

Практическая работа 1. Построение сеток на гридах по макету

Цель работы – опираясь на графический макет, нужно построить сетку крупных блоков страницы с использованием технологии *CSS Grid Layout*. Сетка должна быть спроектирована с учётом разных вариантов переполнения (переполнения контентом, изменения количества колонок) и должна визуально соответствовать макету.

Зачем нужен этот навык

Грамотно построенная сетка не просто выстраивает элементы как на макете, но и делает свёрстанную страницу «пуленепробиваемой»: все элементы будут вести себя предсказуемо, не развалятся при изменении контента и не потребуют постоянных доработок из-за поломки в непредсказуемом месте. Механизм гридов — самый мощный инструмент для построения сеток на данный момент. С его помощью собираются как простые раскладки, которые можно построить и с помощью флексбоксов, так и необычные, которые другим способом, кроме гридов, на CSS собрать нельзя.

Освоив этот навык, можно научиться оценивать макеты с точки зрения возможности их вёрстки на гридах, разбирать макеты на блоки, пригодные для структурирования их с помощью гридов, а также работать с вложенностью грид-контейнеров.

В качестве демонстрационного примера выберем макет страницы кабинета ученика языковой школы «Let's Speak». Это типичная страница личного кабинета пользователя.

[Макет](#) для работы предоставлен в формате Figma. О том, как работать с макетами в формате Figma, читайте в методических материалах 1-го модуля «Экспорт параметров и графики из Figma».


Итак, приступим к подробному рассмотрению макета страницы кабинета ученика.


В шапке страницы кабинета ученика размещается поле для поиска по сайту, ссылка на профиль пользователя и кнопка для управления пользовательским меню, область уведомлений. Также на странице имеется навигационное меню. Центральную часть страницы занимает слайдер с промо-акциями и актуальными предложениями от языковой школы. Также есть область для просмотра видеоуроков и плейлист. Справа расположена панель с контактами ученика. В подвале сайта стандартно расположены копирайт и ссылка на главную страницу языковой школы.

Подробно разберём как построить сетку для страницы с помощью CSS Layout Grid. Построим сетки для каркаса сайта и отдельных элементов. Выясним, в каких случаях лучше не использовать сетки на гридах, а ограничиться флексами.

1. Приступим к созданию страницы кабинета ученика языковой школы «Let's Speak». Создадим папку проекта и добавим в нее файлы

index.html и style.css. На первом шаге добавим тег meta, чтобы сайт «правильно» отображался и на мобильных устройствах. Укажем заголовок для сайта. Подключим необходимые шрифты.

 index.html

 style.css

В качестве «песочницы» используем пример в codepen.

Изменения в **index.html** [<https://codepen.io/Irina64/pen/ExpzzbN>]

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;500;900&display=swap"
rel="stylesheet">
    <title>Let's Speak, языковая школа</title>
  </head>
  <body>

</body>
</html>
```

2	Добавим разметку элементов страницы, в которую мы сразу включили основные классы. Подробно о том, как правильно создавать семантическую разметку, читайте в статье «Создание семантической разметки по макету».	Изменения в index.html [https://codepen.io/Irina64/pen/oNMRREo]
---	---	--

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;500;900&display=swap"
rel="stylesheet">
    <title>Let's Speak, языковая школа</title>
  </head>
  <body class="page">
    <nav class="page-menu">
      <a class="logo" href="#">
        
      </a>
      <ul class="page-menu-list">
        <li class="page-menu-item page-menu-item-active">
          <a class="page-menu-link page-menu-link-cab">Кабинет студента</a>
        </li>
        <li class="page-menu-item">
          <a class="page-menu-link page-menu-link-class" href="#">Учебный класс</a>
        </li>
        <li class="page-menu-item">
```

```

    <a class="page-menu-link page-menu-link-book" href="#">Методичка</a>
  </li>
  <li class="page-menu-item">
    <a class="page-menu-link page-menu-link-home" href="#">Домашняя работа</a>
  </li>
  <li class="page-menu-item">
    <a class="page-menu-link page-menu-link-exam" href="#">Экзамены</a>
  </li>
</ul>
<ul class="page-menu-tools">
  <li class="page-menu-item">
    <a class="page-menu-link page-menu-link-settings" href="#">Настройки</a>
  </li>
</ul>
</nav>
<header class="page-header">
  <form class="search" method="GET" action="#">
    <label class="visually-hidden" for="search">Поле поиска</label>
    <input class="search-field" type="text" name="search" id="search"
placeholder="Поиск...">
  </form>
  <div class="profile">
    <a class="profile-name item-with-picture" href="#" data-text="О">Опознанный
студент</a>
    <button class="profile-btn" type="button">
      <span class="visually-hidden">Открыть/Закрыть меню</span>
    </button>
  </div>
  <button class="notifications" type="button">
    <span>1</span>
  </button>
</header>
<main class="page-main">
  <h1 class="visually-hidden">Главная страница сайта языковой школы Let's
Speak</h1>
  <div class="slider">
    
    <p class="slider-headline">«Бифитер», «Кокни», «Даблдекер»...</p>
    <p class="slider-text">Пройдите тест и узнайте насколько хорошо вы знакомы с
жизнью в Лондоне</p>
    <a href="#">Пройти тест →</a>
    <button class="slider-btn slider-btn-prev" type="button"><span
class="visually-hidden">Назад</span></button>
    <button class="slider-btn slider-btn-next" type="button"><span
class="visually-hidden">Вперёд</span></button>
  </div>
  <section class="offers">
    <h2 class="visually-hidden">Интересные предложения</h2>
    <ul class="offers-list">
      <li class="offers-item">
        <h3>Английский завтрак</h3>
        <p class="offers-item-text">Разговорный клуб</p>
        <p class="offers-item-time">10:00 – 11:00 (ПН–ПТ)</p>
        <a class="offer-link" href="#">Записаться →</a>
      </li>
      <li class="offers-item">
        <h3>Лето в Лондоне</h3>
        <p class="offers-item-text">Месяц экскурсий и практики</p>
        <p class="offers-item-time">10 июня – 3 июля</p>
        <a class="offer-link" href="#">Отправиться →</a>
      </li>
    </ul>
  </section>

