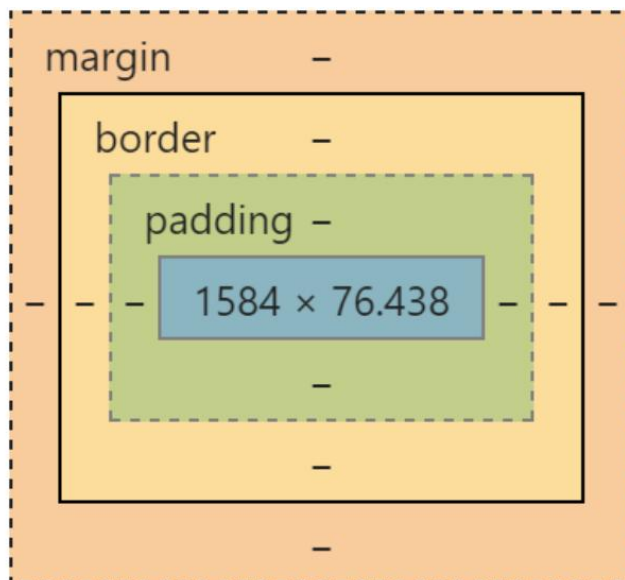


Боксовая модель

Разберём как располагать элементы на странице, то есть поговорим о сетках.

Мы уже знаем немалое количество HTML-тегов. Каждому из этих тегов на странице соответствует прямоугольная область, которая называется боксом (от английского box — «коробка»).

Бокс состоит из содержимого (content), внутренних отступов (padding), рамки (border) и внешних отступов (margin):



То, как бокс выглядит на странице, во многом зависит от его типа (или от типа его родителя). Блочные боксы на странице начинаются с новой строки и растягиваются на всю ширину родительского элемента. Блочный тип по умолчанию имеют, например, теги `<p>`, `<div>` и `<h1>`.

Строчные боксы располагаются друг за другом на одной строке, а их ширина зависит от их содержимого. По умолчанию строчными боксами являются, например, теги `<a>`, `` и ``.

Хорошему верстальщику нужно уметь видеть боксы на странице. Потренируем этот навык, создавая альтернативу сайту колледжа. Для начала создадим новые файлы для работы и наполним их

Задание

1. В папке задания уже лежат некоторые файлы, дополнительно создайте файлы `index.html` и `style.css`
2. Наполнение для `index.html` - стандартный минимум:

```
<!DOCTYPE html>
```

```
<html lang="ru">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<link rel="stylesheet" href="style.css">
<title>Поток HTML документа</title>
</head>
<body>
```

```
</body>
</html>
```

3. В теле создадим структуру нашей страницы, а именно:

a. Сделаем хедер, внутри которого должна находиться картинка логотипа и заголовок первого уровня с названием. Лого можно сохранить перейдя по [ссылке](#). В заголовке напишем “НАТК им. Б.С.Галущака”

b. Теперь создадим основное содержимое страницы с помощью тега <main>. Внутри него создадим неупорядоченный список , содержимым которого будут 6 элементов следующего шаблона:

```
<li class="useful-links">
  <h2>...</h2>
  <p>...</p>
</li>
```

c. Заполните их по образцу



НАТК им. Б.С.Галущака

• Основные сведения

Документация по специальностям, образовательному содержанию и прочие полезные тексты

• Расписание

Расписание занятий на ближайшую неделю

• Контактная информация

Телефоны, адреса, электронные почты отделов и сотрудников

• Преподаватели

Педагогический (научно-педагогический) состав

• Демэкзамен

Демонстрационный экзамен по стандартам Ворлдскиллс

• Совет студентов

Сведения о студенческом самоуправлении

4. Добавьте футер со знаком копирайта (Alt + 0169) и строкой 2022. Все права защищены

