

Направление флексбоксов

Свойство [flex-direction](#) позволяет установить направление, в котором флекс-элементы будут идти внутри своего флекс-контейнера. Оно принимает следующие значения:

- row
- row-reverse
- column
- column-reverse

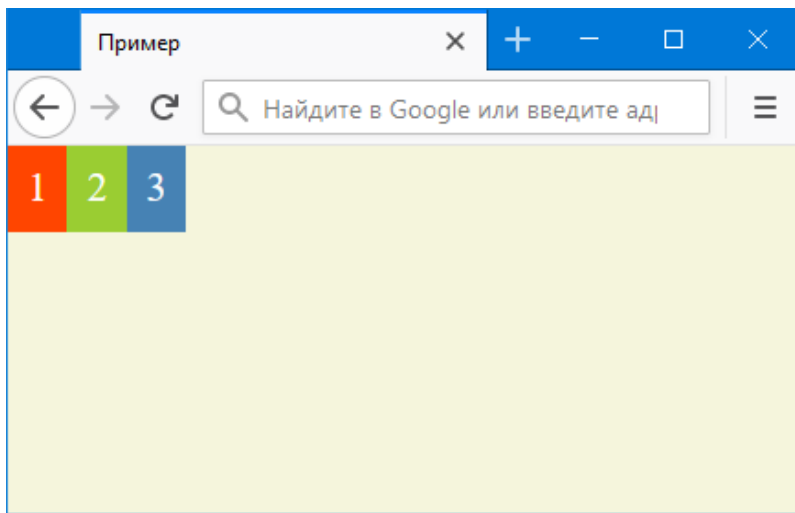
Значения относительно простые — они позволяют указать, располагаются ли флекс-элементы в рядах или колонках, в обратном порядке или нет. Но конечное направление зависит от writing-mode.

Значение row

Это исходное значение, так что если вы не установите свойство [flex-direction](#), именно это значение будет применяться.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```

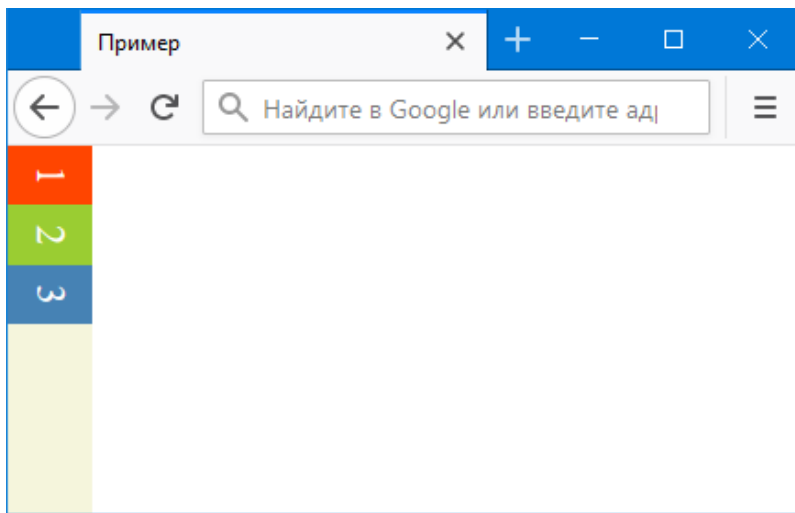


Обратите внимание, что `row` не всегда означает горизонтальный. Режим записи определяет, будут ли ряды идти горизонтально или вертикально. При использовании вертикального режима записи ряд из флекс-элементов будет располагаться сверху вниз.

Вот что произойдёт, если мы применим `writing-mode: vertical-rl` к приведённому выше примеру.

