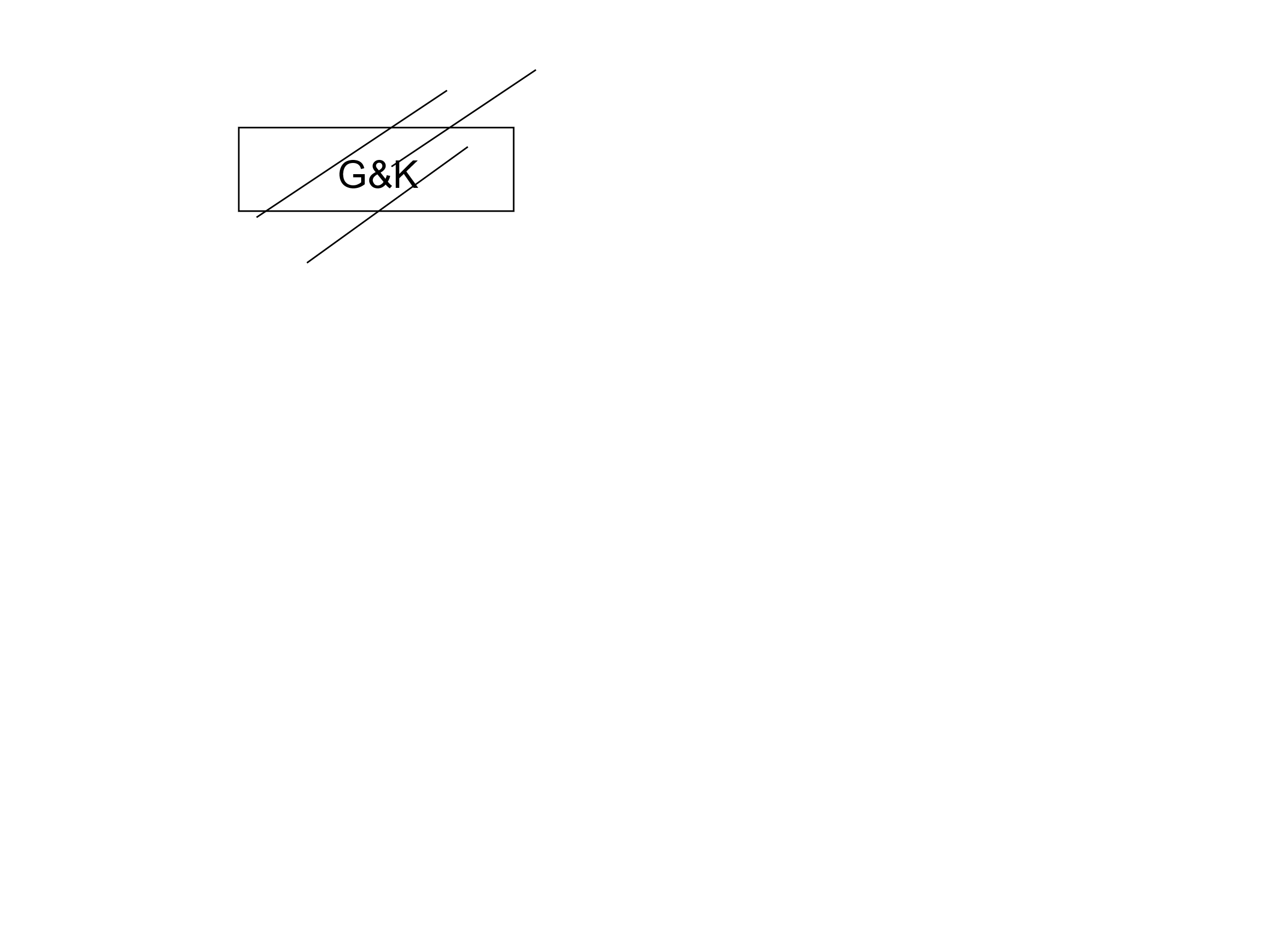


Рис. 4 SVG Зображення

1. Користуючись сервісом [Adobe Color Themes](https://color.adobe.com/ru/" \t "_blank) підібрати для власного редактора вдалу колірну гаму. Обирати можна відштовхуючись як від певного кольору так і від фотографії.

