 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №4**

з дисципліни «Веб-технології та веб-дизайн-1»

**Виконав:**

студентка гр. БС-81

Кисельова І.Д.

**Перевірив:**

ас. каф. БМК Матвійчук О.В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Завдання:**

1. Створити новий HTML документ, зі списком що буде вести на окрему сторінку для кожного з пунктів завдання // це буде ваш index.html
2. 2d перетворення // виконати в окремому HTML документі
   1. створити елемент зі списку A = [ ‘5-point star’, square, ‘rectangle’, ‘ellipse’, ‘triangle up’, ‘triangle down’, ‘parallelogram’’] відповідно до варіанту за наступною умовою: shape = {variant} mod len(A)
   2. над створеною фігурою виконати 2д перетворення зі списку B = [‘scale’, ‘rotate’, ‘translate’, ‘skew’] відповідно до варіанту за наступною умовою: transform = {group + variant} mod len(B)
   3. задати часову функцію для анімації за списку С = [‘linear’, ‘erase’, ‘erase-in’, ‘erase-out’, ‘erase-in-out’, ‘cubic-bezier’] відповідно до варіанту за наступною умовою: timing-function = {group - variant} mod len(С)
   4. тривалість анімації 3 секунди
3. 3d перетворення
   1. створити прямокутник з розміром 200х200 на виконати над ним перетворення зі списку D = [‘rotateX(45deg)’, ‘rotateY(45deg)’, ‘rotateZ(45deg)’, ‘translate({group}px)’] відповідно до варіанту за наступною умовою: action = {variant} mod len(D)
   2. тривалість анімації 2 секунди
4. Переходи та трансформації

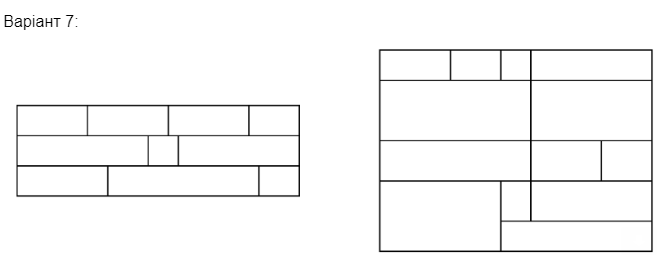
Створити круг діаметром 250px та виконати для нього наступну анімацію переходу:

* 1. тривалість переходу = {(variant mod 10) / 4}
  2. в переході мають участвувати всі поля об’єкта
  3. тип перетворення transform
  4. часова функція перетворення відповідно до варіанту з списку E = [‘linear’, ‘erase’, ‘erase-in’, ‘erase-out’, ‘erase-in-out’, ‘cubic-bezier’] за умовою {variant} mod len(E)

1. Анімація з ключовими кадрами (keyframes)
   1. Зробити keyframe анімацію зміни кольору фона прямокутника розміром 200x200px з проміжними ключовими точками (0, {variant}, 50, {100 - variant}, 100) кольори залишаються на ваш розсуд
   2. тривалість анімації {(group + variant mod 10) / 4}

1. Комбінування анімацій
   1. На сторінці завдання 6 зробити фігуру з завдання 2, виконати для неї одночасно анімації переміщення зміни кольорів та розміру, вихідні розміри фігури, часові функції та координати переміщення на ваш розсуд
   2. тривалість анімації {(group + variant mod 10) / 5}
2. Flexbox та CSS-Grid

На окремій сторінці зробити верстку згідно варіанту. Перший шаблон зверстати засобами css flexbox, другий зверстати за допомогою css grid. Кожен прямокутник зафарбувати у окремий колір. Пропорції прямокутників мають зберігатися при зміні розміру екрана.



Посилання на репозиторій github: <https://github.com/Irines/WEB_commits>

1)index.html

<!DOCTYPE html>

<html>

<head>

<link href ="/sources/indexStyle.css" rel="stylesheet">

<meta charset="utf-8">

<title>Практика4</title>

</head>

<body>

  <p>TASKS</p>

<ul class = "list">

  <li><a href="/sources/2d.html">2D transformation</a></li>

  <li><a href="/sources/3d.html">3D transformation</a></li>

  <li><a href="/sources/transitions.html">Transitions and Transformations</a></li>

  <li><a href="/sources/keyframes.html">Keyframes animations</a></li>

  <li><a href="/sources/comb.html">Combining animations</a></li>

  <li><a href="/sources/flexbox.html">Flexbox + CSS-Grid</a></li>

</ul>

</body>

</html>

Далі я додаю у КП тільки код з .css файлів, оскільки .html файли подібні між собою і основний код у css.