☐ Тёмная тема

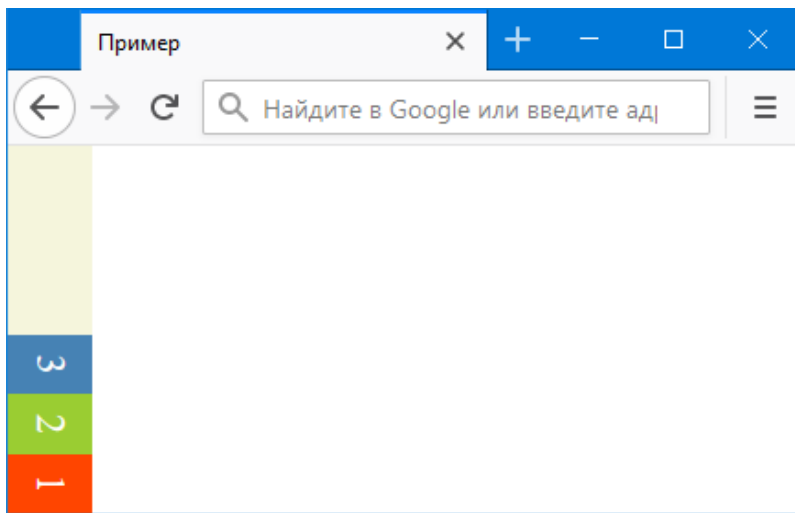
```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
  writing-mode: vertical-rl;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```



А вот что произойдёт, если мы добавим `direction: rtl` в этот пример.

☐ Тёмная тема

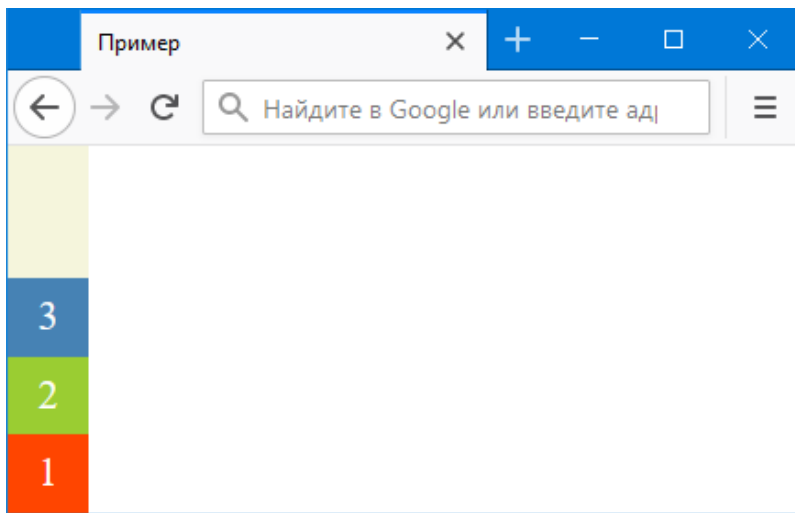
```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
  writing-mode: vertical-rl;
  direction: rtl;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```



Сделаем ещё один шаг вперёд и добавим `text-orientation: upright` в этот пример. Вот что получилось.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
  writing-mode: vertical-rl;
  direction: rtl;
  text-orientation: upright;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```



Таким образом, даже с единственным значением [flex-direction](#), существует множество вариантов, как флекс-элементы будут представлены, выровнены и ориентированы, в зависимости от использования других свойств.

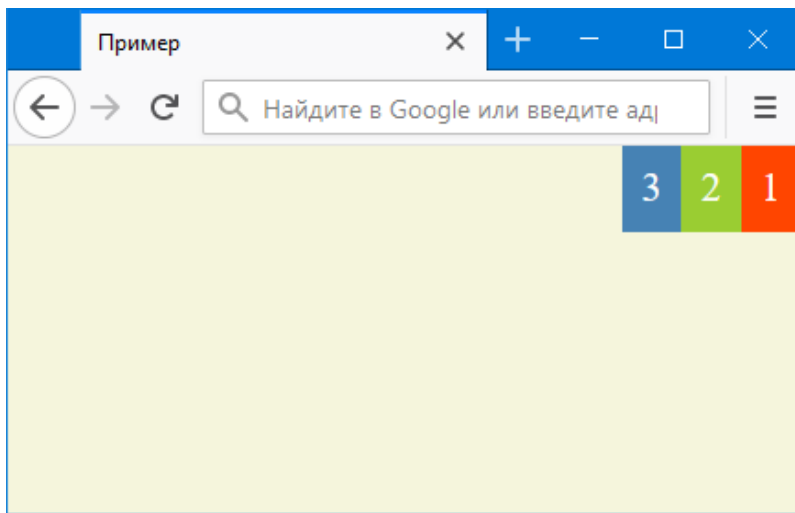
Этот же принцип применим ко всем другим значениям `flex-direction`, показанных ниже.

