 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:**

студент гр. БС-81

Тарабара С.В.

**Перевірив:**

ас. каф. БМК Давидько О. Б.

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) = 3
   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) = 2
   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
   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) = 1
   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} = 1,75
  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) = 5

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 (0, 17, 50, 83, 100)
   2. тривалість анімації {(group + variant mod 10) / 4} = 22
2. 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} = 17,6
3. 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.

**Лістинг програми:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Main\_task</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>Welcome to our Gallery of Tasks</header>

<div class="main">

<h1>Tasks</h1>

<ul>

<li> <a href="task1.html">1) 2d Перетворення</a>

</li>

<li>

<a href="task2.html">2) 3d Перетворення</a>

</li>

<li>

<a href="task3.html">3) transformation + transition</a>

</li>

<li>

<a href="task4.html">4) keyframes</a>

</li>

<li>

<a href="task5.html">4) all in</a>

</li>

</ul>

</div>

<footer>created by Serj tarik, 2020</footer>

</body>

</html>

**Flexbox and Grid**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>lastWork</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1 style="text-align: center;">FlexBox</h1>

<div class="flex">

<div class="firstrow">

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

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

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

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

</div>

<div class="firstrow">

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

<div class="div-6"></div>

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

</div>

<div class="firstrow">

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

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

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

<div class="div-11"></div>

<div class="div-12"></div>

</div>

</div>

<h1 style="text-align: center;">Grid</h1>

<div class="grid">

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

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

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

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

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

<div class="div-6"></div>

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

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

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

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

</div>

</body>

</html>

**Style.css**

body {

background: pink;

height: 100vh;

font-size: 25px;

}

\* {

margin: 0;

padding: 0;

}

h1 {

margin: 100px 0;

}

.main {

display: flex;

/\* justify-content: center; \*/

margin-top: 50px;

align-items: center;

height: 100%;

flex-direction: column;

font-size: 25px;

}

ul li {

font-style: italic;

margin: 10px;

}

header,

footer {

background: gray;

text-align: center;

padding: 20px;

}

ul li {

list-style: none;

}

a {

text-decoration: none;

}

.rectangle {

width: 200px;

height: 100px;

background: red;

}

.rotatestatic {

margin: 50px 0;

transform: rotate(20deg);

}

.rotateAnimation {

margin: 50px 0;

animation: rotAn 3s ease-out;

}

.pryamokytnik {

width: 200px;

height: 200px;

background: yellow;

transition: all 2s;

}

.pryamokytnik:hover {

transform: rotate3d(1, 1, 1, 45deg)

}

.pryamokytnik4 {

width: 200px;

height: 200px;

background: rgb(255, 0, 212);

animation: task4 22s;

}

.pryamokytnik5 {

width: 200px;

height: 200px;

background: rgb(38, 0, 255);

animation: task5 5s ease-in-out;

}

.circle {

width: 250px;

height: 250px;

background: green;

border-radius: 50%;

transition: all 1.75s ease-in-out;

}

.circle:hover {

background: #FF686B;

transform: rotate3d(1, 1, 1, 90deg);

}

@keyframes rotAn {

0% {

transform: none;

}

95% {

transform: rotate(20deg);

}

}

@keyframes task4 {

0% {

background: rgb(38, 0, 255);

}

25% {

background: rgb(87, 8, 38);

}

50% {

background: rgb(75, 11, 128);

}

75% {

background: rgb(19, 10, 97);

}

100% {

background: rgb(13, 141, 158);

}

}

@keyframes task5 {

0% {

background: rgb(255, 0, 212);

transform: none;

width: 200px;

height: 200px;

}

25% {

background: rgb(87, 8, 38);

width: 150px;

height: 150px;

}

50% {

background: rgb(75, 11, 128);

width: 125px;

height: 125px;

}

75% {

background: rgb(19, 10, 97);

width: 100px;

height: 100px;

}

100% {

background: rgb(13, 141, 158);

transform: translateX(1000px);

width: 70px;

height: 70px;

}

}

/\* flex + grid \*/

/\* flex \*/

.flex {

display: flex;

flex-direction: column;

}

.firstrow {

display: flex;

flex-direction: row;

flex-wrap: wrap;

width: 100%;

}

.div-1 {

background: red;

padding: 40px;

flex: 10;

}

.div-2 {

background: rgb(93, 8, 163);

padding: 40px;

flex: 7;

}

.div-3 {

background: blue;

padding: 40px;

flex: 7;

}

.div-4 {

background: black;

padding: 40px;

flex: 20;

}

.div-5 {

background: rgb(10, 182, 5);

padding: 40px;

flex: 10;

}

.div-6 {

background: saddlebrown;

padding: 40px;

flex: 20;

}

.div-7 {

background: khaki;

padding: 40px;

flex: 10;

}

.div-8 {

background: olive;

padding: 40px;

flex: 10;

}

.div-9 {

background: rgb(31, 185, 185);

padding: 40px;

flex: 3.5;

}

.div-10 {

background: rgb(141, 20, 56);

padding: 40px;

flex: 3.5;

}

.div-11 {

background: rgb(221, 166, 14);

padding: 40px;

flex: 3.5;

}

.div-12 {

background: rgb(7, 6, 0);

padding: 40px;

flex: 23;

}

.grid {

display: grid;

grid-template-columns: repeat(9, 1fr);

grid-auto-rows: minmax(100px, auto);

grid-template-areas:

"div-1 div-1 div-1 div-1 div-1 div-2 div-2 div-2 div-2"

"div-3 div-3 div-3 div-3 div-3 div-3 div-3 div-3 div-3"

"div-4 div-4 div-4 div-5 div-5 div-5 div-5 div-5 div-5"

"div-4 div-4 div-4 div-6 div-6 div-6 div-6 div-6 div-6"

"div-7 div-7 div-8 div-8 div-9 div-9 div-9 div-9 div-9""div-7 div-7 div-8 div-8 div-10 div-10 div-10 div-10 div-10";

}

.grid .div-1 {

grid-area: div-1;

}

.grid .div-2 {

grid-area: div-2;

}

.grid .div-3 {

grid-area: div-3;

}

.grid .div-4 {

grid-area: div-4;

}

.grid .div-5 {

grid-area: div-5;

}

.grid .div-6 {

grid-area: div-6;

}

.grid .div-7 {

grid-area: div-7;

}

.grid .div-8 {

grid-area: div-8;

}

.grid .div-9 {

grid-area: div-9;

}

.grid .div-10 {

grid-area: div-10;

}