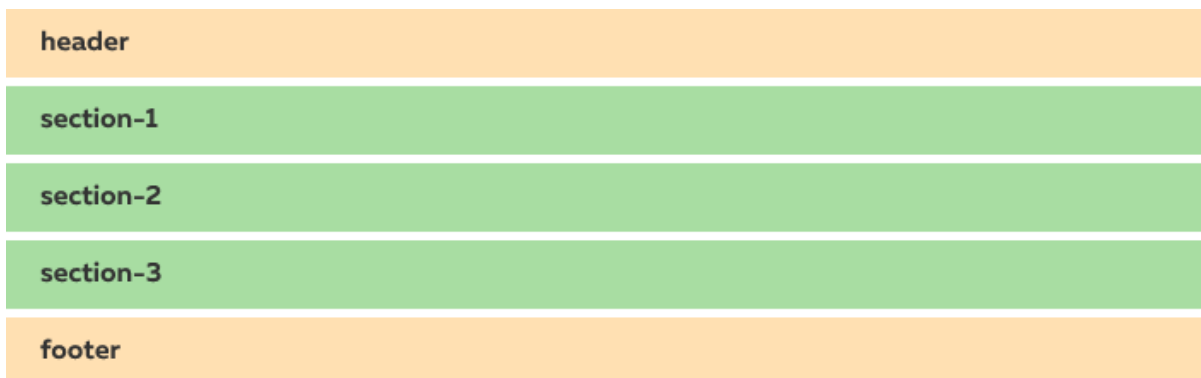
5. В файле style.css обратитесь ко всем элементам и добавьте им границы:

```
* {
  border: 1px solid red;
}
```

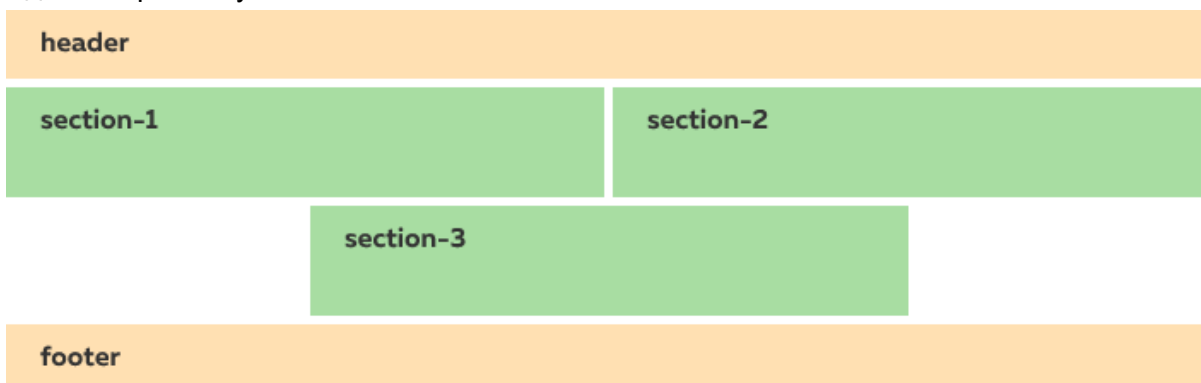
Поток

Теперь, перейдя в браузер и обновив страницу, вы увидите как выглядят боксы. Крайне четко видны границы каждого. Теперь разберём, как ими управлять.

То, как боксы взаимодействуют друг с другом и в каком порядке располагаются на странице, называется потоком. Например, если в потоке находятся блочные боксы, то по умолчанию они выстраиваются сверху вниз и занимают всё доступное пространство по ширине. Такое поведение называют нормальным потоком.



Потоком можно управлять. Например, если изменить с помощью CSS тип боксов и некоторые свойства по умолчанию, то можно заставить боксы выстроиться в несколько колонок или расположиться по центру страницы. Так что одна и та же разметка может выглядеть по-разному.



Научившись изменять поведение боксов в потоке, вы сможете строить сетки любой сложности.

Сеткой называют расположение крупных боксов на странице. К таким боксам обычно относят шапку, подвал сайта, основное (<main>) и дополнительное (<aside>) содержимое, различные секции и разделы. Как правило, количество сеточных элементов на странице не меняется, а их размеры задаются согласно макету.

Макет — это изображение веб-страницы. Его создаёт дизайнер, а веб-разработчик использует его как образец при вёрстке. Макеты можно создавать в графическом редакторе или в специальных программах.

Вот макет страницы, которую нам предстоит сверстать:

 НАТК им. Б.С.Галушца	
Основные сведения Документация по специальностям, образовательному содержанию и прочие полезные тексты	Расписание Расписание занятий на ближайшую неделю
Контактная информация Телефоны, адреса, электронные почты отделов и сотрудников	Преподаватели Педагогический (научно-педагогический) состав
Демэкзамен Демонстрационный экзамен по стандарту Ворлдскиллс	Совет студентов Сведения о студенческом самоуправлении
© 2022. Все права защищены	

Разметка страницы и декоративные стили уже готовы, осталось только их подключить и правильно сверстать сетку. Декоративные стили были вынесены в отдельный файл. В обычной жизни так не поступают, но так проще учиться работать с сеткой.

