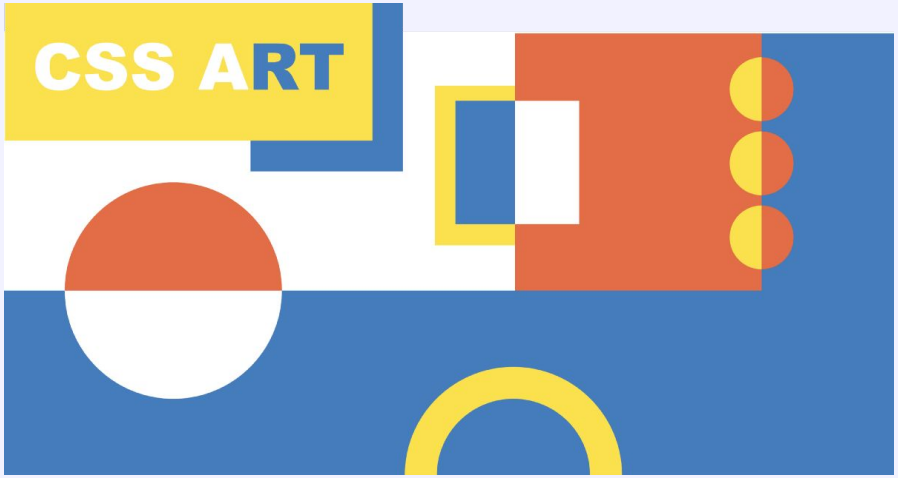
IT-студ//

CSS графіка та анімація

CSS — керує дизайном



HTML і CSS є двома будівельними блоками веброзробки:

- HTML структурує сторінку,
- CSS забезпечує стиль.

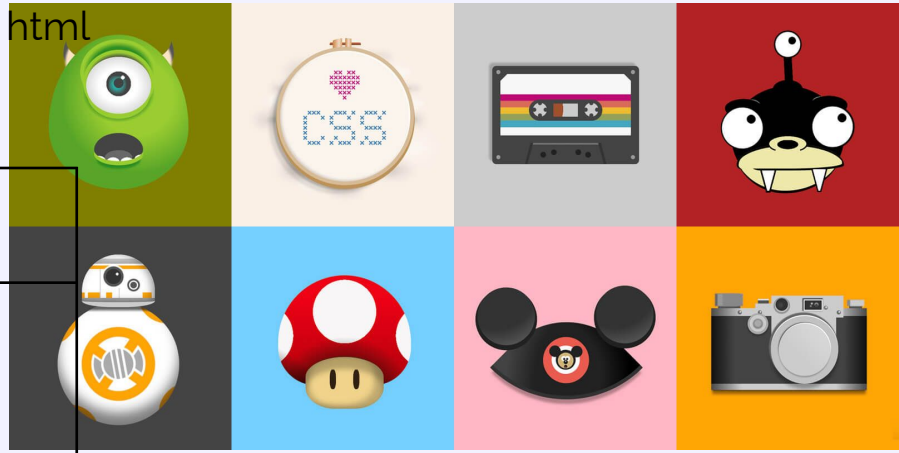
Використовуючи усього 2 цих інструменти можна не тільки робити красиві вебсторінки, і навіть …

створювати цифрові арт- проєкти!

Графіка CSS

CSS здатний створювати будь-які форми. Для цього треба:

- оголосити з класом у розділі html
- стилізувати його за допомогою CSS.



HTML	CSS

<html>
<body>
 	
</body></html>
	.box {
	
	властивості класу box;
	
	}

Графіка в тексті

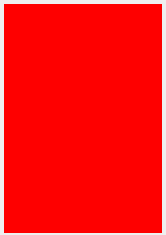
HTML

Створити

У **CSS** прописуємо властивості:

width — ширина,
height — висота,
background — колір.

Прямокутник



```
.square {
  width: 100px;
  height: 200px;
  background: red;
}
```

Графіка в тексті

HTML

Створити

У **CSS** прописуємо властивості:

width — ширина,
height — висота,
background — колір.