```

```

<li class="offers-item">
  <h3>Надоел английский?</h3>
  <p class="offers-item-text">С нами можно выучить и другие языки</p>
  <a class="offer-link" href="#">Открыть список →</a>
</li>
</ul>
</section>
<section class="video">
  <h2 class="visually-hidden">Просмотр видео уроков</h2>
  <video class="lesson-video" poster="img/image-18.jpg" width="540" height="300">
    <source src="video.mp4" type="video/mp4">
  </video>
  <h3>Курс English A1</h3>
  <ul class="play-list">
    <li class="play-item">
      <button class="play-button" type="button"><span
class="visually-hidden">Пуск/Стоп</span></button>
      <h4>Урок 8</h4>
      <p>Времена глаголов</p>
    </li>
    <li class="play-item play-item-active">
      <button class="play-button" type="button"><span
class="visually-hidden">Пуск/Стоп</span></button>
      <h4>Урок 9</h4>
      <p>Неправильные глаголы</p>
    </li>
    <li class="play-item">
      <button class="play-button" type="button"><span
class="visually-hidden">Пуск/Стоп</span></button>
      <h4>Урок 10</h4>
      <p>Present Simple</p>
    </li>
    <li class="play-item">
      <button class="play-button" type="button"><span
class="visually-hidden">Пуск/Стоп</span></button>
      <h4>Урок 11</h4>
      <p>Past Simple</p>
    </li>
  </ul>
</section>
<section class="contacts">
  <h2>Контакты</h2>
  <ul class="contact-list">
    <li class="contact-item">
      <a class="contact-link item-with-picture" data-text="H" href="#">
        <p class="contact-name">Неизвестный студент</p>
        <p class="contact-lang">English A1</p>
      </a>
    </li>
    <li class="contact-item">
      <a class="contact-link item-with-picture" data-text="H" href="#">
        <p class="contact-name"> Неизвестный студент 2</p>
        <p class="contact-lang">Español 2</p>
      </a>
    </li>
    <li class="contact-item">
      <a class="contact-link item-with-picture" data-text="H" href="#">
        <p class="contact-name"> Неизвестный студент 3</p>
        <p class="contact-lang">日本語</p>
      </a>
    </li>
  </ul>

```

```

<li class="contact-item">
  <a class="contact-link item-with-picture" data-text="H" href="#">
    <p class="contact-name"> Неизвестный студент 4</p>
    <p class="contact-lang">English B2</p>
  </a>
</li>
<li class="contact-item">
  <a class="contact-link item-with-picture" data-text="H" href="#">
    <p class="contact-name"> Неизвестный студент 5</p>
    <p class="contact-lang">Le Français</p>
  </a>
</li>
<li class="contact-item">
  <a class="contact-link item-with-picture" data-text="H" href="#">
    <p class="contact-name"> Неизвестный студент 6</p>
    <p class="contact-lang">Español 1</p>
  </a>
</li>
<li class="contact-item">
  <a class="contact-link item-with-picture" data-text="H" href="#">
    <p class="contact-name"> Неизвестный студент 7</p>
    <p class="contact-lang">Italiano</p>
  </a>
</li>
</ul>
</section>
</main>
<footer class="page-footer">
  <p class="copyright">© 2023</p>
  <a class="page-footer-logo" href="#"><b>Let's Speak</b>, языковая школа</a>
  <p class="copyright-details">Сайт создан в <a href="donstu.ru">МКМТ
ДГТУ</a></p>
</footer>
</body>
</html>

```

3	<p>Добавим разметку элементов страницы, в которую мы сразу включили основные классы. Подробно о том, как правильно создавать семантическую разметку читайте в статье «Создание семантической разметки по макету».</p>	<p>Изменения в index.html [https://codepen.io/Irina64/pen/oNMRREo]</p>
---	---	---

1. Добавим разметку элементов страницы, в которую мы сразу включили основные классы. Подробно о том, как правильно создавать семантическую разметку, читайте в статье «Создание семантической разметки по макету».
2. В заголовке страницы подключим стилевой файл [\[https://codepen.io/Irina64/pen/oNMRREo\]](https://codepen.io/Irina64/pen/oNMRREo). В нём определим общие стили страницы. Добавим CSS-правила для сброса внешних отступов у тега `<body>`. Укажем шрифт по умолчанию. Также установим для `<body>` CSS-свойство `min-width` — минимальную ширину, при которой сайт нормально смотрится. Все значения мы получили из макета.