Начнём с того, что подключим на страницу еще пару стилевых файлов: с декоративными стилями и пустой, в котором будем писать сеточные стили. Всего файлов будет 3

Задание

1. Подключите готовый файл стилей main-style.css
2. Подключите готовый файл стилей grid.css

Свойство display, тип бокса

Мы подключили на страницу декоративные стили и подсветили сеточные элементы. Теперь пойдём сверху вниз по макету и будем шаг за шагом выстраивать сетку страницы.

Первый сеточный элемент на странице — шапка. Она обозначается тегом `<header>`, который по умолчанию является блочным боксом. Внутри него находятся логотип (изображение) и заголовок.

```
<header>
  
  <h1>НАТК им. Б.С.Галушца</h1>
</header>
```

Логотип занимает столько места, сколько нужно его содержимому. Заголовок же имеет блочный тип, поэтому он начинается с новой строки и тянется на всю ширину шапки. Из-за этого логотип и заголовок сейчас располагаются друг под другом. Но на макете они находятся рядом, как если бы были выстроены в две колонки.

Расположить элементы в две колонки можно разными способами. Мы воспользуемся гридами (от английского `grid` — «сетка»). Это удобная и популярная технология. И у неё неплохая браузерная поддержка.

Чтобы с помощью гридов управлять расположением логотипа и заголовка, нужно изменить тип бокса у их родителя — тега `<header>`. За тип бокса в CSS отвечает свойство `display`. У этого свойства больше десятка возможных значений, все они перечислены в спецификации.

```
display: grid;
```

Бокс с типом `grid` называют грид-контейнером, а дочерние, то есть непосредственно вложенные в него теги — грид-элементами.

Хотя снаружи (для других элементов, например основного содержимого) грид-контейнер ничем не отличается от блочного бокса, грид-элементы внутри него ведут себя иначе. Например, даже строчные боксы начинают занимать всю доступную им область. Кроме того, в грид-контейнере по-другому ведут себя внешние отступы у элементов.

Превратим шапку в грид-контейнер и посмотрим, как поведут себя грид-элементы — логотип и заголовок. А в следующем задании займёмся колонками

Задание

1. Для удобства зададим хедеру атрибут `class` со значением `page-header`
2. Обратимся к нему в файле `grid.css` по селектору классов:
`.page-header{`

```
}
```

3. Добавьте в это правило свойство `display` со значением `grid`.

Свойство grid-template-columns, шаблон грид-контейнера

Нам нужно расположить логотип и заголовок в шапке в две колонки. Мы сделали шапку грид-контейнером, однако нескольких колонок не появилось. Всё потому, что по умолчанию грид-контейнер одноколоночный.

Чтобы это изменить, нужно описать шаблон грид-контейнера. Для этого используют свойство grid-template-columns:

```
.grid-container {  
  display: grid;  
  grid-template-columns: 100px 150px 80px;  
}
```

Код в примере разобьёт содержимое грид-контейнера на три колонки шириной 100, 150 и 80 пикселей. Грид-элементы автоматически распределятся по колонкам слева направо.

В шапке нам нужно получить две колонки шириной 210 и 200 пикселей

Потренируемся использовать свойство grid-template-columns, а после разделим содержимое шапки на колонки правильного размера