Еліпс або коло

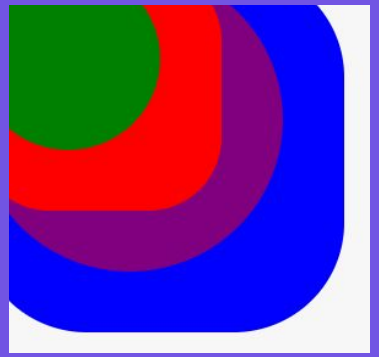
```
.oval {
    width: 200px;
    height: 100px;
    background: red;
    border-radius: 100px / 50px;
  }

.circle {
    width: 200px;
    height: 200px;
    background: red;
    border-radius: 50%;
  }
```

Завдання

JS Bin

Створіть зображення за зразком.



- Властивості фігур rect2 та circle2 пропишіть у CSS.
- Зверніть увагу, що кожний елемент div містить наступний елемент div.

HTML

```
<!DOCTYPE html>
<html>
<body>
<div class="rect1">

  <div class="circle1">
  <div class="rect2">
  <div class="circle2">
</div>  </div>  </div>  </div>
  </body>
</html>
```

Css

```
.rect1 {
    width: 300px;
    height: 300px;
    background: blue;
    border-radius: 30%;
 }
.circle1 {
    background: purple;
    width: 250px;
    height: 250px;
    border-radius: 50%;
 }
```

IT-студ//

Абсолютне та відносне позиціонування фігури

Щоб визначити положення фігури, використовуємо властивість **position.**

Абсолютне позиціонування — зміщення від країв браузера

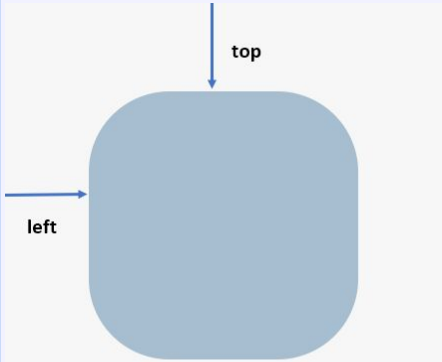
position: absolute

зверху — top,

праворуч — right,

знизу — bottom,

ліворуч — left.



Відносне позиціонування — поточне положення без зміни навколишнього макета.

position: relative

Елементи можна накладати

z-index: 2; — накладання елемента зверху на попередні два;

z-index: –2; — розташування елемента під два попередніх об'єкти.

HTML	CSS

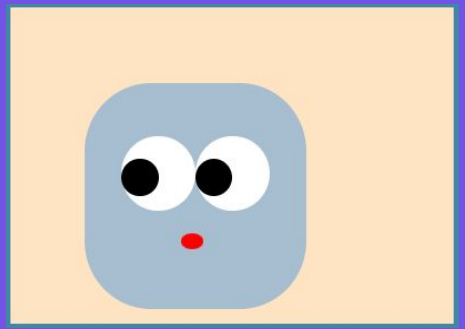
```html
<html>
<body>
<!-- Контейнер для малювання -->
<div class="box">
<!-- Голова -->
<div class="head">
<!-- Ліве око біле -->
<div class="eye-left"> </div>
<!-- Зіниця  чорна -->
<div class="pupil1"> </div>
<!-- Праве око біле -->
<div class="eye-right"></div>
<!-- Зіниця чорна -->
<div class="pupil2"> </div>
<!-- Рот -->
<div class="mouth">
<!-- Закінчення голови --></div>
<!-- Закритий контейнер для малювання -->
</div>

</body></html>
```

```css
body {  background: #4386a8; }
.box {     position: relative;
    width: 600px;
    height: 420px;
    background:#FFE4C4;}
.head {    position: absolute;
    top: 100px;
    left: 100px;
    width: 300px;
    height: 300px;
    background: #A6BECF;
    border-radius: 30%;}
.eye-left {
    position: absolute;
    background: white;
    width: 100px;
    height: 100px;
    top: 70px;
    left: 50px;
    border-radius: 50%;
    z-index: 2;}
.pupil1 {
    position: absolute;
    background: black;
    width: 50px;
    height: 50px;
    top: 100px;
    left: 50px;
    border-radius: 50%;
    z-index: 2;}
```

Завдання

Додайте у код кола для правого ока та рота (div для цих елементів виділені у прикладі коду у розділі HTML).
Зверніть увагу на позиціювання елементів малюнку. Box, який є контейнером для малювання, має position: relative, інші елементи — всередині нього — position: absolute.