## 2)2D перетворення

.five-point-star {

  display: block;

  margin: 20% 43%;

  width: 0px;

  height: 0px;

  border-right: 120px solid transparent;

  border-bottom: 90px solid green;

  border-left: 120px solid transparent;

  transform: rotate(37deg);

  -webkit-transform: rotate(37deg);  /\*префиксы для разных браузеров \*/

  -moz-transform: rotate(37deg);

  -ms-transform: rotate(37deg);

  -o-transform: rotate(37deg);

  transition:ease-out 3s ;

}

.five-point-star:hover {

  transform: scale(3,3);

  -webkit-transform: scale(3,3);  /\*префиксы для разных браузеров \*/

  -moz-transform:  scale(3,3);

  -ms-transform:  scale(3,3);

  -o-transform:  scale(3,3);

}

/\* маленький треугольник вверху \*/

.five-point-star:before {

  position: absolute;

  display: block;

  width: 0;

  height: 0;

  top: -56px;

  left: -72px;

  content: "";

  border-bottom: 84px solid green;

  border-left: 28px solid transparent;

  border-right: 28px solid transparent;

  -webkit-transform: rotate(323deg);/\*префиксы для разных браузеров \*/

  -moz-transform: rotate(323deg);

  -ms-transform: rotate(323deg);

  -o-transform: rotate(323deg);

  transform: rotate(323deg);

}

.five-point-star:after {

  position: absolute;

  display: block;

  width: 0px;

  height: 0px;

  top: 0;

  left: -120px;

  content: "";

  border-right: 120px solid transparent;

  border-bottom: 90px solid green;

  border-left: 120px solid transparent;

  -webkit-transform: rotate(286deg);/\*префиксы для разных браузеров \*/

  -moz-transform: rotate(286deg);

  -ms-transform: rotate(286deg);

  -o-transform: rotate(286deg);

  transform: rotate(286deg);

}

## 3D перетворення

.rectangle {

    display: block;

    margin: 20% 43%;

    height: 200px;

    width: 200px;

    background-color: rgb(11, 187, 70);

  }

  .rectangle:hover{

    transition:ease 2s ;

    transform: perspective(400px) translate3d(81px, 81px, 81px);

    -webkit-transform: perspective(400px) translate3d(81px, 81px, 81px);  /\*префиксы для разных браузеров \*/

    -moz-transform: perspective(400px) translate3d(81px, 81px, 81px);

    -ms-transform: perspective(400px) translate3d(81px, 81px, 81px);

    -o-transform: perspective(400px) translate3d(81px, 81px, 81px);

    /\* The last translation of element-3 is moving the div 120 pixels in the negative direction of the z-axis.

    Note that you have to apply a value for the perspective also, since without it the translation is not visible. \*/

  }

  body{

    transform-style: preserve-3d;

  }

## 4) Переходи та трансформації

.circle {

    display: block;

    margin: 20% 43%;

    height: 250px;

    width: 250px;

    background-color: rgb(255, 175, 2);

    border-radius: 50%;

  }

  .circle:hover{

    transition-duration: 1.75s;

      transition-timing-function: ease;

      -o-transform: translate(81px, 81px);

      -ms-transform: translate(81x, 81px);

      -moz-transform: translate(81px, 81px);

      -webkit-transform: translate(81px, 81px);

      transform: translate(81px, 81px);

  }

## 5) Анімація з ключовими кадрами (keyframes)

.rectangle {

    display: block;

    margin: 20% 43%;

    height: 200px;

    width: 200px;

    background-color: rgb(11, 187, 70);

    animation-duration: 2s;

    animation-name: changeColor;

    animation-iteration-count: infinite;

  }

  @keyframes changeColor {

    from {

        background-color: rgb(11, 187, 70);

    }

    to {

        background-color: rgb(133, 15, 50);

    }

    7% {

        background-color: rgb(187, 255, 1);

      }

    50% {

        background-color: rgb(255, 153, 0);

    }

    93% {

        background-color: rgb(255, 49, 13);