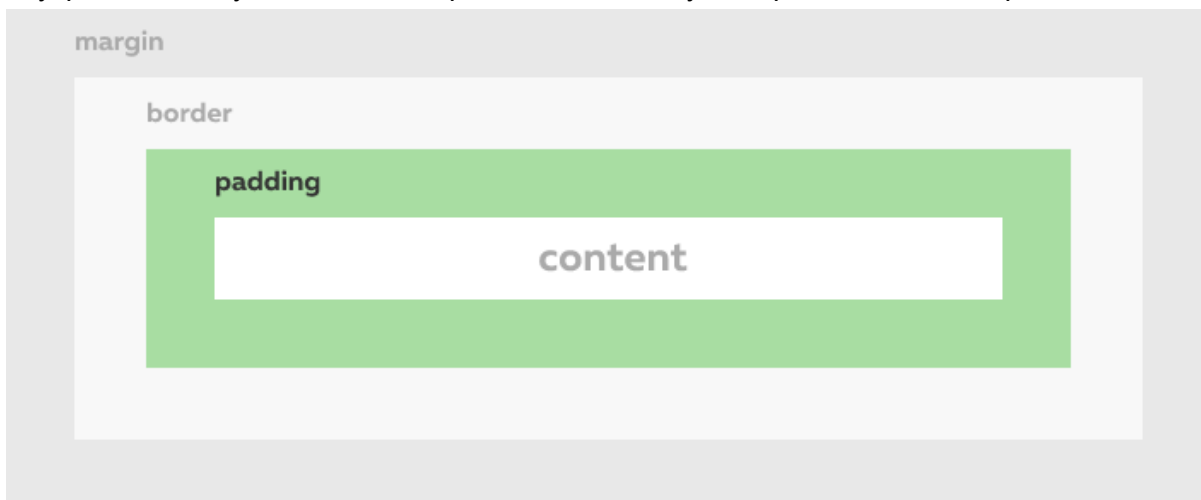
Задание

1. Добавьте в правило для класса page-header свойство grid-template-columns со значением 10px 400px Логотип и заголовок встанут в ряд, причём заголовок наложится на логотип
2. Замените 10px на 100px Теперь логотип и заголовок расположатся почти как на макете
3. Добавьте классу .page-header пару строк:
 - a. justify-content: center; - располагает дочерние элементы по центру горизонтально
 - b. align-items: center; - располагает дочерние элементы по центру вертикально

Свойство padding, внутренний отступ

Мы выстроили содержимое шапки в две колонки, как на макете. Но в шапке не хватает «воздуха». Чтобы элементы не прилипали к границам бокса, ему нужно добавить внутренние отступы.

Внутренним отступом называют расстояние между содержимым бокса и рамкой.



Внутренние отступы у элемента создают с помощью свойства `padding`. Если внутренние отступы одинаковы со всех сторон, то достаточно написать так:

```
.element {  
  padding: 15px;  
}
```

Код в примере задаст элементу внутренние отступы 15px со всех сторон. Такую запись называют краткой. Есть и другие виды краткой записи, о них мы поговорим чуть позже.

Если отступы с разных сторон различаются, то используют полную запись, указывая внутренний отступ отдельно для каждой стороны:

```
.element {  
  padding-top: 5px;  
  padding-right: 10px;  
  padding-bottom: 15px;  
  padding-left: 20px;  
}
```

Свойство `padding-top` создаёт внутренний отступ сверху, `padding-right` — справа, `padding-bottom` — снизу, а `padding-left` — слева.

Необязательно задавать элементу внутренние отступы со всех сторон. Например, если внутренний отступ нужен только сверху, то достаточно использовать лишь `padding-top`.

С разных сторон в шапке разные отступы. Чтобы задать их, используем полную запись.

Если с трёх сторон отступы одинаковы, а с четвёртой отличаются, то иногда используют такой трюк:

```
.element {
```

```
padding: 15px;  
padding-bottom: 30px;  
}
```

Таким образом сначала элементу задают одинаковые отступы со всех сторон, а после переопределяют отличающуюся сторону

Задание

1. Добавьте шапке внутренние отступы — padding:
 - a. сверху — 10px
 - b. справа — 10px
 - c. снизу — 10px
 - d. слева — 20px

Свойство margin, внешний отступ

Мы добавили шапке внутренние отступы. Но снаружи шапки на странице осталось свободное пространство. Откуда оно там взялось?

Дело в том, что многие современные браузеры добавляют тегу `<body>` внешние отступы 8px со всех сторон. Внешним отступом называют отступ от внешней границы элемента до границ родительского элемента или до соседних элементов.



Чтобы управлять внешними отступами, используют свойство `margin`. У него, как и у `padding`, есть краткая и полная записи.

// Краткая запись

```
margin: 20px;
```

// Полная запись

```
margin-top: 0;
```

```
margin-right: 5px;
```

```
margin-bottom: 10px;
```

```
margin-left: 15px;
```

Свойство `margin-top` создаёт внешний отступ сверху, `margin-right` — справа, `margin-bottom` — снизу, а `margin-left` — слева.

Указывать все четыре свойства необязательно. Если требуется задать внешний отступ лишь с одной или нескольких сторон, то достаточно использовать только соответствующие свойства.

На макете у `<body>` внешних отступов нет, поэтому и со страницы их нужно убрать. Лишние отступы также есть и у следующего сеточного элемента на нашей странице — списка преимуществ. Он размечен с помощью тега ``. По умолчанию браузеры добавляют неупорядоченному списку внутренние и внешние отступы. Их придётся переопределить.

Итак, первым делом уберём внешние отступы у `<body>`, потом займёмся списком преимуществ. У списка есть класс `features-list`, используем его для стилизации. Сначала обнулим у списка внешние отступы сверху и снизу, а потом зададим внутренние отступы по 20px со всех сторон.