Значение `row-reverse`

Вы можете использовать `flex-direction: row-reverse`, чтобы элементы в ряду отображались в обратном порядке.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row-reverse;
  align-items: flex-start;
  background: beige;
  height: 100vh;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```

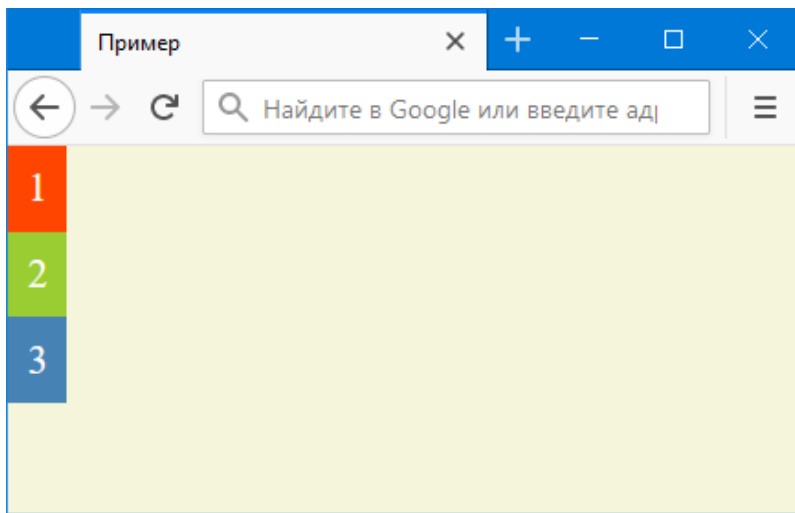


Значение column

Использование flex-direction: column изменяет распределение флекс-элементов так, чтобы они располагались в колонку.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  background: beige;
  height: 100vh;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```

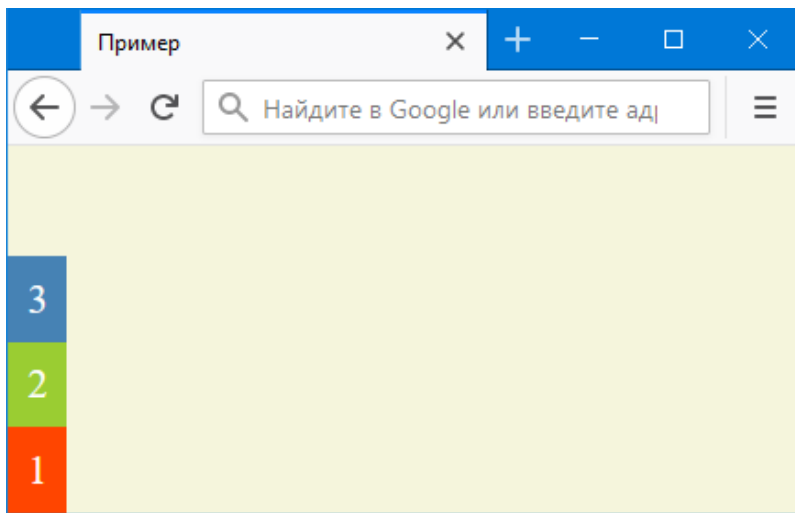


Значение column-reverse

Значение flex-direction: column-reverse устанавливает элементы в колонке в обратном порядке.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: column-reverse;
  align-items: flex-start;
  background: beige;
  height: 100vh;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```



Переносы флекс-элементов

Вы можете использовать свойство [flex-wrap](#) и определить, как флекс-элементы переносятся внутри флекс-контейнера.

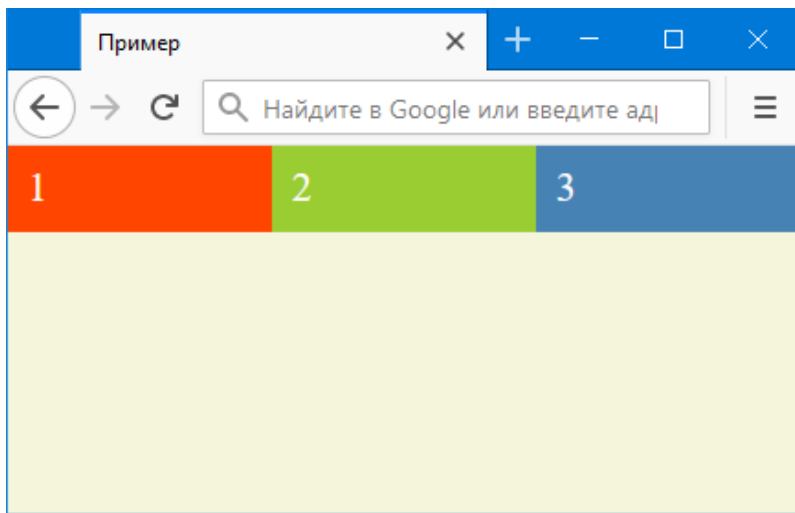
По умолчанию флекс-элементы не переносятся. Если они слишком широкие и не помещаются во флекс-контейнере, то элементы просто сжимаются до тех пор, пока все они не смогут поместиться. Текст внутри флекс-элементов может по-прежнему переноситься, но только не сами флекс-элементы.