IT-студ//

Анімація CSS

CSS-анімація дає змогу анімувати переходи від однієї конфігурації стилю CSS до іншої.

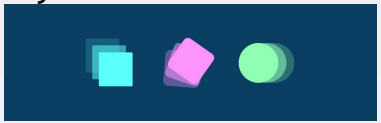
div — будівельний блок анімації.

Визначення ключових кадрів **@keyframes**

Властивості **animation:**

- **animation-name** — визначає ім'я **@keyframes**, який створює кадри анімації;
- **animation-duration** — визначає час одного циклу анімації;
- **animation-delay —** затримка між часом

```
div {
  animation-name: example;
  animation-duration: 4s;
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
}
```



```
@keyframes example {
  0% {background-color:red;
left:0px; top:0px;}
  50% {background-color:blue;
left:200px; top:200px;}
  100% {background-color:red;
left:0px; top:0px;}
}
```

Анімація за вказаною траєкторією

HTML

```
<!DOCTYPE html>
<html>
<body>
<h1>my CSS </h1>

<div></div>

</body>
</html>
```

CSS

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name:
example;
  animation-duration:
4s;
}

@keyframes example {
  0% {left:0px;}

  50% {left:200px;}

  100% {left:0px;}
}
```

Завдання

1. Змініть основний колір div (background-color).
2. Додайте колір для анімованого елемента background {left:0px; background:blue;}.
3. Змініть відсотки keyframes, щоб змінити ключові кадри (наприклад, 65%)

Анімація за складною траєкторією

HTML

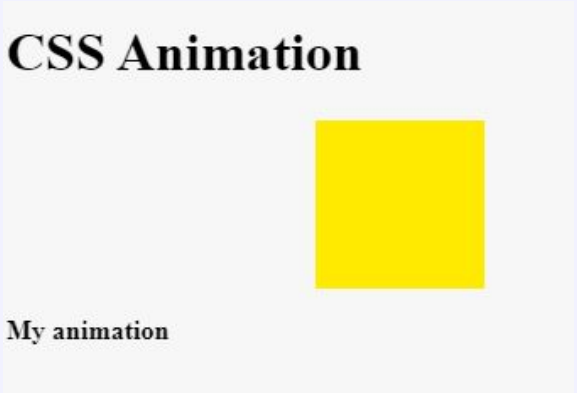
```
<!DOCTYPE html>
<html>
<body>
     <h1>CSS Animation</h1>

     <div></div>

     <p><b>My
animation</b></p>
</body>
</html>
```

CSS

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
}
@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
```



Завдання

1. Змініть основний колір div (background-color).
2. Тривалість анімації (animation-duration: 4s).
3. Змініть відсотки keyframes, щоб змінити ключові кадри.
4. Змініть кольори div у ключових кадрів.
5. Додайте затримку animation-delay: 2s;.

Обертання малюнку

HTML

```
<body>
<div><img
     class="spinner"
     alt="Loading..."
  src="http://surl.li/cwyfj"/>
</div>
</body>
```

CSS

```
 .spinner {
    animation: fancy-spin 2000ms;
    animation-iteration-count: infinite;
    width:70px;
    height:50px
 }

@keyframes fancy-spin {
    10% {transform: rotate(0turn) scale(1);}
    25% {transform: rotate(1turn) scale(1);}
    50% {transform: rotate(1turn) scale(1.5);}
    75% {transform: rotate(0turn) scale(1.5);}
    100% {transform: rotate(0turn) scale(1);}
 }
```

Завдання

1. Додайте відступи малюнку зверху — 100px та зліва — 100px. Для цього вкажіть до властивостей класу spinner — position: absolute; top: 100px; left:100px;
2. Змініть розмір зображення, додаючи до класу spinner width:100px; height:50px;
3. Змініть значення scale у ключовому кадрі 75%.
   75% { transform: rotate(0turn) scale(3);}