Если какому-либо свойству задают нулевое значение, то единицы измерения обычно не указывают

Задание

1. Создайте правило для тега `body`
2. Уберите у страницы внешний отступ по умолчанию: `margin: 0;`
3. Создайте правило для `ul`
4. Уберите у списка внешние отступы `margin` сверху и снизу.
5. Задайте списку внутренние отступы `20px` со всех сторон

fr, доля доступного пространства

Мы переопределили браузерные стили для тега `<body>` и списка преимуществ, теперь займёмся потоком. Содержимое списка должно располагаться в две колонки.

Чтобы сделать колонки, превратим список в грид-контейнер.

В списке преимуществ шесть элементов, а разбить их надо на две колонки. Если элементов в грид-контейнере больше, чем колонок, то следующие элементы автоматически переносятся на новую строку, или ряд, и так же разделяются на колонки. Например, в нашем случае получится три ряда.

По замыслу дизайнера, колонки в списке должны быть одинаковыми по размеру и занимать всё доступное пространство по ширине. Как это сделать, ведь мы не можем знать заранее, какого размера монитор у пользователя?

Используем для размера колонок особую единицу измерения — `fr` (сокращённое от `fraction` — «доля»). Она означает долю доступного пространства в грид-контейнере.

```
.grid-container {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
}
```

Грид-контейнер в примере будет поделён на 3 равные части. Первая колонка получит одну часть ширины грид-контейнера, а вторая колонка — две части. Как бы ни изменялась ширина контейнера, пропорции колонок всегда будут одинаковыми.

`fr` можно использовать и вместе с пикселями. Например, вот так можно создать сетку, где правая колонка имеет фиксированную ширину `200px`, а левая занимает всё оставшееся пространство:

```
.grid-container {  
  display: grid;  
  grid-template-columns: 1fr 200px;  
}
```

Потренируемся задавать размеры в `fr`, чтобы лучше понять, как она работает. А после разделим содержимое списка преимуществ на две колонки одинаковой ширины

Задание

1. Добавьте списку свойство `display` со значением `grid`
2. А теперь — свойство `grid-template-columns` со значением `2fr 1fr` Ширина первой колонки окажется в два раза больше ширины второй колонки.
Убедиться, что пропорции колонок сохраняются, можно изменив ширину браузера. Для этого необходимо перейти в режим разработчика (клавиша `F12`, или `Ctrl + Shift + I`, или ПКМ - Просмотреть код. Потяните края в режиме отзывчивости — блоки будут пропорционально изменяться.
3. Измените значение `grid-template-columns` на `100px 1fr` Ширина первой колонки станет `100px`, а вторая займёт всё оставшееся пространство.

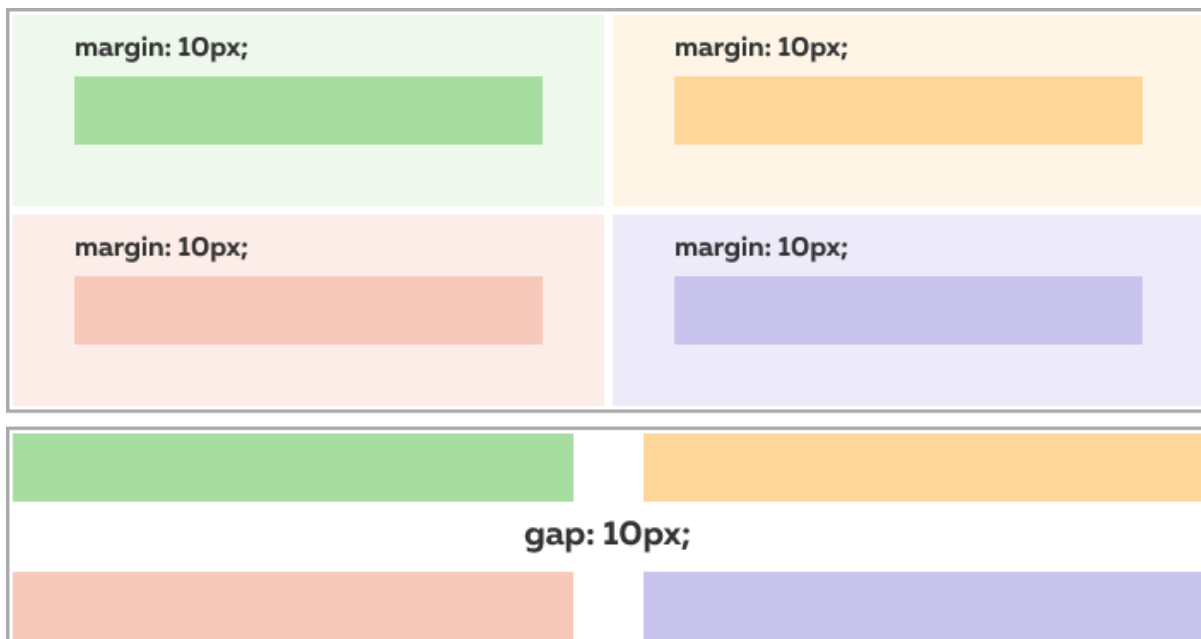
4. Наконец, измените значение `grid-template-columns` на `1fr 1fr` Колонки делят ширину родителя поровну.

