**Теория 6. Утилиты**

После прочтения темы перейдите к выполнению **урока 6**

**Управление margin и padding отступами**

Bootstrap 5 включает в себя набор классов, с помощью которых можно очень легко добавлять к HTML элементам необходимые margin и padding отступы.

Именуются эти классы следующим образом:

* **{property}{sides}-{size}** для **xs**;
* **{property}{sides}-{breakpoint}-{size}** для **sm**, **md**, **lg**, и **xl**.

Вместо {property} необходимо указать первую букву названия отступа, т.е. m или p (m – для margin, p – для padding).

Вместо {sides} необходимо указать сторону (t - для задания отступа только сверху, b – снизу, l – слева, r – справа) или несколько сторон (x – слева и справа, y – сверху и снизу). Если {sides} **не указать**, то тогда **отступ будет установлен для всех 4 сторон элемента**.

Вместо {breakpoint} нужно указать контрольную точку (sm, md, lg, или xl). Другими словами, определить минимальную ширину viewport, начиная с которой, отступы, задаваемые этим классом, будут действовать.

При этом необходимо помнить, что контрольные точки задают именно минимальную ширину viewport. Это означает, что, например, класс с **sm** будет действовать не только на **sm**, но и на всех следующих контрольных точках (**sm**->**md**->**lg**->**xl**). Т.е. до тех пор, пока его действие будет не переопределено с помощью другого класса (например, с помощью **lg**).

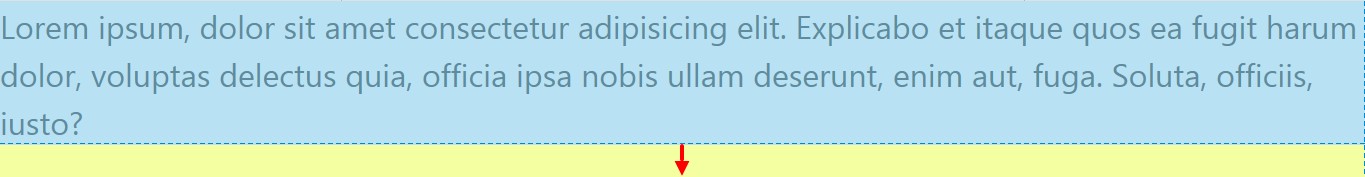
Вместо {size} необходимо задать величину отступа (посредством одного из следующих значений):

* 0 (равно 0);
* 1 (равно $spacer \* .25);
* 2 (равно $spacer \* .5);
* 3 (равно $spacer);
* 4 (равно $spacer \* 1.5);
* 5 (равно $spacer \* 3);
* auto (равно auto) - можно использовать только для margin.

$spacer - это SCSS переменная Bootstrap 5, значение которой, по умолчанию, равно 1rem. А 1rem в большинстве браузеров равен 16px. В итоге, например, размер 4 будет равен 24px (16px \* 1.5).

Рассмотрим примеры использования классов Bootstrap для установления HTML-элементам padding и margin оступов.

**Пример 1**. Установим HTML-элементу padding отступ сверху, равный 3 (рисунок 1):

Рисунок 1 — Пример 1. Управление margin и padding отступами

|  |  |
| --- | --- |
|  | <div class="pb-3">      <p> Lorem ipsum, dolor sit amet consectetur adipisicing elit. Explicabo et itaque quos ea fugit harum dolor, voluptas delectus quia, officia ipsa nobis ullam deserunt, enim aut, fuga. Soluta, officiis, iusto?</p>  </div> |

**Пример 2.** Расположим 2 блока по центру. Для этого первому блоку установим в качестве свойства margin-left значение auto с помощью класса ml-auto, а второму margin-right:auto - с помощью класса mr-auto. Кроме этого дополнительно зададим отступ между блоками, с помощью классов mr-3 и ml-3 (рисунок 2).