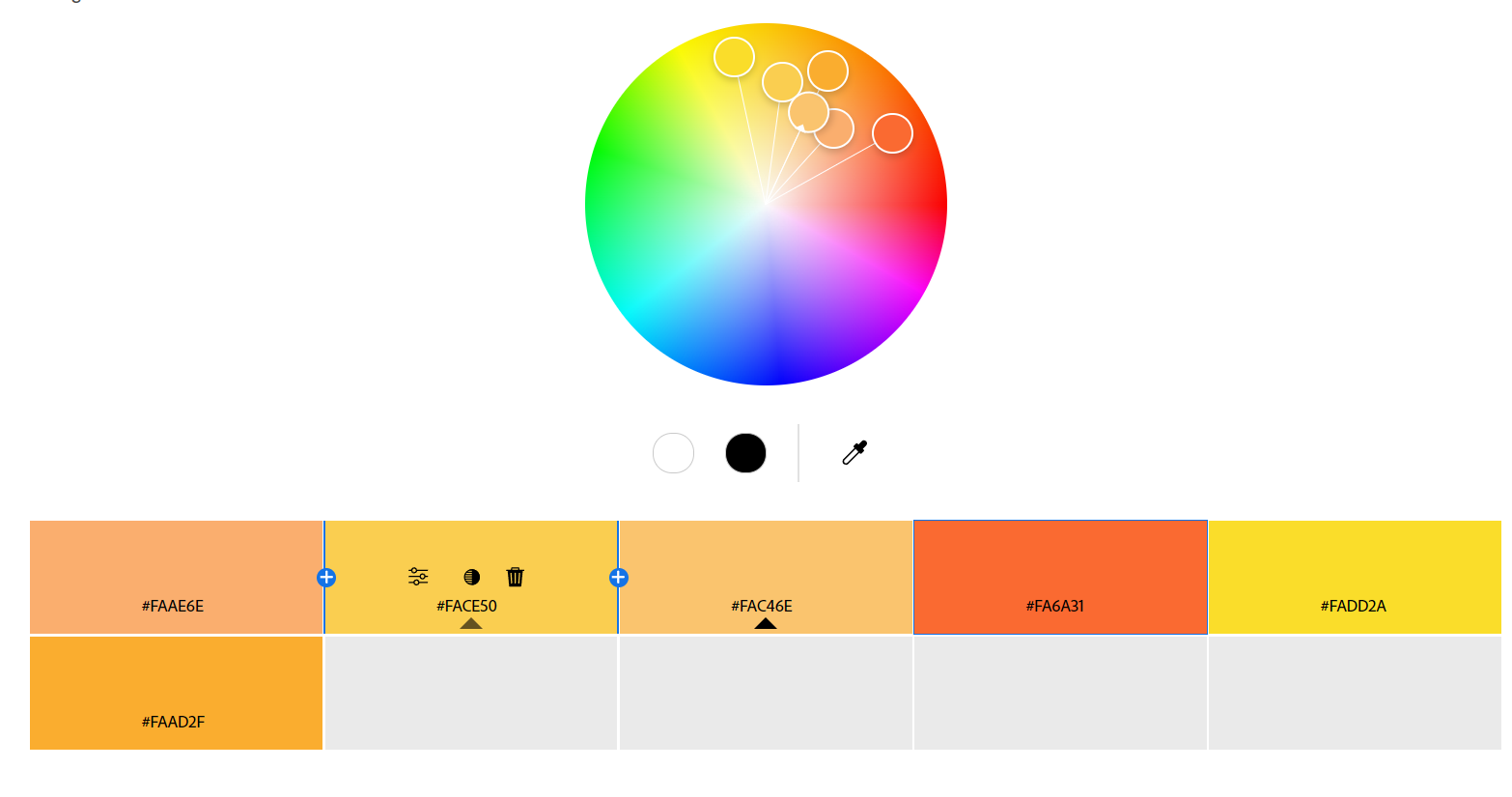


Рис. 5

1. За допомогою HTML-редактора поставити створені графічні елементи у відповідні місця сторінки. Додати стилі до елементів <img>, що стосуються розмірів, розташування, тіней, рамок та іншого оздоблення.
2. img {
3. border: 1px solid black;
4. margin-right: 1em;
5. margin-bottom: 20px;
6. }
7. .img1{
8. width: 300px;
9. height: auto;
10. box-shadow: 2px 2px 2px black;
11. }
12. .img2{
13. width: 300px;
14. object-fit: cover;
15. border-radius: 30%;
16. }
17. .img2:hover{
18. background-color: rgb(132, 99, 240);
19. }
20. .img3{
21. background-color: rgb(243, 22, 158);
22. border: 2px solid black;
23. border-radius: 2%;
24. object-position: 8px;
25. }
26. Користуючись різними підходами створити чи обрати фавіконку, та втілити її на сторінку.

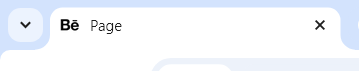


Рис. 6

1. На ресурсах безкоштовних колекцій обрати певне svg-зображення. Уважно розібрати його xml-код, змінити певні значення зображення (розмір, кольори тощо) і вставити на сторінку.

Зміна кольору через stroke



1. Втілити на сторінку зображення (анімації) різних форматів (jpg, png, svg) і способів (css, javascript, svg).

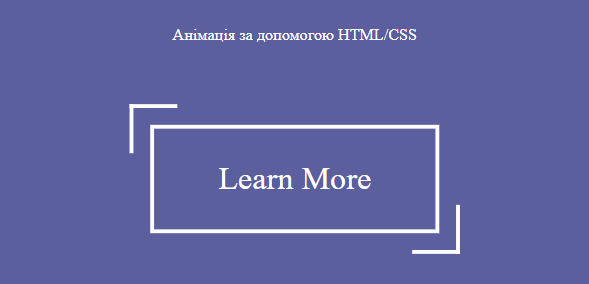


Рис. 7

Html:

<a href="#" class="custom-button">Learn More</a>

Css:

.custom-button:hover{

  text-decoration:none;

  color: #272838;

  background-color:#fff;

/\*   border-radius:4px; \*/