3. Приступаем к построению сетки страницы. Определимся с типом страницы — это приложение. Поэтому нам нужен один общий контейнер-центровщик [\[https://codepen.io/Irina64/pen/oNMRREo\]](https://codepen.io/Irina64/pen/oNMRREo). Используем вариант с дополнительным классом `.container` для `<body>`. Ширину для контейнера-центровщика зададим равной ширине макета — 1440px. Поля для контейнера не устанавливаем.
4. Для наглядности подключим к странице вспомогательный стилевой файл [outline.css](#). С помощью него будем отслеживать поведение элементов в потоке. В `outline.css` определены цветные рамки для крупных блоков и элементов. При этом сразу выделился общий контейнер-центровщик.
5. Построим [каркас страницы или сетку для крупных блоков](#): навигации по сайту, шапки, основного содержимого и подвала. Каркас страницы очень похож на макет Holy Grail, но с одной боковой колонкой. Используем грид с двумя столбцами и тремя строками. Высота первой и третьей строк подстраиваются под содержимое. Средняя строка занимает всё свободное место. Первый столбец имеет фиксированную ширину — 280px, второй — занимает всё свободное пространство. Грид-контейнером сделаем тег `<body class="page">`. Чтобы сетка стала понятной, добавим именованные области, на которые будем ссылаться при определении CSS-свойства `grid-area` у грид-элементов. С помощью CSS-свойства `min-height: 100vh`; прижмем подвал сайта к низу страницы. Для шапки и подвала также определим CSS-свойство `min-height`, так мы обезопасим себя от «схлопывания» элементов при недостаточном количестве контента.
6. Переходим к [следующему уровню детализации](#): построим сетку для панели навигации. На панели расположены логотип, основное меню и меню настроек. В меню настроек сейчас одна ссылка, но это может измениться. Определим цвет фона и параметры границы для панели навигации, а также параметры шрифта для её элементов. О том, как получить нужные значения из макета, можно прочитать в методичке первого модуля «Экспорт параметров и графики из Figma».

```
outline.css

:root {
  --outline-1-level: rgb(76, 169, 255);
  --background-color-sticker-1-level: rgba(76, 169, 255, 0.1);
  --outline-2-level: rgb(255, 107, 0);
  --background-color-sticker-2-level: rgba(255, 107, 0, 0.1);
  --outline-3-level: rgb(116, 76, 230);
  --background-color-sticker-3-level: rgba(116, 76, 230, 0.1);
  --outline-4-level: rgb(78, 181, 67);
  --background-color-sticker-4-level: rgba(78, 181, 67, 0.2);
  --outline-5-level: rgb(239, 51, 255);
  --background-color-sticker-5-level: rgba(239, 51, 255, 0.2);
  --outline-6-level: rgb(255, 191, 0);
  --background-color-sticker-6-level: rgba(255, 191, 0, 0.2);
}

body,
body > *,
```

```

body > * > *,
body > * > * > *,
body > * > * > * > *,
body > * > * > * > * > *,
body > * > * > * > * > * > * {
  outline-offset: -2px;
  position: relative;
}

body::before,
body > *::before,
body > * > *::before,
body > * > * > *::before,
body > * > * > * > *::before,
body > * > * > * > * > *::before {
  position: absolute;
  top: 0;
  left: 0;
  padding: 0 10px 2px;
  font-family: "Courier New", monospace;
  font-weight: bold;
  font-size: 14px;
  line-height: 18px;
  letter-spacing: initial;
  text-transform: lowercase;
  color: white;
  white-space: nowrap;
  z-index: 1;
}

.page-menu {
  outline: 2px solid var(--outline-2-level);
  background: var(--background-color-sticker-2-level);
}

.page-menu::before {
  content: attr(class);
  background: var(--outline-2-level);
}

```

7. Чтобы выстроить элементы навигации по нужной сетке, используем грид. На этот раз грид-контейнером сделаем элемент `<nav class="page-menu">`. Зададим шаблон строк для грид-контейнера. Строка с логотипом имеет фиксированную высоту — 80px, меню с настройками должно поместиться, а основному меню выделим всё оставшееся пространство.

```

style.css //(изменения)
.page-menu {
  grid-area: navigation;
  display: grid;
  grid-template-rows:
    80px

```



```

    1fr
    auto;
    background-color: #ffffff;
    border-right: 1px solid #b4bef0;
}

outline.css //(изменения)
.page-menu::before {
    content: attr(class);
    background: var(--outline-2-level);
}

//(добавлено)
.page-menu > * {
    outline: 2px solid var(--outline-3-level);
    background: var(--background-color-sticker-3-level);
}

.page-menu > *::before {
    content: attr(class);
    left: auto;
    right: 0;
    background: var(--outline-3-level);
}

```

8. Добавим внешний отступ слева для логотипа. Чтобы выровнять логотип по вертикали, используем индивидуальное выравнивание для грид-элемента относительно вертикальной оси `align-self: center`.

```

style.css //(изменения)
.page-footer {
    grid-area: footer;
    min-height: 60px;
}
// добавлено
.logo {
    margin-left: 30px;
    align-self: center;
}

.page-menu-item-active {
    background-color: #f1f3fe;
}

```

9. Построим сетку для элементов основного меню. Для списка сбросим стили по умолчанию. Ссылки сделаем блочными элементами, чтобы указать для них внутренние отступы.

```

style.css //(изменения)
....
// добавлено
.page-menu-list {
    padding: 0;
    margin: 0;
}

```



```

list-style: none;
}

.page-menu-item-active {
background-color: #f1f3fe;
}

.page-menu-link {
display: block;
padding: 20px 25px 20px 63px;
font-size: 16px;
line-height: 20px;
color: #3c425f;
text-decoration: none;
}

```

10. Аналогичные действия выполним для списка с настройками.

```

style.css //(изменения)
....
// добавляем в конце файла
.page-menu-tools {
padding: 0;
margin: 0 0 10px 0;
list-style: none;
}

```

11. Переходим к шапке сайта. Она состоит из формы поиска, ссылки на профиль пользователя с кнопкой управления пользовательским меню и кнопки для отображения уведомлений. В этом блоке нам понадобятся стили для доступного скрытия элементов. Подробную информацию о том, как правильно прятать, можно найти в [блоге HtmlАкадемии](#).

```

style.css //(изменения)
....
// добавляем

.visually-hidden {
position: absolute;
clip: rect(1px, 1px, 1px, 1px);
padding: 0;
border: 0;
height: 1px;
width: 1px;
overflow: hidden;
}

outline.css //(изменения)
:root {
--outline-1-level: rgb(76, 169, 255);
--background-color-sticker-1-level: rgba(76, 169, 255, 0.1);
--outline-2-level: rgb(255, 107, 0);
--background-color-sticker-2-level: rgba(255, 107, 0, 0.1);
--outline-3-level: rgb(116, 76, 230);
--background-color-sticker-3-level: rgba(116, 76, 230, 0.1);
}

