

???????

Вертикальное выравнивание

Единый класс для строки

Единый класс для строки с фиксированной высотой строки настраивается общее выравнивание

align-items-start	вверху
align-items-center	посередине
align-items-end	снизу

Например, устанавливается высота в 100px и выравнивание посередине для всех колонок

```
<style>
  .middle-row{
    height: 100px;
  }
</style>
<div class="container">
  <div class="row align-items-center middle-row">
    <div class="col-sm-8">col-sm-8</div>
    <div class="col-sm-4">col-sm-4</div>
  </div>
</div>
```

Индивидуальное выравнивание каждого столбца

Класс align-self-* который применяется для класса столбца

```
<style>
  .middle-row{
    height: 100px;
  }
</style>
<div class="container">
  <div class="row align-items-center middle-row">
    <div class="col-sm-8">col-sm-8</div>
```

```
<div class="col-sm-4 align-self-end">col-sm-4</div>
</div>
</div>
```

Горизонтальное выравнивание

`justify-content-*` для выравнивания всех столбцов в классе `row`

<code>justify-content-start</code>	Колонки смещены в начало, между колонками нет промежутка
<code>justify-content-center</code>	Колонки в центре, между колонками нет промежутка
<code>justify-content-end</code>	Колонки смещены в конец, между колонками нет промежутка
<code>justify-content-around</code>	Последующие три класса с промежутками (считая промежуток как колонку) и выравниванием относительно этого. +/- для использования.
<code>justify-content-between</code>	
<code>justify-content-evenly</code>	

Порядок отображения

Классы `order-`, поддерживают контрольные точки. Элементы без порядка имеют приоритет выше.

```
<div class="row">
  <div class="col">
    Первый в DOM, первый в отображении
  </div>
  <div class="col order-5">
    Второй в DOM, последний в отображении
  </div>
  <div class="col order-1">
    Третий в DOM, третий в отображении
  </div>
  <div class="col">
    Последний в DOM, второй в отображении
  </div>
</div>
```

Однако классы `order-first` и `order-last` имеют наивысший приоритет.

Смещение колонок

Классы `offset-контр_точка-*` смещают колонку влево на указанную ширину.

```
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
</div>
```

Класс `offset-контр_точка-0` сбрасывает смещение в контрольной точке.

Автономное использование колонок

Классы `.col-*` также могут использоваться вне `.row`, чтобы дать элементу определенную ширину. Когда классы колонок используются как не прямые дочерние элементы строки, отступы опускаются. Могут использоваться контрольные точки.

```
<div class="col-3 p-3 mb-2">
  .col-3: ширина 25%
</div>

<div class="col-sm-9 p-3">
  .col-sm-9: ширина 75% выше контрольной точки sm
</div>
```

Плавающее изображение

```
<div class="clearfix">
  
  <p>
```

Абзац текста-заполнителя. Мы используем его здесь, чтобы показать использование класса `clearfix`. Мы добавляем здесь довольно много бессмысленных фраз, чтобы продемонстрировать, как столбцы здесь взаимодействуют с плавающим изображением.

```
</p>
```

```
<p>
```

Как видите, абзацы изящно обтекают плавающее изображение. А теперь представьте, как это будет выглядеть с фактическим содержанием здесь, а не просто с этим скучным текстом-заполнителем, который продолжается и продолжается, но на самом деле не передает никакой осязаемой информации. Он просто занимает место, и его не стоит читать.

```
</p>
```

```
<p>
```

И тем не менее, здесь вы, по-прежнему настойчив в чтении этот текст-заполнитель, в надежде еще несколько прозрений, или некоторые скрытые пасхальные яйца содержания. Возможно, шутка. К

сожалению, здесь ничего этого нет.

</p>

</div>

Revision #4

Created 21 August 2024 17:05:45 by Admin

Updated 23 August 2024 15:50:21 by Admin