}

.custom-button{

    margin-top: 50px;

  text-decoration: none;

  color: #ffffff;

  border: 3px solid #ffffff;

  padding: 1em 4rem;

  font-size:2rem;

  position:relative;

  transition: all .5s

}

.custom-button::before,

.custom-button::after{

  content:"";

  position:absolute;

  width:3rem;

  height:3rem;

  border-style: solid;

  border-color: #fff;

  transition: all .5s

}

.custom-button::before{

  top: -1.5rem;

  left:-1.5rem;

  border-width: 3px 0 0 3px;

}

.custom-button::after{

  bottom:-1.5rem;

  right:-1.5rem;

  border-width: 0px 3px 3px 0px;

}

.custom-button:hover::before,

.custom-button:hover::after{

  width:calc(100% + 3rem);

  height:calc(100% + 3rem);

  border-color:#fff;

}

html, body, .hello-parent {

    height: 100%;

    width: 100%;

    margin: 0;

  }

  .hello-parent {

    display: flex;

    background: #fff;

    background: -webkit-radial-gradient(#fff, #eaeaea);

    background: -o-radial-gradient(#fff, #eaeaea);

    background: -moz-radial-gradient(#fff, #eaeaea);

    background: radial-gradient(#fff, #eaeaea);

  }



Рис. 8

HTML:  
 <div class="svg\_animation container">

        <svg class="hello-word" width="365" height="365" viewBox="0 0 365 365">

            <g id="H-letter">

            <line class="H-left-stroke" x1="17" y1="0" x2="17" y2="124" stroke="#000" fill="none" stroke-width="34" />