Рисунок 2 — Пример 2. Управление margin и padding отступами

<div class="container">

    <div class="row">

        <!-- Первый блок -->

        <div class="col-4 ml-auto mr-3 bg-success" style="height: 200px;"></div>

        <!-- Второй блок -->

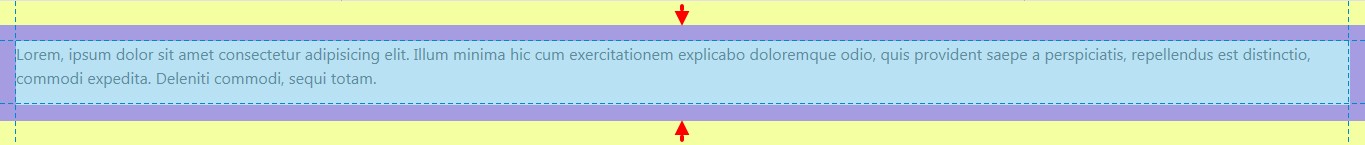
        <div class="col-4 mr-auto ml-3 bg-danger" style="height: 200px;"></div>

    </div>

</div>

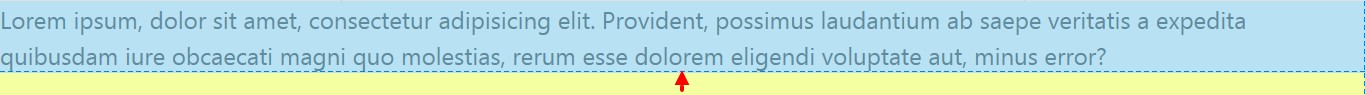
**Пример 3.** Установим блоку (рисунок 3):

* margin отступы слева и справа, равные auto (mx-auto);
* на xs margin отступы сверху и снизу, равные 2 (my-2);
* на sm и md (т.е. >sm) margin отступы сверху и снизу, равные 3 (my-sm-3);
* на lg и xl (т.е. >lg) margin отступы сверху и снизу, равные 4 (my-lg-4);
* на всех контрольных точках padding отступы для 4 сторон, равные 3 (p-3).

Рисунок 3 — Пример 3. Управление margin и padding отступами

|  |  |
| --- | --- |
|  | <div class="mx-auto my-2 my-sm-3 my-lg-4 p-3">      <p> Lorem ipsum, dolor sit amet consectetur adipisicing elit. Explicabo et itaque quos ea fugit harum dolor, voluptas delectus quia, officia ipsa nobis ullam deserunt, enim aut, fuga. Soluta, officiis, iusto?</p>  </div> |

**Пример 4.** Установим элементу на xs margin отступ снизу, равный 4. На остальных контрольных точках этот отступ уберём, т.е. установим его, равным 0 (рисунок 4).

Рисунок 4 — Пример 4. Управление margin и padding отступами

|  |  |
| --- | --- |
|  | <div class="mb-4 mb-sm-0">      <p> Lorem ipsum, dolor sit amet consectetur adipisicing elit. Explicabo et itaque quos ea fugit harum dolor, voluptas delectus quia, officia ipsa nobis ullam deserunt, enim aut, fuga. Soluta, officiis, iusto?</p>  </div> |

Пример 5. Установим HTML-элементу (рисунок 5):

* по умолчанию (min-width >= 0 или xs) padding отступы, равные 0 (p-0);
* на sm и следующих контрольных точках padding отступ слева, равный 3(pl-sm-3);
* на md и следующих контрольных точках (т.е. > md) margin уберём отступ слева (установим его равным 0, т.е. pl-md-0) и добавим отступ справа, равный 4 (pr-sm-4).