    }

  }

## 6) Комбінування анімацій

.five-point-star {

    display: block;

    margin: 20% 43%;

    position: relative;

    width: 0px;

    height: 0px;

    border-right: 120px solid transparent;

    border-bottom: 90px solid orange;

    border-left: 120px solid transparent;

    /\* transform: rotate(37deg);

    -webkit-transform: rotate(37deg);

    -moz-transform: rotate(37deg);

    -ms-transform: rotate(37deg);

    -o-transform: rotate(37deg); \*/

    animation-duration: 1.6s;

    animation-name: animation;

    animation-iteration-count: infinite;

    transition-timing-function: linear;

    transform-origin: 50% 50%;

  }

   @keyframes animation {

    from {

        border-bottom: 90px solid orange;

    }

    to {

        border-bottom: 90px solid rgb(133, 15, 50);

        transform: translateX(100px) scale(1.5,1.5);

        -webkit-transform:translateX(100px) scale(1.5,1.5);

       -moz-transform:translateX(100px) scale(1.5,1.5);

       -ms-transform: translateX(100px) scale(1.5,1.5);

       -o-transform:translateX(100px) scale(1.5,1.5);

    }

    50% {

        border-bottom:90px solid rgb(255, 49, 13);

        transform: translateX(150px) scale(1.7,1.7);

        -webkit-transform:translateX(150px) scale(1.7,1.7);

       -moz-transform:translateX(150px) scale(1.7,1.7);

       -ms-transform:translateX(150px) scale(1.7,1.7);

       -o-transform:translateX(150px) scale(1.7,1.7);

    }

    75% {

      border-bottom:90px solid rgb(184, 8, 8);

     transform: translateX(200px) scale(2,2);

     -webkit-transform:translateX(200px) scale(2,2);  /\*префиксы для разных браузеров \*/

     -moz-transform: translateX(200px) scale(2,2);

     -ms-transform: translateX(200px) scale(2,2);

     -o-transform: translateX(200px) scale(2,2);

    }

  }

  @keyframes animation1 {

    from {

          border-bottom: 90px solid orange;

      }

      to {

          border-bottom: 90px solid rgb(133, 15, 50);

      }

      50% {

          border-bottom: 84px solid rgb(255, 49, 13);

      }

      75% {

         border-bottom:90px solid rgb(184, 8, 8);

      }

    }

  /\* маленький треугольник вверху \*/

  .five-point-star1 {

    position: absolute;

    display: block;

    width: 0;

    height: 0;

    top: -56px;

    left: -72px;

    content: "";

    border-bottom: 84px solid orange;

    border-left: 28px solid transparent;

    border-right: 28px solid transparent;

    -webkit-transform: rotate(323deg);/\*префиксы для разных браузеров \*/

    -moz-transform: rotate(323deg);

    -ms-transform: rotate(323deg);

    -o-transform: rotate(323deg);

    transform: rotate(323deg);

    animation-duration: 1.6s;

    animation-name: animation1;

    animation-iteration-count: infinite;

    transition-timing-function: linear;

  }

  .five-point-star2 {

    position: absolute;

    display: block;

    width: 0px;

    height: 0px;

    top: 0;

    left: -120px;

    content: "";

    border-right: 120px solid transparent;

    border-bottom: 90px solid orange;

    border-left: 120px solid transparent;

    -webkit-transform: rotate(286deg);/\*префиксы для разных браузеров \*/

    -moz-transform: rotate(286deg);

    -ms-transform: rotate(286deg);

    -o-transform: rotate(286deg);

    transform: rotate(286deg);

    animation-duration: 1.6s;

    animation-name: animation1;

    animation-iteration-count: infinite;

    transition-timing-function: linear;