Свойство gap, отступ между грид-элементами

Теперь нам не хватает отступов между элементами. Добавим их.

Использовать в данном случае `margin` не очень удобно, ведь он также влияет на расстояние до границ родительского элемента. А ещё в грид-контейнере внешние отступы двух соседних элементов складываются и по горизонтали и по вертикали. Так что расстояние между элементами оказывается в два раза больше указанного.

Как быть? Используем свойство грид-контейнера `gap`. Оно задаёт расстояние между грид-элементами, но не влияет на расстояние между элементами и контейнером. Сравните:



Обратите внимание, что `gap` добавляется грид-контейнеру, в то время как `margin` — элементам.

С помощью `gap` отступы можно указать отдельно по вертикали и по горизонтали: `column-gap` отвечает за расстояние между колонками, а `row-gap` — за расстояние между рядами.

```
.grid-container {  
  column-gap: 15px;  
  row-gap: 5px;  
}
```

Если же отступы одинаковы, удобно использовать составное свойство `gap`:

```
.grid-container {  
  gap: 20px;  
}
```

Потренируемся указывать отступы между грид-элементами с помощью `column-gap` и `row-gap`. А после используем `gap`, чтобы задать элементам отступы как на макете.

Если указать для `gap` два значения, то первое задаст интервал между рядами, а второе — между колонками:

```
.grid-container {  
  gap: 20px 50px;  
}
```

Эта запись короче, но её труднее понять. Поэтому для разных отступов лучше использовать `row-gap` и `column-gap`

Задание

1. Добавьте списку `ul` отступ между рядами `row-gap: 5px`, сразу ниже назначения `grid-template-columns`;
2. А теперь добавьте отступ между колонками `column-gap: 15px`;
3. Удалите `row-gap` и `column-gap` и добавьте списку преимуществ свойство `gap` со значением `10px`

Завершаем страницу колледжа

Страница практически готова. Из сеточных элементов остался только подвал страницы.

Подвал обозначается тегом `<footer>`, это блочный бокс. Изменять тип бокса в данном случае не требуется, ведь в подвале нет колонок. Но для соответствия макету нужно добавить ему внутренние отступы.

Для стилизации используем обращение по тегу - `footer`.

В завершение отключим подсветку сеточных элементов, она больше не нужна.

На этой странице мы использовали гриды для шапки и списка преимуществ и блочные боксы для подвала и самого верхнего бокса на странице - тега `<body>` (его тип тоже можно изменить при необходимости). Грид-контейнеры нам понадобились, чтобы изменить поток и выстроить элементы в несколько колонок. Там, где поток менять не требовалось, мы использовали тип бокса по умолчанию. Мы также научились задавать элементам фиксированные внутренние и внешние отступы.

Гриды — очень мощная и гибкая технология, и мы пока рассмотрели только часть её возможностей.

Задание

1. Создайте правило для тега `footer`
2. В этом правиле задайте внутренние отступы со всех сторон `20px`
3. В `index.html` на 6 строке удалите подключение файла `style.css`, в котором мы записывали добавление рамок всем элементам

Конспект «Сетки». Раздел 1

Бокс

Каждому тегу на странице соответствует прямоугольная область, которая называется боксом (от английского box — «коробка»).

Бокс состоит из содержимого (content), внутренних отступов (padding), рамки (border) и внешних отступов (margin):

То, как бокс выглядит на странице, во многом зависит от его типа (или от типа его родителя).

Блочные боксы на странице начинаются с новой строки и растягиваются на всю ширину родительского элемента. Блочный тип по умолчанию имеют, например, теги `<p>`, `<div>` и `<h1>`.

Строчные боксы располагаются друг за другом на одной строке, а их ширина зависит от их содержимого. По умолчанию строчными боксами являются, например, теги `<a>`, `` и ``.

Поток, сетки и макет

То, как боксы взаимодействуют друг с другом и в каком порядке располагаются на странице, называется потоком. Поток можно управлять, изменяя тип боксов и свойства по умолчанию.

