

## Теория к практическому занятию

### Количество и ширина рядов в CSS гридах

Элементы грида можно также разместить по рядам. Для этого предназначено свойство *grid-template-rows*. Это свойство через пробел принимает размеры рядов. В нем можно применять те же значения, которые мы изучали и для столбцов.

```
1 <div id="parent">
2   <div>1</div>
3   <div>2</div>
4   <div>3</div>
5   <div>4</div>
6 </div>

1 #parent {
2   display: grid;
3   grid-template-rows: 50px 100px 50px 50px;
4   border: 2px solid #696989;
5   padding: 10px;
6   width: 600px;
7 }
8 #parent > div {
9   padding: 10px;
10  border: 1px solid #696989;
11 }
```

Результат выполнения кода:

1
2
3
4

### Сокращение для рядов и столбцов в CSS гридах

Бывают ситуации, когда удобно указать размеры рядов и столбцов в краткой форме. Для этого мы используем свойство *grid-template*, которое указывается в элементе-родителе и представляет собой сокращенную форму записи двух свойств *grid-template-rows* и *grid-template-columns*.

Ряды и столбцы для грид-контейнера указываются через слеш, их размеры определяются в единицах измерения.

```

1 <div id="parent">
2   <div>1</div>
3   <div>2</div>
4   <div>3</div>
5   <div>4</div>
6   <div>5</div>
7   <div>6</div>
8   <div>7</div>
9   <div>8</div>
10  <div>9</div>
11 </div>

1 #parent {
2   display: grid;
3   grid-template: 60px 1fr 60px / 20% 1fr 15%;
4   border: 2px solid #696989;
5   padding: 10px;
6   width: 600px;
7   height: 300px;
8 }
9 #parent > div {
10  padding: 10px;
11  border: 1px solid #696989;
12 }

```

Результат выполнения кода:

1	2	3
4	5	6
7	8	9

## Объединение рядов в CSS гридах

Для того, чтобы элемент занял более одного ряда, ему следует задать свойство **grid-row**. В этом свойстве через слеш задаются числа, указывающие начальную и конечную позицию элементов в гриде.

При этом элемент растянется от первой позиции до второй позиции (не включая ее). То есть значение 1 / 2 заставит элемент занять одну первую ячейку, а значение 1 / 3 заставит элемент занять первую и вторую ячейку (но не включая третью).

```

1 <div id="parent">
2   <div id="elem1">1</div>
3   <div id="elem2">2</div>
4   <div id="elem3">3</div>
5   <div id="elem4">4</div>
6 </div>

1 #parent {
2   display: grid;
3   grid-template-columns: 2fr 1fr;
4   height: 300px;
5   width: 400px;
6   padding: 10px;
7   border: 2px solid #696989;
8 }
9 #parent > div {
10  padding: 10px;
11  border: 1px solid #696989;
12 }
13 #elem1 {
14   grid-row: 1 / 2;
15 }
16 #elem2 {
17   grid-row: 2 / 3;
18 }
19 #elem3 {
20   grid-row: 3 / 4;
21 }
22 #elem4 {
23   grid-row: 1 / 4;
24 }

```

Результат выполнения кода:

1	4
2	
3	

## Комбинации объединений по рядам и столбцам в CSS гридах

Давайте сделаем таблицу, скомбинировав свойства `grid-column` и `grid-row`:

```

1 <div id="parent">
2   <div id="elem1">1</div>
3   <div id="elem2">2</div>
4   <div id="elem3">3</div>
5   <div id="elem4">4</div>
6 </div>

1 #parent {
2   display: grid;
3   grid-template-columns: 2fr 1fr 1fr;
4   border: 2px solid #696989;
5   padding: 10px;
6   height: 300px;
7   width: 400px;
8 }
9 #parent > div {
10  padding: 10px;
11  border: 1px solid #696989;
12 }
13 #elem1 {
14   grid-row: 1 / 3;
15 }
16 #elem2 {
17   grid-row: 1 / 2;
18 }
19 #elem3 {
20   grid-row: 1 / 2;
21 }
22 #elem4 {
23   grid-row: 2 / 3;
24   grid-column: 2 / 4;
25 }

```

Результат выполнения кода:

1	2	3
	4	

## Расстояние между столбцами и рядами в CSS гридах

С помощью свойства **gap** можно одновременно задать расстояние между столбцами и рядами грида. Можно передать либо одно значение, либо два через пробел. Если передано одно значение, оно задает одновременно расстояние между столбцами и рядами. Если же передано два значения, то первое задает расстояние между рядами, а второе - между столбцами.

```
1 <div id="parent">
2   <div>1</div>
3   <div>2</div>
4   <div>3</div>
5   <div>4</div>
6   <div>5</div>
7   <div>6</div>
8   <div>7</div>
9   <div>8</div>
10  <div>9</div>
11 </div>

1 #parent {
2   display: grid;
3   gap: 10px;
4   grid-template: 1fr 1fr 1fr / 1fr 1fr 1fr;
5   padding: 10px;
6   border: 2px solid #696989;
7   width: 600px;
8   height: 200px;
9 }
10 #parent > div {
11   padding: 10px;
12   border: 1px solid #696989;
13 }
```

Результат выполнения кода:

1	2	3
4	5	6
7	8	9