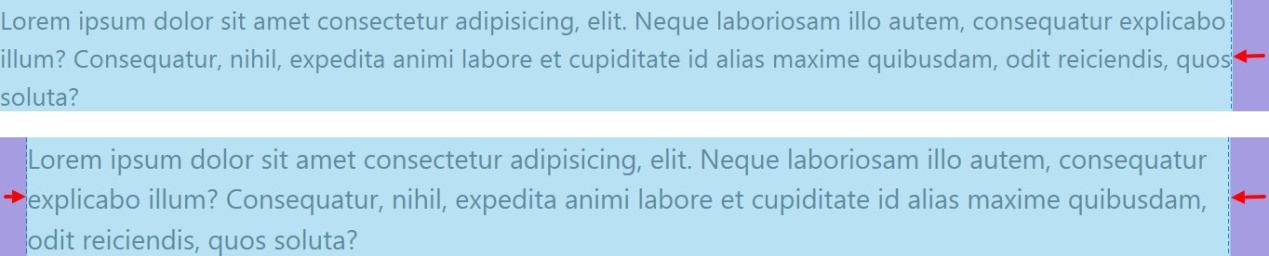


Рисунок 5 — Пример 5. Управление margin и padding отступами

|  |  |
| --- | --- |
|  | <div class="p-0 pl-sm-3 pl-md-0 pr-sm-4">      <p> Lorem ipsum, dolor sit amet consectetur adipisicing elit. Explicabo et itaque quos ea fugit harum dolor, voluptas delectus quia, officia ipsa nobis ullam deserunt, enim aut, fuga. Soluta, officiis, iusto?</p>  </div> |

**Управление границами и скруглениями углов элементов**

**Добавление и удаление границ**

Для управления границами элемента предназначены классы border, border-0, border-top, border-right, border-bottom, border-left, border-top-0, border-right-0, border-bottom-0 и border-left-0.

Класс border устанавливает границы для всех сторон элемента. Классы border-top, border-right, border-bottom и border-left выполняет это действие только для одной из них. А именно класс border-top устанавливает верхнюю границу, border-right – правую, border-bottom – нижнюю, а border-left – левую. Другая часть классов (border-top-0, border-right-0, border-bottom-0 и border-left-0) выполняет обратное действие, а именно убирает одну из границ (border-top-0 – верхнюю, border-right-0 – правую, border-bottom-0 – нижнюю, border-left-0 – левую). Последний класс из этой группы (border-0) предназначен для убирания у элемента всех границ.

Например, для установления верхней и нижней границы HTML элементу можно воспользоваться одним из следующих 2 способов (рисунок 6):

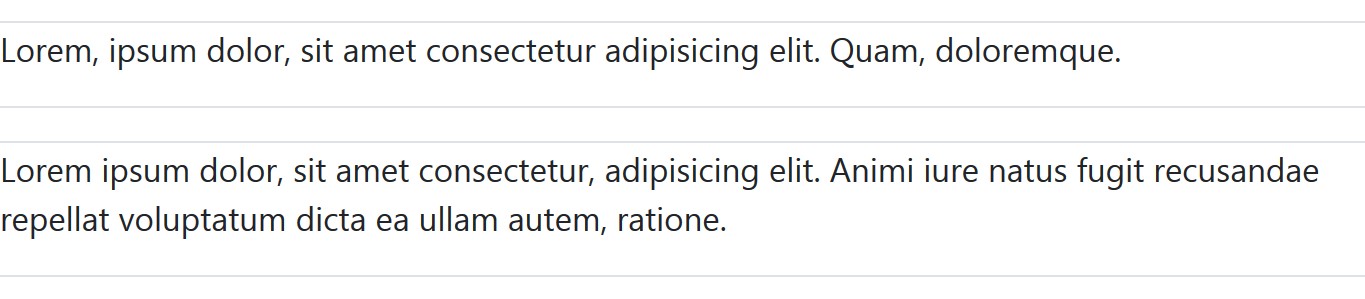


Рисунок 6 — Пример 1. Границы

<!-- Добавляем все границы и убираем 2 (левую и правую) -->

<div class="border border-left-0 border-right-0 mt-3">

    <p>Lorem, ipsum dolor, sit amet consectetur adipisicing elit. Quam, doloremque.</p>

