

ЗАДАНИЕ № 12. ВЕРСТКА ПО МАКЕТУ.

В папке **sample** представлена страница Достижения.

На этой странице расположен макет слайдера Достижения Омской области

Проанализируйте код и макет и сверстайте страницу в соответствии с алгоритмом. В работе задействованы некоторые новые CSS свойства, например `border`. Изучите их работу самостоятельно, попробуйте задать разные значения, посмотрите справку по свойствам.

Макет в файле `achievements.png`

1. Создаем верстку

```
<div class="container">
  <div class="slide">
    <div class="slide-photo image">
      
    </div>
    <div class="slide-content">
      <div class="text">
        <h2>Завод синтетического каучука, конец 1960-х гг.</h2>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Lorem ipsum dolor, sit amet consectetur
        adipisicing elit.Lorem ipsum dolor, sit amet consectetur adipisicing elit.Lorem ipsum dolor, sit
        amet consectetur adipisicing elit.Lorem ipsum dolor, sit amet consectetur adipisicing elit.Lorem
        ipsum dolor, sit amet consectetur adipisicing elit.</p>
        <input type="button" value="Подробнее">
      </div>
    </div>
  </div>
</div>
```

2. Добавляем в контейнер радиокнопки для переключения слайдера.

```

<div class="slide-control">
  <input type="radio" name="slide" id="btn1">
  <label for="btn1"></label>
  <input type="radio" name="slide" id="btn2">
  <label for="btn2"></label>
  <input type="radio" name="slide" id="btn3" checked>
  <label for="btn3"></label>
  <input type="radio" name="slide" id="btn4">
  <label for="btn4"></label>
  <input type="radio" name="slide" id="btn5">
  <label for="btn5"></label>
  <input type="radio" name="slide" id="btn6">
  <label for="btn6"></label>
</div>

```

3. В стилях добавляем переменные цветов и скругление границ для кнопки, как вариант. Такой вариант используется для удобства настройки, если в проекте много цветов и прочих подобных свойств. Можете задавать эти значения непосредственно в правилах или здесь, на ваш выбор.

```

html {
  --puzz-color: #ffbe99;
  --blue-color: #7f92c2;
  --violet-color: #00876d;
  --border-radius: 30px;
}

```

4. Повторяем оформление шапки, подвала и меню из предыдущего задания.
5. Настраиваем блок слайдера, обратите внимание на комбинированные правила, такой вариант обращения к элементам тоже возможен для сокращения кода.

```

div.slide {
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: space-between;

  div.slide-photo{
    background-color: #00876d;
    height: 400px;

    img{
      height: 400px;
      object-fit: cover;
    }
  }

  div.slide-content{
    text-align: center;
    width: 50%;
    padding: 1% 5%;

    h2{
      font-size: 2em;
      margin-bottom: 50px;
      font-weight: lighter;
    }
  }
}

```

6. Настраиваем стили для радиокнопок. Обратите внимание для переключения состояния кнопок используется псевдокласс `:checked`. Пока просто используем, псевдоклассы будем изучать позднее.

Прописываются стили состояния кнопок в нажатом состоянии. Затем еще раз в не нажатом.

```

div.slide-control{
  margin: 20px 0 40px 0;
  display: flex;
  justify-content: center;
  align-items: center;

  input[type=radio]{
    display: none;
  }

  label{
    display: block;
    width: 20px;
    height: 20px;
    margin: 0 10px;
    border-radius: 100px;
    border: 2px var(--violet-color) solid;
  }

  input#btn1:checked+label,
  input#btn2:checked+label,
  input#btn3:checked+label,
  input#btn4:checked+label,
  input#btn5:checked+label,
  input#btn6:checked+label {
    padding: 1px;
    background-color: var(--puzz-color);
  }

  input#btn1:not(:checked)+label,
  input#btn2:not(:checked)+label,
  input#btn3:not(:checked)+label,
  input#btn4:not(:checked)+label,
  input#btn5:not(:checked)+label,
  input#btn6:not(:checked)+label {
    background-color: #ffffff;
    border: 2px var(--violet-color) solid;
  }
}

```

Подразумевается, что здесь мы должны подключить JS для работы слайдера, но пока этой задачи перед нами не стоит.

7. В завершении делаем красивое оформление кнопки. Как видите здесь тоже используется псевдокласс `:hover` и свойство `cursor` для изменения состояния кнопки при наведении курсора.

```
input, button{
  font-family: Shall;
  font-size: 24pt;
  width: 100%;
  margin-top: 40px;
  padding: 20px;
  border-radius: var(--border-radius);
  border: 1px transparent solid;
}

input[type=button], button{
  background-color: var(--puzz-color);
  border: 1px transparent solid;
}

input[type=button]:hover{
  cursor: pointer;
  background-color: var(--base-color);
  border: 1px var(--dark-color) solid;
}
```

На этом все, макет слайдера готов. Сама анимация может быть реализована на чистом CSS с помощью свойств `transition` и `transform`, либо с помощью `Java Script`.