
| | |
|------------------------|---|
| ДПО | Фронтенд и бэкенд разработка |
| ДИСЦИПЛИНА | Основы фронтенд-разработки |
| ВИД УЧЕБНОГО МАТЕРИАЛА | Методические указания к практическим занятиям |

Практическое занятие 4: Адаптивная вёрстка с использованием Flexbox, Grid и медиазапросов

Цель: освоить применение Flexbox для одномерного, Grid для двумерного размещения элементов и медиазапросов для создания адаптивных интерфейсов. Закрепить навыки работы с использованием методологий.

Введение и теоретическая база

Перед выполнением практики рекомендуется ознакомиться с материалами Лекции 4:

- **Flexbox** - слайды 26-50 (основные свойства, оси, выравнивание)
 - **Grid** - слайды 51-70 (сетка, треки, размещение элементов)
 - **Masonry** - слайды 71-77 (нестандартная сетка с автоматическим заполнением)
 - **Медиазапросы** - слайды 78-90 (адаптивность, breakpoints, подходы)
-

Последовательность выполнения работы

Шаг 1: Подготовка проекта

1. Продолжайте работу в том же репозитории, что и в Практическом занятии 3.
 2. Убедитесь, что у вас есть структурированный HTML и CSS, соответствующий выбранной методологии.
 3. Для работы с Flexbox, Grid, Masonry-раскладкой и медиазапросами используйте существующий макет. Вы можете создавать новые разделы или модифицировать существующие элементы.
-

Шаг 2: Применение Flexbox для горизонтального меню

Используйте Flexbox для создания адаптивного навигационного меню. В шаге 5 предусмотрите изменение положения пунктов меню для мобильной версии.

Пример реализации:

```
<nav class="nav">
    <ul class="nav_list">
        <li class="nav_item">
            <a href="#" class="nav_link">Главная</a>
        </li>
        <li class="nav_item">
```

```
<a href="#" class="nav__link">0 проекте</a>
</li>
<li class="nav__item">
    <a href="#" class="nav__link">Услуги</a>
</li>
<li class="nav__item">
    <a href="#" class="nav__link">Контакты</a>
</li>
</ul>
</nav>
```

```
.nav__list {
    display: flex;
    justify-content: center;
    gap: 2rem;
    list-style: none;
    padding: 0;
    margin: 0;
}

.nav__link {
    text-decoration: none;
    color: #333;
    padding: 0.5rem 1rem;
    border-radius: 4px;
    transition: all 0.3s ease;
}

.nav__link:hover {
    background-color: #f0f0f0;
    color: #2c5530;
}
```

Шаг 3: Создание Grid-сетки для контентных блоков

Реализуйте адаптивную сетку для размещения контентных блоков. Можете использовать Grid для размещения карточек с преимуществами / особенностями производителя. В обоих макетах они расположены сразу после шапки сайта.

Пример реализации сетки с Grid:

```
<section class="content-grid">
    <h2 class="content-grid__title">Наши направления</h2>
    <div class="content-grid__container">
        <article class="content-card">
            <h3 class="content-card__title">Направление 1</h3>
            <p class="content-card__text">Описание первого направления деятельности.</p>
```

```
</article>
<article class="content-card">
    <h3 class="content-card__title">Направление 2</h3>
    <p class="content-card__text">Описание второго направления деятельности.</p>
</article>
<article class="content-card">
    <h3 class="content-card__title">Направление 3</h3>
    <p class="content-card__text">Описание третьего направления деятельности.</p>
</article>
<article class="content-card">
    <h3 class="content-card__title">Направление 4</h3>
    <p class="content-card__text">Описание четвёртого направления деятельности.</p>
</article>
</div>
</section>
```

```
.content-grid__container {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 1.5rem;
    margin-top: 2rem;
}

.content-card {
    padding: 1.5rem;
    background: #f8f9fa;
    border-radius: 8px;
    border: 1px solid #e9ecf;
    transition: transform 0.3s ease;
}

.content-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}
```

Шаг 4: Реализация Masonry-раскладки (!данний пункт выполняется по желанию)

Создайте нестандартную сетку с элементами разной высоты. Примеры реализации представлены в Лекции 4.