            <line class="H-mid-stroke" x1="33" y1="62" x2="68" y2="62" stroke="#000" fill="none" stroke-width="34" />

            <line class="H-right-stroke" x1="84" y1="0" x2="84" y2="124" stroke="#000" fill="none" stroke-width="34" />

            </g>

            <g id="E-letter">

              <line class="E-left-stroke" x1="138" y1="0" x2="138" y2="124" stroke="#000" fill="none" stroke-width="34" />

              <line class="E-top-stroke" x1="154" y1="17" x2="201" y2="17" stroke="#000" fill="none" stroke-width="34" />

              <line class="E-mid-stroke" x1="154" y1="62" x2="196" y2="62" stroke="#000" fill="none" stroke-width="34" />

              <line class="E-bottom-stroke" x1="154" y1="107" x2="201" y2="107" stroke="#000" fill="none" stroke-width="34" />

            </g>

            <g id="L-one-letter">

              <line class="L-one-long-stroke" x1="17" y1="153" x2="17" y2="277" stroke="#000" fill="none" stroke-width="34" />

              <line class="L-one-short-stroke" x1="33" y1="260" x2="77" y2="260" stroke="#000" fill="none" stroke-width="34" />

            </g>

            <g id="L-two-letter">

              <line class="L-two-long-stroke" x1="104" y1="153" x2="104" y2="277" stroke="#000" fill="none" stroke-width="34" />

              <line class="L-two-short-stroke" x1="120" y1="260" x2="164" y2="260" stroke="#000" fill="none" stroke-width="34" />

            </g>

            <g id="O-letter">

              <circle class="O-stroke" cx="231" cy="215" r="48" stroke="#000" fill="none" stroke-width="31" />

            </g>

            <g id="red-dot">

              <!-- Initially I tried creating a circle but it was harder to manipulate it how I wanted to in CSS so I resorted to using a line trick to make it look like a circle ....

              <circle class="red-dot" cx="325" cy="260" r="20" fill="#FF5851" stroke="none" />

              -->

              <line x1="325" y1="260" x2="325" y2="260" stroke="#FF5851" class="red-dot" />

            </g>

        </svg>

        </div>

CSS:

 /\* SVG animation \*/

  .hello-word {

    margin:auto;

  }

  /\* H Animation \*/

  .H-left-stroke {

    stroke-dasharray: 124px;

    stroke-dashoffset: 124px;

    animation: H-left-move 20s ease forwards;

  }

  .H-mid-stroke {

    stroke-dasharray: 37px;

    stroke-dashoffset: 37px;

    animation: H-mid-move 9s ease forwards;

  }

  .H-right-stroke {

    stroke-dasharray: 124px;

    stroke-dashoffset: 124px;

    animation: H-right-move 13s ease forwards;

  }

  @keyframes H-left-move {

    0% {

      stroke-dashoffset: 124px;

    }

    5% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  @keyframes H-mid-move {

    0% {

      stroke-dashoffset: 37px;

    }

    5% {

      stroke-dashoffset: 37px;

    }

    10% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  @keyframes H-right-move {

    0% {

      stroke-dashoffset: 124px;

    }

    5% {

      stroke-dashoffset: 124px;

    }

    10% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  /\* E Animation \*/

  .E-left-stroke {

    stroke-dasharray: 124px;

    stroke-dashoffset: 124px;

    animation: E-left-move 20s ease forwards;

  }

  .E-top-stroke {

    stroke-dasharray: 47px;

    stroke-dashoffset: 47px;

    animation: E-top-move 10s ease forwards;

  }

  .E-mid-stroke {

    stroke-dasharray: 42px;

    stroke-dashoffset: 42px;

    animation: E-mid-move 10s ease forwards;

  }

  .E-bottom-stroke {

    stroke-dasharray: 47px;

    stroke-dashoffset: 47px;

    animation: E-bottom-move 10s ease forwards;