```

```

--outline-4-level: rgb(78, 181, 67);
--background-color-sticker-4-level: rgba(78, 181, 67, 0.2);
--outline-5-level: rgb(239, 51, 255);
--background-color-sticker-5-level: rgba(239, 51, 255, 0.2);
--outline-6-level: rgb(255, 191, 0);
--background-color-sticker-6-level: rgba(255, 191, 0, 0.2);
}
body,
body > *,
body > * > *,
body > * > * > *,
body > * > * > * > *,
body > * > * > * > * > *,
body > * > * > * > * > * > * {
  outline-offset: -2px;
  position: relative;
}
body::before,
body > *::before,
body > * > *::before,
body > * > * > *::before,
body > * > * > * > *::before,
body > * > * > * > * > *::before,
body > * > * > * > * > * > *::before {
  position: absolute;
  top: 0;
  left: 0;
  padding: 0 10px 2px;
  font-family: "Courier New", monospace;
  font-weight: bold;
  font-size: 14px;
  line-height: 18px;
  letter-spacing: initial;
  text-transform: lowercase;
  color: white;
  white-space: nowrap;
  z-index: 1;
}
.page-header {
  outline: 2px solid var(--outline-2-level);
  background: var(--background-color-sticker-2-level);
}
.page-header::before {
  content: attr(class);
  top: auto;
  bottom: 0;
  background: var(--outline-2-level);
}

```

12. Определим цвет фона, размеры, параметры границ и шрифта для элементов блока.

```

.page-header {
  grid-area: header;
  min-height: 80px;
  background-color: #ffffff;
  border-bottom: 1px solid #b4bef0;
}

```

```

}
.....
// добавляем в конец файла style.css

.search-field {
  font-family: inherit;
  font-weight: 300;
  font-size: 16px;
  line-height: 19px;
  color: #3c425f;
  border: none;
}
.search-field::placeholder {
  color: #b4bef0;
}
.profile-name {
  font-size: 16px;
  line-height: 40px;
  color: #3c425f;
  text-decoration: none;
}
.notifications {
  width: 25px;
  height: 25px;
}
.notifications span {
  font-weight: 900;
  font-size: 10px;
  line-height: 14px;
  color: #ffffff;
  background-color: #ff8f0b;
  border-radius: 50%;
  border: 2px solid #ffffff;
}

```

13. Построим сетку для шапки сайта. Добавим внутренние отступы. Используем грид, чтобы выстроить элементы в ряд. Колонка с уведомлениями и колонка для элемента с ссылкой на профиль пользователя соответствуют ширине размещаемых элементов, а для поля с поиском используем всё оставшееся место. Добавим отступы между столбцами. Это значение примерно соответствует расстоянию между соседними элементами в макете. За счёт того, что первая колонка займёт всё свободное пространство, две последние колонки сетки будут «прибиты» к правому краю.

Добавляем в style.css

```

.....
.page-header {
  grid-area: header;
  min-height: 80px;
  display: grid;
  grid-template-columns: 1fr auto auto;
  grid-column-gap: 45px;
  padding: 0 30px;
}

```

```
background-color: #ffffff;
border-bottom: 1px solid #b4bef0;
}
```

.....

14. Определим выравнивание относительно вертикальной оси для элементов шапки. Расположим их по центру align-items: center;.

```
.page-header {
  grid-area: header;
  min-height: 80px;
  display: grid;
  grid-template-columns: 1fr auto auto;
  grid-column-gap: 45px;
  align-items: center;
  padding: 0 30px;
  background-color: #ffffff;
  border-bottom: 1px solid #b4bef0;
}
```

15. Построим сетку для элемента с ссылкой на профиль пользователя и кнопкой управления пользовательским меню. С помощью сетки на гридах выстроим содержимое элемента <div class="profile"> в две колонки (имя и кнопка). Ширина колонок подстраивается под их содержимое. Отступ между колонками 10px. Декоративный элемент с инициалами пользователя добавим с помощью псевдоэлемента для ссылки . Ссылку мы тоже преобразуем в грид-контейнер, который состоит из двух колонок, первая шириной 40px, вторая подстраивается под содержимое grid-template-columns: 40px auto;.

16. Переходим к блоку с основным содержимым страницы. Построим каркас сетки для блока. Зададим внутренний отступ 30px. Внутренние блоки: слайдер, предложения, видео и список контактов выстроены по сетке. Сетка состоит из двух колонок и трёх строк, причём первая колонка в три раза шире, чем вторая, поэтому можно первой колонке указать ширину 3fr, а второй — 1fr. Высота первой строки зафиксирована и равна 300px — по высоте слайдера. Высота второй строки подстраивается под содержимое. Третья строка займёт всё оставшееся место. Между ячейками есть грид-интервал в 20px. Блок со слайдером занимает в этой сетке первую строку и первую колонку, предложения клуба — вторую строку и первую колонку, блок с видео и плейлистом третью строку и первую колонку, во второй колонке размещается раздел с контактами, он занимает три строки. Для определения сетки также воспользуемся именованными областями.