</div>

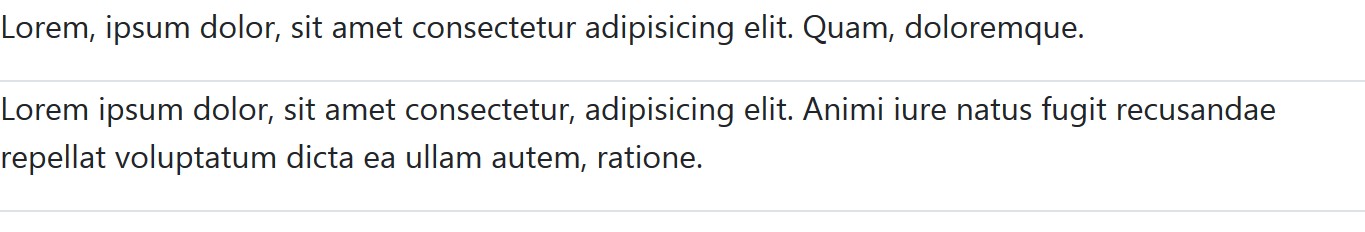
<!-- Добавляем верхнюю и нижнюю границу -->

<div class="border-top border-bottom mt-3">

    <p>Lorem ipsum dolor, sit amet consectetur, adipisicing elit. Animi iure natus fugit recusandae repellat voluptatum dicta ea ullam autem, ratione.</p>

</div>

Пример, как к элементу можно добавить только нижнюю границу (рисунок 7):

Рисунок 7 — Пример 2. Границы

<!-- Добавляем границы с помощью класса border и убираем все кроме левой -->

<div class="border border-top-0 border-right-0 border-left-0">

    <p>Lorem, ipsum dolor, sit amet consectetur adipisicing elit. Quam, doloremque.</p>

</div>

<!-- Устанавливаем границу с помощью border-left -->

<div class="border-bottom">

    <p>Lorem ipsum dolor, sit amet consectetur, adipisicing elit. Animi iure natus fugit recusandae repellat voluptatum dicta ea ullam autem, ratione.</p>

</div>

Например, для того убрать все границы у элемента к нему необходимо добавить класс border-0 (рисунок 8).

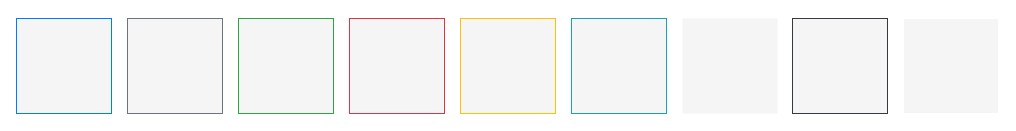
Рисунок 8 — Пример 3. Границы

|  |  |
| --- | --- |
|  | <!-- Удаляем все границы у элемента -->  <div class="border-0">      <p>Lorem, ipsum dolor, sit amet consectetur adipisicing elit. Quam, doloremque.</p>  </div> |

**Классы для изменения цвета границ**

Для задания цвета границ, предназначены классы border-{theme}. Вместо {theme} необходимо указать название темы (primary, secondary, success, danger, warning, info, light, dark или white).

Пример цветных границ (рисунок 9):

Рисунок 9 — Пример 4. Цветные границы

CSS код:

|  |  |
| --- | --- |
|  | .box {      display: inline-block;      width: 5rem;      height: 5rem;      margin: .25rem;      background-color: #f8f9fa;  } |

HTML код:

<span class="border border-primary box"></span>

<span class="border border-secondary box"></span>

<span class="border border-success box"></span>

<span class="border border-danger box"></span>

<span class="border border-warning box"></span>

<span class="border border-info box"></span>

<span class="border border-light box"></span>

<span class="border border-dark box"></span>

<span class="border border-white box"></span>

**Скругление углов элемента**

Во фреймворке Bootstrap 5 имеются классы, с помощью которых можно очень просто скруглять углы элементам. Одни классы (rounded, rounded-circle) выполняют это действие для всех углов, а другие (rounded-top, rounded-right, rounded-bottom, rounded-left) - только для определённых. Кроме этого, в Bootstrap 5 имеется класс (rounded-0), который позволяет выполнить обратное действие, а именно убрать скругление углов.