  }

## 7) Flexbox та CSS-Grid

## Html

<!DOCTYPE html>

<html>

<head>

<link href="/sources/flexboxStyle.css" rel="stylesheet">

<meta charset="utf-8">

<title>Combining animations</title>

</head>

<body>

    <header></header>

        <div class="main">

            <p><bold>Верстка шаблонов методами css flexbox (1) и css grid(2)</bold></p>

            <div class="container">

                <div class="flexbox">

                    <div class ="row1">

                        <div class = "item">1</div>

                        <div class = "item">2</div>

                        <div class = "item">3</div>

                        <div class = "item">4</div>

                    </div>

                    <div class ="row2">

                        <div class = "item" >5</div>

                        <div class = "item" style="flex-basis:60px">6</div>

                        <div class = "item">7</div>

                    </div>

                    <div class ="row3">

                        <div class = "item">8</div>

                        <div class = "item">9</div>

                        <div class = "item">10</div>

                    </div>

                </div>

                <div class="grid">

                    <div id = "block1"></div>

                    <div id = "block2"></div>

                    <div id = "block3"></div>

                    <div id = "block4"></div>

                    <div id = "block5"></div>

                    <div id = "block6"></div>

                    <div id = "block7"></div>

                    <div id = "block8"></div>

                    <div id = "block9"></div>

                    <div id = "block10"></div>

                    <div id = "block11"></div>

                    <div id = "block12"></div>

                    <div id = "block13"></div>

                    <div id = "block14"></div>

                </div>

            </div>

        </div>

    <footer></footer>

</body>

</html>

## Css

p{

  font-size: x-large;

  padding-left: 30%;

  padding-top: 1%;

}

.container {

    width: 100%;

    max-width: 1600px;

    margin:100px 0;

  }

  .flexbox {

    width: 45%;

    max-width: 700px;

    position: relative;

    padding: 0;

    margin: 20px;

    float: left;

    background-color: peachpuff ;

  }

  .row1 {

    display: inline-flex;

    flex-wrap: nowrap;

    width: 100%;

    justify-content: space-between;

    flex-direction: row;

  }

  .row2 {

    display: inline-flex;

    flex-wrap: nowrap;

    width: 100%;

    justify-content: space-between;

    flex-direction: row;

  }

  .row3 {

    display: inline-flex;

    flex-wrap: nowrap;

    width: 100%;

    justify-content: space-between;

    flex-direction: row;

  }

  .item{

    background-color: #f1f1f1;

    width: 130px;

    text-align: center;

    line-height: 75px;

    font-size: 30px;

    line-height:100px;

  }

  .item:nth-child(1) {background-color:grey; flex-basis: 200px;}

  .item:nth-child(2) { background-color:lightblue; flex-basis: 200px;}

  .item:nth-child(3) {background-color:lightgoldenrodyellow; flex-basis: 200px;}

  .item:nth-child(4) {background-color:lightsteelblue; flex-basis: 100px;}

  .row2 > .item:nth-child(1) {flex-basis: 320px; background-color:linen;}

  .row2 >.item:nth-child(2) {flex-basis: 60px; background-color:lightskyblue;}

  .row2 >.item:nth-child(3) {flex-basis: 320px; background-color:lightslategray;}

  .row3 >.item:nth-child(1) {flex-basis: 250px; background-color:gainsboro;}

  .row3 >.item:nth-child(2) {flex-basis: 370px; background-color:palegoldenrod;}

  .row3 >.item:nth-child(3) {background-color:papayawhip; flex-basis: 80px;}

  .grid {

    width: 45%;

    float: right;

    background-color: powderblue;

    display: grid;

    grid-template-rows: 16% 32% 19.5% 19.5% 13% ;

    grid-template-columns: 26% 19% 12% 27.5% 15.5%;

    height: 500px;

    /\* width: 700px; \*/

  }

#block1{grid-column: 1; grid-row: 1; background: rgb(223, 189, 187);}

#block2{grid-column: 2; grid-row: 1; background: rgb(211, 223, 255);}

#block3{grid-column: 3; grid-row: 1; background: rgb(135, 108, 255);}

#block4{grid-column: 4/6; grid-row: 1; background: rgb(97, 252, 136);}

#block5{grid-column: 1/4; grid-row: 2; background: rgb(253, 255, 118);}

#block6{grid-column: 4/6; grid-row: 2; background: rgb(255, 150, 250);}

#block7{grid-column: 1/4; grid-row: 3; background: rgb(255, 151, 151);}

#block8{grid-column: 4/5; grid-row: 3; background: rgb(47, 197, 97);}

#block9{grid-column: 5; grid-row: 3; background: rgb(141, 133, 255);}

#block10{grid-column: 1/3; grid-row: 4/6; background: rgb(116, 106, 247);}

#block11{grid-column: 3; grid-row: 4/6; background: rgb(75, 170, 118);}

#block12{grid-column: 3; grid-row: 4; background: rgb(160, 96, 96);}

#block13{grid-column: 3/6; grid-row: 5; background: rgb(247, 235, 127);}

#block14{grid-column: 4/6; grid-row: 4; background: rgb(221, 115, 207);}