Добавляем в style.css

```
... // изменения
.page-main {
  grid-area: main;
  padding: 30px;
  display: grid;
```

```

    grid-template-columns: 3fr 1fr;
    grid-template-rows: 300px auto 1fr;
    grid-template-areas:
        "slider contacts"
        "offers contacts"
        "video contacts";
    grid-gap: 20px;
}
...// в конце файла

.slider {
    grid-area: slider;
}

.contacts {
    grid-area: contacts;
}

.offers {
    grid-area: offers;
}

.video {
    grid-area: video;
}

```

17. Определим стили для слайдера. Начнем с параметров шрифта, цвета фона и параметров границы.

```

.slider {
    grid-area: slider;
    font-weight: 300;
    font-size: 18px;
    line-height: 28px;
}

.slider-headline {
    font-weight: 900;
    font-size: 28px;
    line-height: 48px;
    color: #ffffff;
}

.slider-text {
    color: #ffffff;
}

.slider a {
    font-weight: 500;
    font-size: 18px;
    line-height: 22px;
    color: #ffffff;
    text-decoration: none;
    background-color: #ff8f0b;
    border-radius: 4px;
}

.slider-btn {
    width: 64px;
    height: 64px;
}

```

```

border-radius: 50%;
border: none;
background-color: #ffffff;
background-image: url("img/arrow.svg");
background-repeat: no-repeat;
background-position: left 35px center;
background-size: 18px 18px;
}

```

```

.slider-btn-next {
  transform: rotate(180deg);
}

```

```

.contacts {
  grid-area: contacts;
}

```

```

.offers {
  grid-area: offers;
}

```

```

.video {
  grid-area: video;
}

```

outline.css //(изменения)

```

:root {
  --outline-1-level: rgb(76, 169, 255);
  --background-color-sticker-1-level: rgba(76, 169, 255, 0.1);
  --outline-2-level: rgb(255, 107, 0);
  --background-color-sticker-2-level: rgba(255, 107, 0, 0.1);
  --outline-3-level: rgb(116, 76, 230);
  --background-color-sticker-3-level: rgba(116, 76, 230, 0.1);
  --outline-4-level: rgb(78, 181, 67);
  --background-color-sticker-4-level: rgba(78, 181, 67, 0.2);
  --outline-5-level: rgb(239, 51, 255);
  --background-color-sticker-5-level: rgba(239, 51, 255, 0.2);
  --outline-6-level: rgb(255, 191, 0);
  --background-color-sticker-6-level: rgba(255, 191, 0, 0.2);
}

```

```

body,
body > *,
body > * > *,
body > * > * > *,
body > * > * > * > *,
body > * > * > * > * > *,
body > * > * > * > * > * > * {
  outline-offset: -2px;
  position: relative;
}
body::before,
body > *::before,
body > * > *:not(.item-with-picture)::before,
body > * > * > *:not(.item-with-picture)::before,
body > * > * > * > *:not(.item-with-picture)::before,

```

```

body > * > * > * > * > *:not(.item-with-picture)::before,
body > * > * > * > * > * > *:not(.item-with-picture)::before {
  position: absolute;
  top: 0;
  left: 0;
  padding: 0 10px 2px;
  font-family: "Courier New", monospace;
  font-weight: bold;
  font-size: 14px;
  line-height: 18px;
  letter-spacing: initial;
  text-transform: lowercase;
  color: white;
  white-space: nowrap;
  z-index: 1;
}

.slider {
  outline: 2px solid var(--outline-3-level);
  background: var(--background-color-sticker-3-level);
}

.slider::before {
  content: attr(class);
  background: var(--outline-3-level);
}

.slider > * {
  outline: 2px solid var(--outline-4-level);
  background: var(--background-color-sticker-4-level);
}

.slider > *::before {
  background: var(--outline-4-level);
}

```

18. Многослойность реализуем с помощью CSS-свойства `position: absolute`; (подробнее о позиционировании читайте в интерактивных курсах). Изображение и кнопки управления слайдером «вырываем» из потока. Для остальных элементов задаём сеточные стили. Для того, чтобы ширина слайдера учитывала внутренние отступы, CSS-свойство `box-sizing` зададим равным `border-box`. За счёт CSS-свойства `overflow: hidden`; отображаем только область внутри элемента, остальное будет скрыто.

Добавляем в `style.css`

```

body{}
...
.notifications span {
... // изменения

.slider {
  position: relative;
  grid-area: slider;

```



```

width: 820px;
height: 300px;
padding: 32px 60px;
overflow: hidden;
box-sizing: border-box;
font-weight: 300;
font-size: 18px;
line-height: 28px;
}

.slider-img {
position: absolute;
left: 0;
top: 0;
}

.slider-headline {
position: relative;
margin-bottom: 10px;
font-weight: 900;
font-size: 28px;
line-height: 48px;
color: #ffffff;
}

.slider-text {
position: relative;
margin: 0 0 30px 0;
max-width: 470px;
color: #ffffff;
}

.slider a {
position: relative;
padding: 10px 20px;
font-weight: 500;
font-size: 18px;
line-height: 22px;
color: #ffffff;
text-decoration: none;
background-color: #ff8f0b;
border-radius: 4px;
}

.slider-btn {
position: absolute;
width: 64px;
height: 64px;
left: -32px;
top: 0;
bottom: 0;
margin: auto;
border-radius: 50%;
border: none;
background-color: #ffffff;
background-image: url("img/arrow.svg");
background-repeat: no-repeat;
background-position: left 35px center;
}

```

```

    background-size: 18px 18px;
}

.slider-btn-next {
    left: auto;
    right: -32px;
    top: 0;
    bottom: 0;
    margin: auto;
    transform: rotate(180deg);
}

```

19. Переходим к разделу с предложениями. Определим цвет фона, параметры эффектов, границы и шрифта для элементов.