  }

  @keyframes E-left-move {

    0% {

      stroke-dashoffset: 124px;

    }

    2% {

      stroke-dashoffset: 124px;

    }

    6% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  @keyframes E-top-move {

    0% {

      stroke-dashoffset: 47px;

    }

    6% {

      stroke-dashoffset: 47px;

    }

    11% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  @keyframes E-mid-move {

    0% {

      stroke-dashoffset: 42px;

    }

    8% {

      stroke-dashoffset: 42px;

    }

    13% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  @keyframes E-bottom-move {

    0% {

      stroke-dashoffset: 47px;

    }

    11% {

      stroke-dashoffset: 47px;

    }

    16% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  /\* L One Animation \*/

  .L-one-long-stroke {

    stroke-dasharray: 124px;

    stroke-dashoffset: 124px;

    animation: L-one-long-move 20s ease forwards;

  }

  .L-one-short-stroke {

    stroke-dasharray: 44px;

    stroke-dashoffset: 44px;

    animation: L-one-short-move 10s ease forwards;

  }

  @keyframes L-one-long-move {

    0% {

      stroke-dashoffset: 124px;

    }

    2% {

      stroke-dashoffset: 124px;

    }

    7% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  @keyframes L-one-short-move {

    0% {

      stroke-dashoffset: 44px;

    }

    13% {

      stroke-dashoffset: 44px;

    }

    18% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  /\* L Two Animation \*/

  .L-two-long-stroke {

    stroke-dasharray: 124px;

    stroke-dashoffset: 124px;

    animation: L-two-long-move 20s ease forwards;

  }

  .L-two-short-stroke {

    stroke-dasharray: 44px;

    stroke-dashoffset: 44px;

    animation: L-two-short-move 10s ease forwards;

  }

  @keyframes L-two-long-move {

    0% {

      stroke-dashoffset: 124px;

    }

    3% {

      stroke-dashoffset: 124px;

    }

    8% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  @keyframes L-two-short-move {

    0% {

      stroke-dashoffset: 44px;

    }

    15% {

      stroke-dashoffset: 44px;

    }

    20% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  /\* O Animation \*/

  .O-stroke {

    stroke-dasharray: 302px;

    stroke-dashoffset: 302px;

    animation: O-move 20s ease forwards;

  }

  @keyframes O-move {

    0% {

      stroke-dashoffset: 302px;

    }

    4% {

      stroke-dashoffset: 302px;

    }

    9% {

      stroke-dashoffset: 0px;

    }

    100% {

      stroke-dashoffset: 0px;

    }

  }

  /\* Red Dot Animation \*/

  .red-dot {

    stroke-width: 44px;

    stroke-linecap: round;

    animation: red-dot-grow 8s ease-out forwards;

  }

  @keyframes red-dot-grow {

    0% {

      stroke-width: 0px;

    }

    15% {

      stroke-width: 0px;

    }

    20% {

      stroke-width: 44px;

    }

    100% {

      stroke-width: 44px;

    }

  }

  .svg\_animation{

    margin-top: 50px;

  }

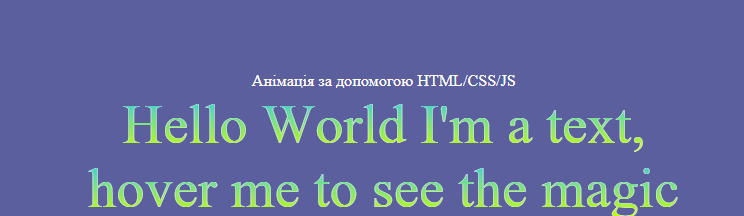


Рис. 9

HTML:

    <div class="split container">

            <h2>Hello World I'm a text,<br/> hover me to see the magic </h2>

        </div>

CSS:

\* {

  -webkit-font-smoothing: antialiased;

}

html,

body {

  width: 100%;

  height: 100%;

  padding: 0;

  margin: 0;

  /\* background: #000; \*/

  color: #fff;

  display: flex;

  align-items: center;

  justify-content: center;