Сеткой называют расположение крупных боксов на странице. К таким боксам обычно относят шапку, подвал сайта, основное (`<main>`) и дополнительное (`<aside>`) содержимое, различные секции и разделы. Как правило, количество сеточных элементов на странице не меняется, а их размеры задаются согласно макету.

Макет — это изображение веб-страницы. Его создаёт дизайнер, а веб-разработчик использует его как образец при вёрстке.

Свойство padding

Внутренним отступом называют расстояние между содержимым бокса и рамкой.

Внутренние отступы у элемента создают с помощью свойства `padding`. Если внутренние отступы одинаковы со всех сторон, то достаточно написать так:

```
.element {  
  padding: 15px;  
}
```

Такую запись называют краткой.

Если отступы с разных сторон различаются, то используют полную запись, указывая внутренний отступ отдельно для каждой стороны:

```
.element {  
  padding-top: 5px;  
  padding-right: 10px;  
  padding-bottom: 15px;  
  padding-left: 20px;  
}
```


Свойство padding-top создаёт внутренний отступ сверху, padding-right — справа, padding-bottom — снизу, а padding-left — слева.

Свойство margin

Внешним отступом называют отступ от внешней границы элемента до границ родительского элемента или до соседних элементов.

Чтобы управлять внешними отступами, используют свойство margin. У него, как и у padding, есть краткая и полная записи.

// Краткая запись

```
margin: 20px;
```

// Полная запись

```
margin-top: 0;
```

```
margin-right: 5px;
```

```
margin-bottom: 10px;
```

```
margin-left: 15px;
```

Свойство margin-top создаёт внешний отступ сверху, margin-right — справа, margin-bottom — снизу, а margin-left — слева.

Свойство display

За тип бокса в CSS отвечает свойство display. У этого свойства больше десятка возможных значений, все они перечислены в [спецификации](#).

```
display: grid;
```

Grid

Бокс с типом grid называют грид-контейнером, а дочерние, то есть непосредственно вложенные в него теги — грид-элементами.

Хотя снаружи (для других элементов, например основного содержимого) грид-контейнер ничем не отличается от блочного бокса, грид-элементы внутри него ведут себя иначе. Например, даже строчные боксы начинают занимать всю доступную им область. Кроме того, в грид-контейнере по-другому ведут себя внешние отступы у элементов.

По умолчанию грид-контейнер одноколоночный. Чтобы это изменить, нужно описать шаблон грид-контейнера. Для этого используют свойство grid-template-columns:

```
.grid-container {  
  display: grid;  
  grid-template-columns: 100px 150px 80px;  
}
```

Существуют и другие свойства для описания шаблона грид-контейнера. Например, grid-template-rows и grid-template-areas.

Если элементов в грид-контейнере больше, чем колонок, то следующие элементы автоматически переносятся на новую строку, или ряд, и так же разделяются на колонки.

fr

fr (сокращённое от fraction — «доля») — особая единица измерения. Она означает долю доступного пространства в грид-контейнере.

```
.grid-container {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
}
```

Грид-контейнер в примере будет поделен на 3 равные части. Первая колонка получит одну часть ширины грид-контейнера, а вторая колонка — две части. Как бы ни изменялась ширина контейнера, пропорции колонок всегда будут одинаковыми.

fr можно использовать и вместе с пикселями. Например, вот так можно создать сетку, где правая колонка имеет фиксированную ширину 200px, а левая занимает всё оставшееся пространство:

```
.grid-container {  
  display: grid;  
  grid-template-columns: 1fr 200px;  
}
```

Свойство gap

Свойство gap задаёт расстояние между грид-элементами, но не влияет на расстояние между элементами и контейнером.

Свойство gap добавляется грид-контейнеру, в то время как margin — элементам.

С помощью gap отступы можно указать отдельно по вертикали и по горизонтали: column-gap отвечает за расстояние между колонками, а row-gap — за расстояние между рядами.