Добавляем в style.css
// после offers

```

.offers-item {
    background-color: #ffffff;
    box-shadow: 0 1px 4px rgba(60, 66, 95, 0.2);
    border-radius: 4px;
}

.offers-item h3 {
    font-weight: 500;
    font-size: 18px;
    line-height: 22px;
    color: #3c425f;
}

.offers-item-text {
    font-weight: 300;
    font-size: 14px;
    line-height: 20px;
    color: #3c425f;
}

.offers-item-time {
    font-weight: 900;
    font-size: 14px;
    line-height: 20px;
    color: #0bc4cf;
}

.offer-link {
    font-weight: 500;
    font-size: 16px;
    line-height: 20px;
    color: #576bcc;
    text-decoration: none;
}

.video {

```

```

    grid-area: video;
}

outline.css //(изменения)
:root {

}
...

body::before,
body > *::before,
body > * > *:not(.item-with-picture)::before,
body > * > * > *:not(.item-with-picture)::before,
body > * > * > * > *::before,
body > * > * > * > * > *:not(.item-with-picture)::before,
body > * > * > * > * > * > *:not(.item-with-picture)::before {
    position: absolute;
    top: 0;
    left: 0;
    padding: 0 10px 2px;
    font-family: "Courier New", monospace;
    font-weight: bold;
    font-size: 14px;
    line-height: 18px;
    letter-spacing: initial;
    text-transform: lowercase;
    color: white;
    white-space: nowrap;
    z-index: 1;
}

.offers {
    background-color: var(--background-color-sticker-3-level);
    outline: 2px solid var(--outline-3-level);
}

.offers::before {
    content: attr(class);
    background: var(--outline-3-level);
}

.offers > * {
    background-color: var(--background-color-sticker-4-level);
    outline: 2px solid var(--outline-4-level);
}

.offers > *::before {
    content: attr(class);
    background: var(--outline-4-level);
}

```

20. Переходим к сеточным стилям. Для списка сбросим стили по умолчанию. Сделаем список грид-контейнером. В ряду будут размещаться три колонки одинаковой ширины. В данном случае мы могли бы

использовать флекс-контейнер, но у грид-контейнера есть преимущество — не нужно сбрасывать отступ справа для каждого третьего элемента списка.

Добавляем в `style.css`

// после `offers`

```
.offers-list {
  padding: 0;
  margin: 0;
  list-style: none;
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-gap: 20px;
}
```

`outline.css` //(изменения)

...

```
body::before,
body > *::before,
body > * > *::before,
body > * > * > *:not(.item-with-picture)::before,
body > * > * > * > *::before,
body > * > * > * > * > *:not(.item-with-picture)::before,
body > * > * > * > * > * > *:not(.item-with-picture)::before {
  position: absolute;
  top: 0;
  left: 0;
  padding: 0 10px 2px;
  font-family: "Courier New", monospace;
  font-weight: bold;
  font-size: 14px;
  line-height: 18px;
  letter-spacing: initial;
  text-transform: lowercase;
  color: white;
  white-space: nowrap;
  z-index: 1;
}
```

```
.offers {
  background-color: var(--background-color-sticker-3-level);
  outline: 2px solid var(--outline-3-level);
}
```

```
.offers::before {
  content: attr(class);
  right: 0;
  left: auto;
  background: var(--outline-3-level);
}
```

```
.offers > * {
  background-color: var(--background-color-sticker-4-level);
  outline: 2px solid var(--outline-4-level);
}
```

```

}

.offers > *::before {
  content: attr(class);
  background: var(--outline-4-level);
}

.offers > * > * {
  background-color: var(--background-color-sticker-5-level);
  outline: 2px solid var(--outline-5-level);
}

.offers > * > *::before {
  content: attr(class);
  top: auto;
  bottom: 0;
  background: var(--outline-5-level);
}

```

21. Определим отступы для элементов карточки.
Добавляем в style.css после offers

```

.offers-item {
  padding: 20px 30px;
  background-color: #ffffff;
  box-shadow: 0 1px 4px rgba(60, 66, 95, 0.2);
  border-radius: 4px;
}

.offers-item h3 {
  margin: 0;
  font-weight: 500;
  font-size: 18px;
  line-height: 22px;
  color: #3c425f;
}

.offers-item-text {
  margin: 5px 0 5px 0;
  font-weight: 300;
  font-size: 14px;
  line-height: 20px;
  color: #3c425f;
}

.offers-item-time {
  margin: 0 0 8px 0;
  font-weight: 900;
  font-size: 14px;
  line-height: 20px;
  color: #0bc4cf;
}

```

22. Ссылка в карточке предложения выравнивается по нижнему краю.
Такое поведение можно реализовать с помощью вертикального грида на три

строки, с центральной частью, занимающей всю возможную высоту (1fr). Для первой и третьей строк зададим автоматическую высоту.

```
.offers-item {
  display: grid;
  grid-template-rows:
    auto
    1fr
    auto;
  padding: 20px 30px;
  background-color: #ffffff;
  box-shadow: 0 1px 4px rgba(60, 66, 95, 0.2);
  border-radius: 4px;
}
```

23. Добавим ссылке на подробную информацию индивидуальное выравнивание относительно горизонтальной оси.

```
.offer-link {
  justify-self: end;
  font-weight: 500;
  font-size: 16px;
  line-height: 20px;
  color: #576bcc;
  text-decoration: none;
}
```

24. Переходим к разделу для просмотра видеоуроков. Раздел состоит из двух колонок: область для просмотра видео и плейлист. Нам понадобится [дополнительная обёртка](#) `<div class="video-play-list-wrapper">`.

outline.css //(изменения)

...

```
.video {
  outline: 2px solid var(--outline-3-level);
  background: var(--background-color-sticker-3-level);
}

.video::before {
  content: attr(class);
  background: var(--outline-3-level);
}

.video > * {
  outline: 2px solid var(--outline-4-level);
  background: var(--background-color-sticker-4-level);
}

.video > *::before {
  content: attr(class);
  right: 0;
  left: auto !important;
  background: var(--outline-4-level);
}
```

25. Для создания сетки раздела используем грид. Первая колонка в два раза шире второй. Зададим грид-интервал.

```
.video {  
  grid-area: video;  
  display: grid;  
  grid-template-columns: 2fr 1fr;  
  grid-gap: 20px;  
}
```

outline.css //(изменения)

line-height: 20px; (вместо 18px)

26. Определим цвет фона, параметры границы и шрифта для элементов блока.