Один из форматов также продублирован ниже:

```
<section class="masonry-grid">
    <h2 class="masonry-grid__title">Дополнительные материалы</h2>
```

```
<div class="masonry-grid__container">
    <article class="masonry-item masonry-item--short">
        <h3 class="masonry-item__title">Краткий материал</h3>
        <p class="masonry-item__text">Короткое описание.</p>
    </article>
    <article class="masonry-item masonry-item--tall">
        <h3 class="masonry-item__title">Подробный материал</h3>
        <p class="masonry-item__text">Развёрнутое описание с дополнительной
информацией и деталями реализации.</p>
    </article>
    <article class="masonry-item">
        <h3 class="masonry-item__title">Стандартный материал</h3>
        <p class="masonry-item__text">Обычное описание материала средней
длины.</p>
    </article>
    <article class="masonry-item masonry-item--very-tall">
        <h3 class="masonry-item__title">Обширный материал</h3>
        <p class="masonry-item__text">Полное описание со всеми техническими
характеристиками, особенностями применения и рекомендациями по использованию в
различных условиях.</p>
    </article>
    <!-- Сюда можно добавить сколь угодно элементов разной высоты -->
</div>
</section>
```

```
.masonry-grid__container {
    column-count: 3;
    column-gap: 24px;
}

.masonry-item {
    display: inline-block;
    margin: 0 0 1em;
    padding: 1.5rem;
    background: #f8f9fa;
    border-radius: 8px;
    border: 1px solid #e9ecf;
    transition: transform 0.3s ease;
}

.masonry-item--short {
    grid-row-end: span 15;
}

.masonry-item--tall {
    grid-row-end: span 30;
}

.masonry-item--very-tall {
    grid-row-end: span 40;
}
```

Шаг 5: Реализация адаптивности (Desktop First)

Добавьте медиазапросы для адаптации интерфейса к разным устройствам. Рекомендуется использовать подход Desktop First так как у вас уже был реализован веб-интерфейс для десктопной версии.

Пример реализации данного подхода (изменяя только размер и цвет шрифта):

```
.text {  
    color: navy;  
    font-size: 18px;  
}  
  
/* Ноутбуки (меньше 1200px) */  
@media (max-width: 1199px) {  
    .text {  
        color: darkblue;  
    }  
}  
  
/* Планшеты (меньше 992px) */  
@media (max-width: 991px) {  
    .text {  
        color: blue;  
    }  
}  
  
/* Мобильные в альбомной ориентации (меньше 768px) */  
@media (max-width: 767px) {  
    .text {  
        color: royalblue;  
    }  
}  
  
/* Мобильные (меньше 576px) */  
@media (max-width: 575px) {  
    .text {  
        color: dodgerblue;  
        font-size: 16px;  
    }  
}  
  
/* Очень маленькие экраны (меньше 400px) */  
@media (max-width: 399px) {  
    .text {  
        color: lightblue;  
        font-size: 14px;  
    }  
}
```

Шаг 6: Проверка и публикация

1. Протестируйте отображение на разных размерах экрана с помощью инструментов разработчика
 2. Убедитесь в корректности работы Flexbox, Grid и медиазапросов
 3. Проверьте соответствие кода выбранной методологии
 4. Зафиксируйте изменения в системе контроля версий
 5. Опубликуйте обновлённую версию на GitHub Pages
-

Критерии оценивания

1. **Семантическая вёрстка** — корректное использование HTML5-тегов
 2. **Flexbox** — навигация реализована с использованием flex-контейнера
 3. **Grid** — созданы две разные сетки (равномерная и masonry)
 4. **Медиазапросы** — реализован Desktop First подход с breakpoints
 5. **Адаптивность** — страница корректно отображается на всех типах устройств
 6. **Качество кода** — чистый, хорошо структурированный код без ошибок
-

Дополнительные задания (по желанию)

1. Добавьте плавные анимации при изменении размеров сетки.
 2. Попробуйте изменить основные цвета интерфейса при переключении между различным типами устройств (изменении ширины экрана).
-

Формат отчёта

1. В область для загрузки прикреплена ссылка на GitHub Pages/
 2. В комментариях к ответу прописаны реализованные breakpoints. Пример: реализовано 3 breakpoints (320px, 720px, 1024px)
-

Чек-лист самопроверки

Перед отправкой работы проверьте:

- Меню адаптируется на мобильных устройствах (например, горизонтальное → вертикальное)
- Grid-сетка меняет количество колонок на разных экранах
- Masonry-элементы имеют разную высоту и правильно располагаются (при выполнении шага 4)
- На всех реализованных breakpoints страница выглядит корректно
- Нет горизонтального скролла на мобильных устройствах, изображения и текст не выходят за границы контейнеров
- Код проходит валидацию (HTML Validator, CSS Validator)