Значение nowrap

Вот что я имею в виду (это поведение по умолчанию).

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
  width: 60%;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```



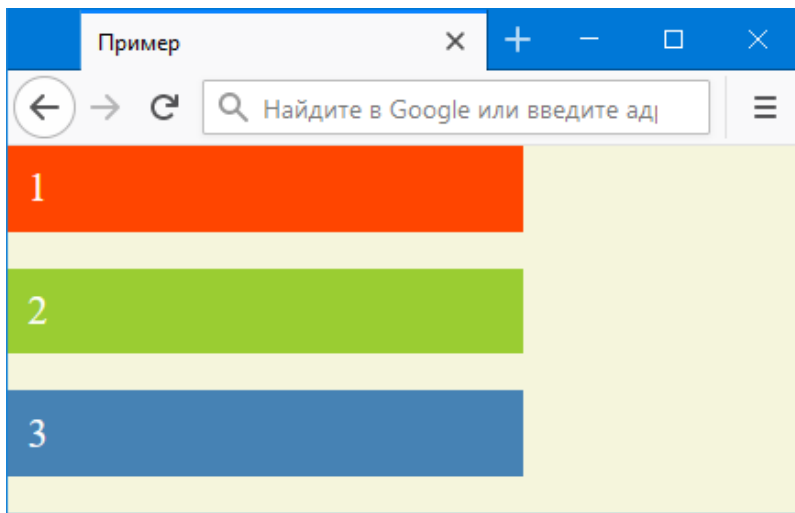
У всех флекс-элементов в данном примере ширина задана как 60%. Это 60 процентов от ширины флекс-контейнера. Вы не можете разместить два элемента такой ширины рядом друг с другом, не говоря уже о трёх. Но поскольку начальное значение [flex-wrap](#) равно `nowrap`, то элементы не переносятся, а вместо этого они сжимаются.

Значение `wrap`

Теперь давайте добавим `flex-wrap: wrap`.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
  flex-wrap: wrap;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
  width: 60%;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```

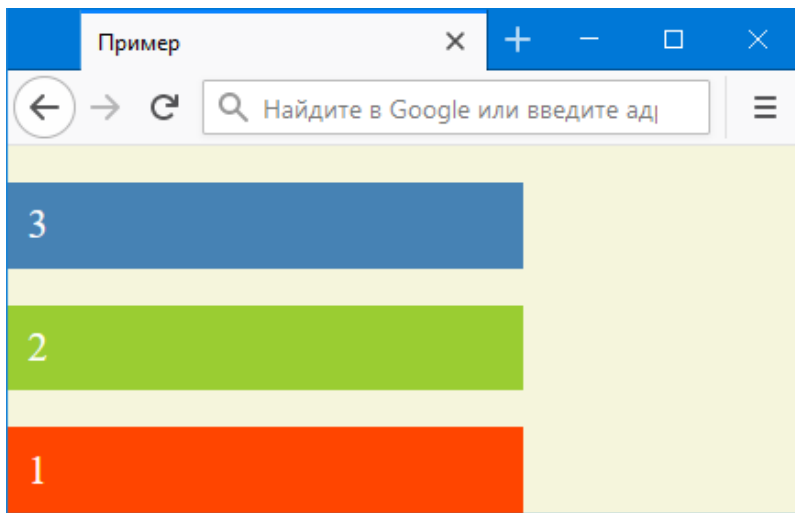


Значение wrap-reverse

Давайте теперь добавим flex-wrap: wrap-reverse.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
  flex-wrap: wrap-reverse;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
  width: 60%;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```



Свойство flex-flow

Свойство [flex-flow](#) является сокращением для свойств [flex-direction](#) и [flex-wrap](#), поэтому вы можете указать их одновременно. Вот пример.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
  background: beige;
  height: 100vh;
  flex-flow: row-reverse wrap;
}
.red {
  background: orangered;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container > div {
  font-size: 5vw;
  padding: .5em;
  color: white;
  width: 60%;
}
body {
  margin: 0;
}
</style>
<div class="container">
  <div class="red">1</div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```