Добавляем в style.css
// после video{}

```
.video h3 {  
  border-bottom: 1px solid #e4e8fc;  
  font-weight: 500;  
  font-size: 18px;  
  line-height: 22px;  
}  
  
.video-play-list-wrapper {  
  background: #ffffff;  
  box-shadow: 0 1px 4px rgba(60, 66, 95, 0.2);  
  border-radius: 4px;  
}  
  
.play-item {  
  font-weight: 300;  
  font-size: 14px;  
  line-height: 16px;  
  color: #3c425f;  
}  
  
.play-button {  
  width: 32px;  
  height: 32px;  
  background-color: #f1f3fe;  
  border-radius: 4px;  
  border: none;  
}  
  
.play-item h4 {  
  font-weight: 900;  
}  
  
.play-item-active {  
  background-color: #f1f3fe;  
}
```



```
.play-item-active .play-button {  
  background-color: #576bcc;  
}
```

27. Сбросим стили по умолчанию для списка. Укажем отступы для элементов.

```
.video h3 {  
  margin: 20px 0 0 30px;  
  padding-bottom: 10px;  
  ...  
}
```

....

```
.play-list {  
  padding: 0;  
  margin: 0;  
  list-style: none;  
}
```

```
.play-item {  
  padding: 8px 10px 8px 30px;  
  font-weight: 300;  
  font-size: 14px;  
  line-height: 16px;  
  color: #3c425f;  
}
```

```
.play-button {  
  width: 32px;  
  height: 32px;  
  background-color: #f1f3fe;  
  border-radius: 4px;  
  border: none;  
}
```

```
.play-item h4 {  
  margin: 0;  
  font-weight: 900;  
}
```

```
.play-item p {  
  margin: 0;  
}
```

```
.play-item-active {  
  background-color: #f1f3fe;  
}
```

```
.play-item-active .play-button {  
  background-color: #576bcc;  
}
```

outline.css //(изменения)

```
:root {
  --outline-1-level: rgb(76, 169, 255);
  --background-color-sticker-1-level: rgba(76, 169, 255, 0.1);
  --outline-2-level: rgb(255, 107, 0);
  --background-color-sticker-2-level: rgba(255, 107, 0, 0.1);
  --outline-3-level: rgb(116, 76, 230);
  --background-color-sticker-3-level: rgba(116, 76, 230, 0.1);
  --outline-4-level: rgb(78, 181, 67);
  --background-color-sticker-4-level: rgba(78, 181, 67, 0.2);
  --outline-5-level: rgb(239, 51, 255);
  --background-color-sticker-5-level: rgba(239, 51, 255, 0.2);
  --outline-6-level: rgb(255, 191, 0);
  --background-color-sticker-6-level: rgba(255, 191, 0, 0.2);
}
```

```
body,
body > *,
body > * > *,
body > * > * > *,
body > * > * > * > *,
body > * > * > * > * > *,
body > * > * > * > * > * > * {
  outline-offset: -2px;
  position: relative;
}
```

```
body::before,
body > *::before,
body > * > *::before,
body > * > * > *:not(.item-with-picture)::before,
body > * > * > * > *::before,
body > * > * > * > * > *:not(.item-with-picture)::before,
body > * > * > * > * > * > *:not(.item-with-picture)::before {
  position: absolute;
  top: 0;
  left: 0;
  padding: 0 10px 2px;
  font-family: "Courier New", monospace;
  font-weight: bold;
  font-size: 14px;
  line-height: 18px;
  letter-spacing: initial;
  text-transform: lowercase;
  color: white;
  white-space: nowrap;
  z-index: 1;
}
```

```
.video {
  outline: 2px solid var(--outline-3-level);
  background: var(--background-color-sticker-3-level);
}
```

```

.video::before {
  content: attr(class);
  background: var(--outline-3-level);
}

.video > * {
  outline: 2px solid var(--outline-4-level);
  background: var(--background-color-sticker-4-level);
}

.video > *::before {
  content: attr(class);
  right: 0;
  left: auto !important;
  background: var(--outline-4-level);
}

.video > * > * {
  outline: 2px solid var(--outline-5-level);
  background: var(--background-color-sticker-5-level);
}

.video > * > *::before {
  content: attr(class);
  background: var(--outline-5-level);
}

.video > * > * > * {
  background-color: var(--background-color-sticker-6-level) !important;
  outline: 2px solid var(--outline-6-level) !important;
}

.video > * > * > *::before {
  content: attr(class);
  top: auto !important;
  bottom: 0;
  background: var(--outline-6-level);
}

```

28. Ограничим список по высоте. А с помощью CSS-свойства `overflow-y: auto`; определим, что вертикальная полоса прокрутки будет появляться только при необходимости.

```

.play-list {
  padding: 0;
  margin: 0;
  list-style: none;
  height: 235px;
  overflow-y: auto;
}

```

29. Ограничим по ширине и высоте видеопроигрыватель.

```

.video {

}

.lesson-video {

```

```

width: 540px;
height: 300px;
}

.video h3 {
}

```

30. Построим сетку для элементов списка с видеоуроками. Каждый элемент списка станет гридом из двух колонок и двух строк. Ширина первой колонки 32px — по ширине кнопки запуска видео, второй — всё оставшееся пространство. Добавим отступ между колонками 13px. Кнопка запуска видео занимает две строки в первой колонке грид-контейнера, номер и название урока займут по одной строке во второй. Для наглядности используем именованные грид-области.

```

.play-item {
  display: grid;
  grid-template-columns: 32px 1fr;
  grid-template-rows: auto auto;
  grid-template-areas:
    "button title"
    "button desc";
  grid-column-gap: 13px;
  padding: 8px 10px 8px 30px;
  font-weight: 300;
  font-size: 14px;
  line-height: 16px;
  color: #3c425f;
}

.play-button {
  grid-area: button;
  width: 32px;
  height: 32px;
  background-color: #f1f3fe;
  border-radius: 4px;
  border: none;
}

.play-item h4 {
  grid-area: title;
  margin: 0;
  font-weight: 900;
}

.play-item p {
  grid-area: desc;
  margin: 0;
}

