

**ОСНОВИ ПРОГРАМУВАННЯ
ДЛЯ ШКОЛЯРІВ**

**ГРУПА
PRO**

**СТВОРЕННЯ САЙТУ
HTML ТА CSS**

**СТВОРЕННЯ ФІГУР
CSS**

Подекуди, нам потрібно створювати фігури у CSS.

Наприклад, це може бути незвичної форми кнопки, або якісь рамки, або навіть задній фон блоків з плавними формами.

Для цього у нас використовуються вже знайомі нам властивості **border**.

Властивість **border** зображує обведення навколо елемента.

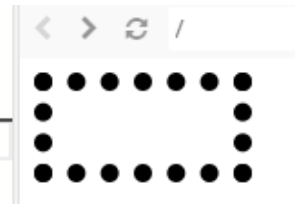
Нагадаю, що **border** – це шорткор до кількох значень, де ми можемо записати стиль обведення, його товщину та колір.

В началі курсу, ми з вами вивчали такі значення **border-style**:

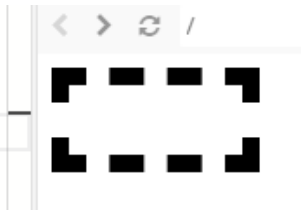
```
div {
  width: 100px;
  height: 40px;
  border: 10px solid black;
}
```



```
div {
  width: 100px;
  height: 40px;
  border: 10px dotted black;
}
```



```
div {
  width: 100px;
  height: 40px;
  border: 10px dashed black;
}
```



```
div {
  width: 100px;
  height: 40px;
  border: 10px double black;
}
```



```
div {
  width: 100px;
  height: 40px;
  border: 10px groove black;
}
```



ridge - інверсне значення **groove**

```
div {
  width: 100px;
  height: 40px;
  border: 10px inset black;
}
```



outset - інверсне значення **inset**

border-color – встановлює колір обведення.

Якщо значення одне – це один колір, але ми можемо встановлювати колір окремо для всіх сторін!

За правилами «годинникової стрілки» ми встановлюємо колір послідовно:

Одне значення

Колір кордону буде встановлений всіма сторонами елемента.

```
div {
  width: 100px;
  height: 40px;
  border: 10px solid;
  border-color: red;
}
```



Два значення

Перше значення встановлює колір верхньої та нижньої межі, друге – лівої та правої.

```
div {
  width: 100px;
  height: 40px;
  border: 10px solid;
  border-color: red black;
}
```



Три значення

Перше значення задає колір верхньої межі, друге — одночасно лівої та правої межі, а третє — нижньої межі.

```
div {
  width: 100px;
  height: 40px;
  border: 10px solid;
  border-color: red black blue;
}
```



Чотири значення

Почергово встановлюється колір верхньої, правої, нижньої та лівої межі.

```
div {
  width: 100px;
  height: 40px;
  border: 10px solid;
  border-color: red black blue green;
}
```



border-width – властивість, яка встановлює ширину обведення.

Так само працює за принципом «годинникової стрілки»

Для кожної сторони існують окремі значення

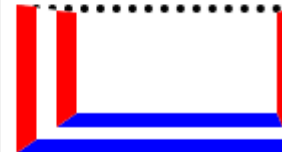
```
.box {
  width: 100px;
  height: 50px;
  border-color: red;
  border-style: double;
  border-width: 4px 10px 20px 30px;
}
```



border-top-width
border-top-style
border-top-color

border-left-width
border-left-style
border-left-color

```
.box {
  width: 100px;
  height: 50px;
  border-color: red;
  border-style: double;
  border-width: 4px 10px 20px 30px;
  border-bottom-color: blue;
  border-top-style: dotted;
  border-top-color: black;
}
```



border-right-width
border-right-style
border-right-color

border-bottom-width
border-bottom-style
border-bottom-color

Властивість (шорткод) , яка дозволяє одночасно встановити товщину, стиль та колір обведенню елемента з різних боків .

border-top
border-right
border-left
border-bottom

Завдяки тому, що кожен бордер може бути прописаний окремо, ми можемо створювати різні фігури

```
.triangle2 {
width: 0;
height: 0;
border-top: 100px solid #68B8AE;
border-right: 100px solid yellow;
}
```



```
.triangle3{
width: 0;
height: 0;
border-left: 50px solid red;
border-right: 50px solid yellow;
border-bottom: 100px solid #24375B;
}
```



```
.triangle2 {
width: 0;
height: 0;
border-top: 100px solid #68B8AE;
border-right: 100px solid
transparent;
}
```



```
.triangle3{
width: 0;
height: 0;
border-left: 50px solid transparent;
border-right: 50px solid transparent;
border-bottom: 100px solid #24375B;
}
```



```
.arrow {
width: 60px;
height: 20px;
margin: 10px 0;
background: #B74388;
position: relative;
}
.arrow:after {
content: "";
width: 0;
height: 0;
position: absolute;
top: -10px;
left: 100%;
border-width: 20px 0 20px 40px;
border-style: solid;
border-color: transparent #B74388;
}
```



```
.heart {
position: relative;
width: 100px;
height: 80px;
}
.heart:before,
.heart:after {
content: "";
position: absolute;
left: 50px;
top: 0;
width: 50px;
height: 80px;
background: #EE4156;
border-radius: 50px 50px 0 0;
transform: rotate(-45deg);
transform-origin: 0 100%;
}
.heart:after {
left: 0;
transform: rotate(45deg);
transform-origin: 100% 100%;
}
```

