

Практическая №1

Практическая Работа: Верстка Макета с Использованием Техники БЭМ и Адаптивной Верстки

В данной работе мы создадим веб-страницу, реализуя адаптивный макет в формате блоков на основе технологии БЭМ. Мы также будем использовать CSS Flexbox и Grid для построения макета. Макет будет состоять из четырех секций, каждая из которых будет представлена в определенном формате (flex, flex с абсолютным позиционированием, grid и flex).

Цели практической работы:

Реализация структуры макета с использованием подхода БЭМ.

Обеспечение адаптивности верстки с помощью медиа-запросов.

Создание трех различных секций, каждая с использованием различных подходов к размещению элементов.

Ссылка на макет:

<https://www.figma.com/design/ANckcuQmf2qqhRO4D3ACqZ/Untitled?node-id=0-1&t=bQZ6UZACqYa16J7I-1>

Теоретический Материал для Практической Работы по Верстке с Использованием БЭМ и Адаптивной Верстки

В данном разделе мы рассмотрим теоретические основы, на которых основана практическая работа. Мы уделим внимание методологии БЭМ, подходам к адаптивной верстке с использованием CSS Flexbox и Grid, а также основным элементам HTML-разметки.

1. Методология БЭМ

БЭМ (Блок, Элемент, Модификатор) — это методология организации и именования кода CSS и HTML, которая позволяет легко поддерживать и расширять веб-приложения. Основные понятия:

Блок: независимая сущность, которая имеет смысл в контексте интерфейса. Блоки могут быть как простыми (например, кнопка или заголовок), так и сложными (например, навигационное меню). В коде будет представлено как `<div class="button">`.

Элемент: составная часть блока, которая не может существовать отдельно от него. Элементы описываются через двойное подчёркивание: `block__element` (например, ``).

Модификатор: обозначает разные состояния или версии блока или элемента, которые изменяют их внешний вид. Модификаторы описываются через двойное тире: `block--modifier` или `block__element--modifier` (например, `<div class="button button_large">`).

Примеры использования БЭМ:

Блок header

Элементы: header__title, header__nav

Модификаторы: header_fixed

Используя БЭМ, мы можем создавать составные компоненты, легко управляя программным кодом и стилями.

2. Адаптивная Верстка

Адаптивная верстка — это подход разработке сайтов, который предполагает, что интерфейс будет плавно подстраиваться под различные размеры экрана (от мобильных до планшетов и десктопов). Это достигается за счет использования:

Медиа-запросов: Позволяют применять разные стили в зависимости от характеристик устройства (например, ширина экрана). С помощью медиазапросов можно изменять цвета, шрифты, расположение элементов и даже порядок их отображения.

CSS

```
@media (max-width: 600px) {  
  .flex-container {  
    flex-direction: column; /* Изменение направления для мобильных */  
  }  
}
```

Отзывчивых единиц измерения: Ширина, высота и другие свойства могут быть заданы в процентах, 'em', 'rem' и 'vw/vh', что сохраняет пропорции и позволяет адаптировать элементы под размер родительского контейнера.

3. Flexbox

Flexbox (Flexible Box Layout) — это модель разметки, предназначенная для распределения пространства вдоль одной оси. Она упрощает выравнивание элементов и позволяет легко управлять пространством между ними. Основные свойства:

- `display: flex;` - устанавливает контейнер как flex-контейнер.
- `flex-direction` - задает направление расположения flex-элементов (по умолчанию — row).
- `justify-content` - свойство, определяющее выравнивание по основной оси (например, space-between, center).
- `align-items` - свойство, определяющее выравнивание по поперечной оси (например, stretch, center, flex-start, flex-end).

Пример Flexbox:

CSS

```
.container {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

4. CSS Grid

CSS Grid Layout позволяет создавать двумерные сетки, организовывать элементы как по строкам, так и по колонкам. Это мощный инструмент для построения сложных макетов. Основные понятия:

- `display: grid;` - создаёт грид-контейнер.
- `grid-template-columns` и `grid-template-rows` - определяют количество и размеры колонок и строк.
- `grid-column` и `grid-row` - позволяют контролировать, какие элементы располагаются в каких ячейках сетки.

Пример создания сетки:

CSS

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr); /* три равные колонки */  
  grid-gap: 10px; /* расстояние между элементами сетки */  
}
```

5. Работа с позиционированием

Позиционирование элементов также важно для достижения нужного макета. Основные значения:

- `position: relative;` - элемент позиционирован относительно его исходного положения.
- `position: absolute;` - элемент позиционирован относительно ближайшего предка с позиционированием (`relative`, `absolute` или `fixed`).

Пример:

CSS

```
.relative-container {
```

```
position: relative;
}

.absolute-item {
  position: absolute;
  top: 20px;
  left: 20px;
}
```

Описание Структуры

- Header: Обозначает заголовок страницы с навигацией. Использует flex для выравнивания элементов по горизонтали.
- Секция 1 (Flex): Включает три элемента, которые равномерно расположены в одну строку с помощью flex-контейнера.
- Секция 2 (Flex + Position): Включает три карточки, заданные с абсолютным позиционированием, разместив их по центру контейнера.
- Секция 3 (Grid): Использует сетку для размещения элементов, которые автоматически заполняются по ширине контейнера.
- Секция 4 (Flex): Как и первая, последняя секция использует flex для выравнивания контента.

Заключение

Эта практическая работа позволяет ознакомиться с принципами верстки веб-страниц, используя методологию БЭМ, а также с подходами адаптивной верстки, включая Flexbox и Grid. Вы можете изменять параметры стилей, добавлять больше контента или настраивать визуальные элементы, чтобы углубить свои знания и улучшить навыки верстки. Реализовать данный макет в виде блоков и текст рыбы