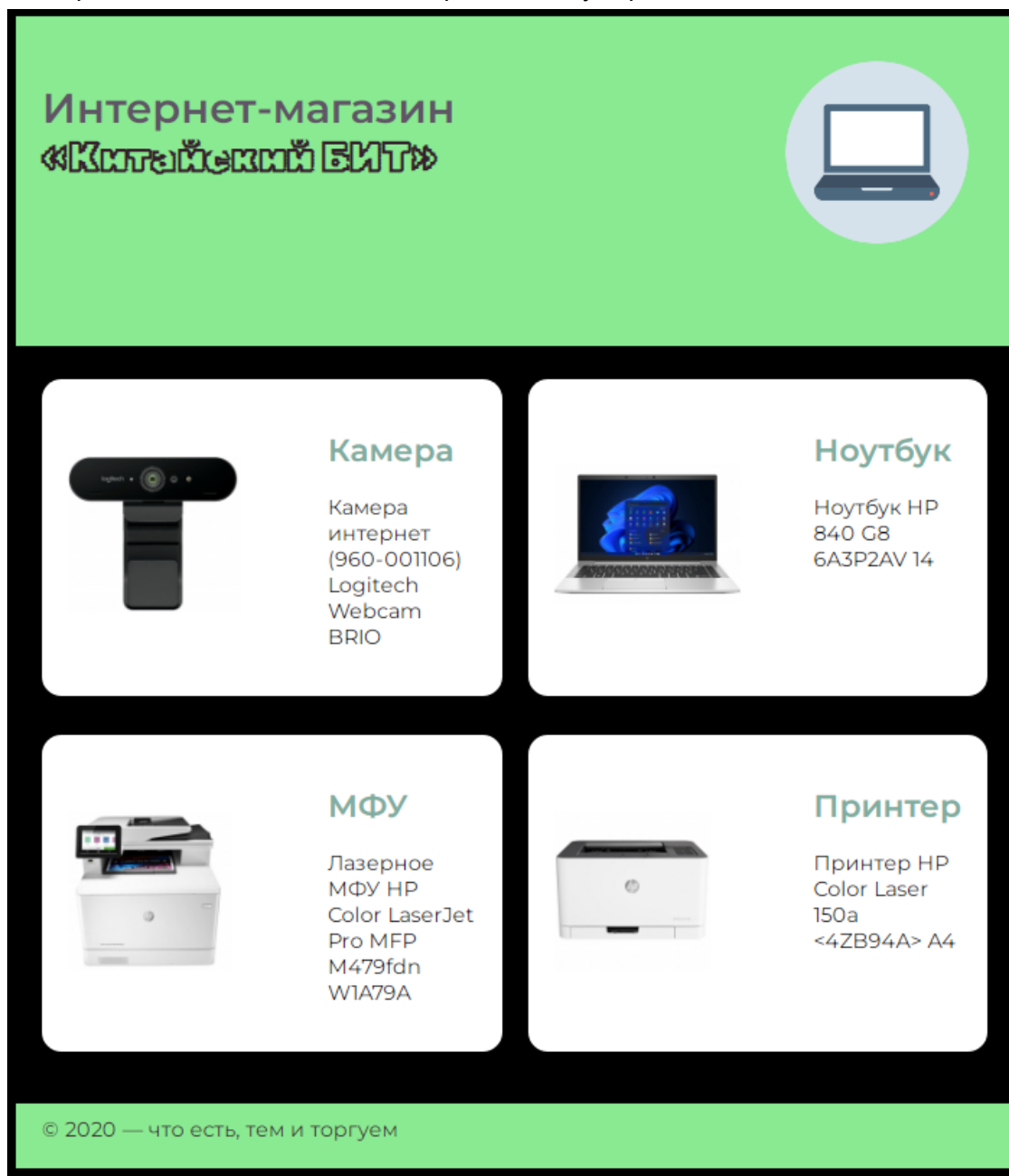
```
.grid-container {  
  column-gap: 15px;  
  row-gap: 5px;  
}
```

Если же отступы одинаковы, удобно использовать составное свойство gap:

```
.grid-container {  
  gap: 20px;  
}
```

Испытание: Интернет магазин

Вам предстоит самостоятельно сверстать сетку страницы:



Разметка и декоративные стили готовы (все необходимые файлы лежат в папке unit), осталось написать сеточные стили. Советуем сразу обнулить внешние отступы у <body>, а для создания колонок использовать гриды.

В шапке должно быть две колонки: правая имеет фиксированный размер 160px, а левая занимает всё оставшееся пространство. Не забудьте про внутренние отступы: сверху 30, справа и слева по 20 и снизу 70 пикселей

Список дней также нужно разделить на две колонки, на этот раз одинаковой ширины. Колонки должны занимать все доступное пространство. Не забудьте убрать у списка внешние отступы по умолчанию и переопределить внутренние отступы - сверху

25, справа и слева по 20 и снизу 40 пикселей. Обратите особое внимание на отступы между элементами, у рядов - 30 пикселей и колонок 20 пикселей.

Подвал страницы довольно простой, но у него отличается внутренний отступ сверху - 10 пикселей, остальные по 20