Принцип действия этих классов основан на использовании CSS свойства border-radius.

Пример, как можно сделать изображение круглыми или со скруглёнными краями (рисунок 10):



Рисунок 10 — Пример 5. Круглые и скругленные границы

CSS код:

|  |  |
| --- | --- |
|  | .box {      width: 80px;      height: 80px;      margin: 4px;  } |

HTML код:

<img src="img/sea-turtle.jpg" class="rounded box" alt="...">

<img src="img/sea-turtle.jpg" class="rounded-top box" alt="...">

<img src="img/sea-turtle.jpg" class="rounded-right box" alt="...">

<img src="img/sea-turtle.jpg" class="rounded-bottom box" alt="...">

<img src="img/sea-turtle.jpg" class="rounded-left box" alt="...">

<img src="img/sea-turtle.jpg" class="rounded-circle box" alt="...">

<img src="img/sea-turtle.jpg" class="rounded-pill box" alt="...">

<img src="img/sea-turtle.jpg" class="rounded-0 box" alt="...">

**Управление отображением элементов**

В Bootstrap 5 имеется категория классов, с помощью которых можно не только адаптивно скрывать элементы, но и управлять тем как они должны быть показаны на странице.

Список display-классов Bootstrap 5:

* d-none
* d-inline
* d-inline-block
* d-block
* d-table
* d-table-cell
* d-flex
* d-inline-flex

Первый класс (d-none) предназначен для скрытия элемента на странице. Осуществляет это он посредством установки HTML элементу CSS свойства display со значение none.

Кроме этого фреймворк Bootstrap 5 позволяет управлять CSS свойством display HTML элементов адаптивно. Это означает, что HTML элемент при одной ширине viewport может отображаться одним образом, а при другой – по-другому.

Синтаксис display-класса Bootstrap 5:

d{-breakpoint}-?

В данном классе нужно вместо:

* breakpoint указать название контрольной точки (sm, md, lg, xl), начиная с которой будет применяться указанное CSS свойство display к HTML элементу (без обозначения - начиная с самых крохотных устройств).
* ? задать одно из значений: none, inline, inline-block, block, table, table-cell, flex, inline-flex.

Список всех display-классов Bootstrap 5:

* d-none
* d-inline
* d-inline-block
* d-block
* d-table
* d-table-cell
* d-flex
* d-inline-flex
* d-sm-none
* d-sm-inline
* d-sm-inline-block
* d-sm-block
* d-sm-table
* d-sm-table-cell
* d-sm-flex
* d-sm-inline-flex
* d-md-none
* d-md-inline
* d-md-inline-block
* d-md-block
* d-md-table
* d-md-table-cell
* d-md-flex
* d-md-inline-flex
* d-lg-none
* d-lg-inline
* d-lg-inline-block
* d-lg-block
* d-lg-table
* d-lg-table-cell
* d-lg-flex
* d-lg-inline-flex
* d-xl-none
* d-xl-inline
* d-xl-inline-block
* d-xl-block
* d-xl-table
* d-xl-table-cell
* d-xl-flex
* d-xl-inline-flex

**Контрольные вопросы**

1. Что такое утилиты?
2. С помощью каких классов можно применить margin и padding?
3. Для чего нужна переменная $spacer в Bootstrap?
4. Как добавить границы?
5. Как убрать верхнюю и нижнюю границы?
6. Как добавить цвет к границам?
7. Как скруглить углы у границ?
8. Что выполняет данный класс .rounded-circle?
9. Какой класс предназначен для скрытия элемента на странице?