}

.split h2 {

  font-size: 3.5em;

  font-weight: 100;

  text-align: center;

  font-family: "Roboto Flex";

  font-weight: 300;

  font-style: normal;

  font-stretch: 100%;

  line-height: initial;

  font-variation-settings: "opsz" 48, "slnt" 0, "GRAD" 0, "XTRA" 468, "YOPQ" 79,

    "YTAS" 750, "YTDE" -203, "YTFI" 738, "YTLC" 514, "YTUC" 712;

  font-optical-sizing: auto;

}

.split h2 span {

  display: inline-block;

  white-space: break-spaces;

  transition: font-weight 0.2s ease;

  position: relative;

}

.split h2 span::before,

.split h2 span::after {

  content: attr(data-char);

  position: absolute;

  left: 0;

  color: transparent;

}

.split h2 span::after{

  z-index: 2;

  filter: none;

  color: transparent;

   background: linear-gradient(#2cc8f7, #ceff00) 0% 0% / 400% 100%;

  -webkit-background-clip: text;

  -webkit-text-fill-color: transparent;

}

JS:  
        const color = `255 255 255`;

const splitText = () => {

  const split = document.querySelector(".split");

  const textContainer = split.querySelector("h2");

  const text = textContainer.innerHTML;

  textContainer.innerHTML = "";

  const lines = text.split(/<br\s\*\/?>/i);

  lines.forEach((line) => {

    const lineDiv = document.createElement("div");

    split.querySelector("h2").appendChild(lineDiv);

    [...line].forEach((char) => {

      const span = document.createElement("span");

      span.dataset.char=char;

      span.textContent = char === "" ? "&nbsp;" : char;

      lineDiv.appendChild(span);

    });

  });

};

const addHoverEvent = () => {

  const h2 = document.querySelector(".split h2");

  const spans = h2.querySelectorAll("span");

  const max = 1000;

  spans.forEach((span, index) => {

    span.addEventListener("mouseover", function () {

      this.style.fontWeight = max;

      this.style.textShadow = `0 0 0 rgb(${color} / 1)`

      setFontWeight(spans, index, -1, max \* 0.8);

      setFontWeight(spans, index, -2, max \* 0.6);

      setFontWeight(spans, index, 1, max \* 0.8);

      setFontWeight(spans, index, 2, max \* 0.6);

    });

    span.addEventListener("mouseout", function () {

      this.style.backgroundColor = "";

      this.style.fontSize = "";

      this.style.textShadow = ``

      resetFontWeight(spans, index, -1);

      resetFontWeight(spans, index, -2);

      resetFontWeight(spans, index, 1);

      resetFontWeight(spans, index, 2);

    });

  });

  h2.addEventListener("mouseout", function () {

    spans.forEach((span, index) => {

      resetFontWeight(spans, index, -1);

      resetFontWeight(spans, index, -2);

      resetFontWeight(spans, index, 1);

      resetFontWeight(spans, index, 2);

    });

  });

};

const setFontWeight = (spans, index, offset, weight) => {

  if (spans[index + offset]) {

    console.log(index)

    spans[index + offset].style.fontWeight = weight + "";

    spans[index + offset].style.textShadow = `0 0 0 rgb(${color} / ${1 \* (weight/10)/1.2}%)`

  }

};

const resetFontWeight = (spans, index, offset) => {

  if (spans[index + offset]) {

    spans[index + offset].style.fontWeight = "";

    spans[index + offset].style.textShadow = "";

  }

};

const init = () => {

  splitText();

  addHoverEvent();

};

window.addEventListener("DOMContentLoaded", init());

1. Створену сторінку та супутні файли (стилі, зображення) розмістити на хостингу. Перевірити результат виконання через різні браузери. Результат виконання демонструвати в Інтернеті.

Висновок:

В цій лабораторній роботі я дізначала як робити мінімальні анімації за допомогою псевдокласів. Також дізналася в чому різниця між векторними і растовими зображеннями.