```

31. Стилизуем раздел с контактами. Традиционно сначала определяем цвет фона, параметры границ и параметры шрифта для элементов раздела.

```
.contacts {
  grid-area: contacts;
  background-color: #ffffff;
  box-shadow: 0 1px 4px rgba(60, 66, 95, 0.2);
  border-radius: 4px;
}
```

32. Определим отступы для элементов. Для списка сбросим стили по умолчанию.

```
...
.contacts {
  grid-area: contacts;
  padding: 20px 0 20px 30px;
  background-color: #ffffff;
  box-shadow: 0 1px 4px rgba(60, 66, 95, 0.2);
  border-radius: 4px;
}
```

.....

```
.contacts h2 {
  margin: 0 0 10px 0;
  font-weight: 500;
  font-size: 18px;
  line-height: 22px;
  color: #3c425f;
}

.contact-list {
  padding: 5px 0 0 0;
  margin: 0;
  list-style: none;
  border-top: 1px solid #e4e8fc;
}

.contact-item {
  padding: 16px 10px 0 0;
}

.contact-link {
  text-decoration: none;
  font-weight: 300;
  font-size: 14px;
  line-height: 20px;
  color: #3c425f;
}
```

33. Построим сетку для каждого отдельного элемента в списке контактов. Чтобы инициалы пользователя разместить напротив его имени и изучаемого курса, ссылку вокруг перечисленных элементов сделаем грид-контейнером из двух колонок и двух строк. Первой колонке укажем фиксированную ширину 40px — по размеру элемента с инициалами, вторая занимает оставшееся пространство. Укажем расстояние между колонками грид-контейнера.

Определим положение каждого грид-элемента. Инициалы займут всю первую колонку. Имя пользователя разместим во второй колонке и первой строке. Изучаемый курс — во второй колонке и второй строке.

```
.contact-list {
.....
}

.contact-item {
...
}

.contact-link {
  display: grid;
  grid-template-columns: 40px 1fr;
  grid-template-rows:
    auto
    auto;
  grid-column-gap: 12px;
  text-decoration: none;
  font-weight: 300;
  font-size: 14px;
  line-height: 20px;
  color: #3c425f;
}

.contact-link::before {
  grid-row: 1 / 3;
}

.contact-name {
  grid-column: 2 / 3;
  grid-row: 1 / 2;
  margin: 0;
}

.contact-lang {
  grid-column: 2 / 3;
  grid-row: 2 / 3;
  margin: 0;
}
```

34. Переходим к подвалу сайта. Определим цвет фона и параметры шрифта для элементов.

```
....

.page-footer {
  grid-area: footer;
  min-height: 60px;
  background-color: #3c425f;
}

.....
// в конце style.css
```

```

    .copyright {
      font-weight: 300;
      font-size: 16px;
      line-height: 20px;
      color: #aaaaaa;
    }

    .page-footer-logo {
      font-weight: 300;
      font-size: 16px;
      line-height: 20px;
      color: #aaaaaa;
      text-decoration: none;
    }

    .copyright-details {
      font-weight: 300;
      font-size: 16px;
      line-height: 20px;
      color: #aaaaaa;
    }

    .copyright-details a {
      color: #aaaaaa;
      text-decoration: none;
    }

```

35. Укажем отступы для подвала.

```

.page-footer {
  grid-area: footer;
  padding: 0 30px;
  min-height: 60px;
  background-color: #3c425f;
}

```

.....

36. Построим сетку для подвала сайта. Используем грид с тремя колонками. Ширина первой и третьей колонок равна 280px. Центральная колонка занимает всё свободное место. На случай переполнения добавим отступы между колонками.

```

.page-footer {
  grid-area: footer;
  display: grid;
  grid-template-columns: 280px 1fr 280px;
  grid-column-gap: 20px;
  padding: 0 30px;
  min-height: 60px;
  background-color: #3c425f;
}

```


}

37. Добавим выравнивание грид-элементам в подвале сайта. Выравнивание относительно горизонтальной и вертикальной оси установим равным center. Для `<p class="copyright">` определим индивидуальное выравнивание вдоль строчной оси `justify-self: start;`, а для `<p class="copyright-details">` — `justify-self: end;`.

```
.page-footer {
  grid-area: footer;
  display: grid;
  grid-template-columns: 280px 1fr 280px;
  grid-column-gap: 20px;
  align-items: center;
  justify-items: center;
  padding: 0 30px;
  min-height: 60px;
  background-color: #3c425f;
}
.....
.copyright-details {
  justify-self: end;
  font-weight: 300;
  font-size: 16px;
  line-height: 20px;
  color: #aaaaaa;
}

.copyright-details a {
  color: #aaaaaa;
  text-decoration: none;
}
```

38. Отключаем [вспомогательные стили](#). Сетка страницы готова! Осталось добавить недостающие декоративные элементы. Сделаем это на следующем шаге.

39. [Все готово!](#)

!!! Можете попробовать сделать разметку на других макетах:

1. [Лендинг спектакля](#)
2. [Макет статьи в блоге о путешествиях «Traveler's Blog»](#) (модная статья о путешествиях своим ходом с интересным расположением иллюстраций)
3. [макет каталога магазина цветов «Floral Boutique»](#) (каталог с небольшим выбором)
4. [макет каталога на сайте бренда женской одежды «Furly»](#) (монобрендовый интернет-магазин)
5. Любой макет, схожий по тематике с вышеуказанными макетами.