

Лабораторная работа №3. Основы CSS. Оформление текста

Подключение стилей оформления

На прошлом этапе мы спроектировали сайт, а на этом — поработаем с оформлением сайта.

Для начала нужно отключить стили для прототипирования, а вместо них подключить «настоящие» стили. Эти стили находятся в файле `style.css`, который можно скачать [отсюда](#).

Разместите файл `style.css` в папке сайта на одном уровне с файлом `index.html`. Затем исправьте название подключаемого файла с `outlines.css` на `style.css` (рис.3).

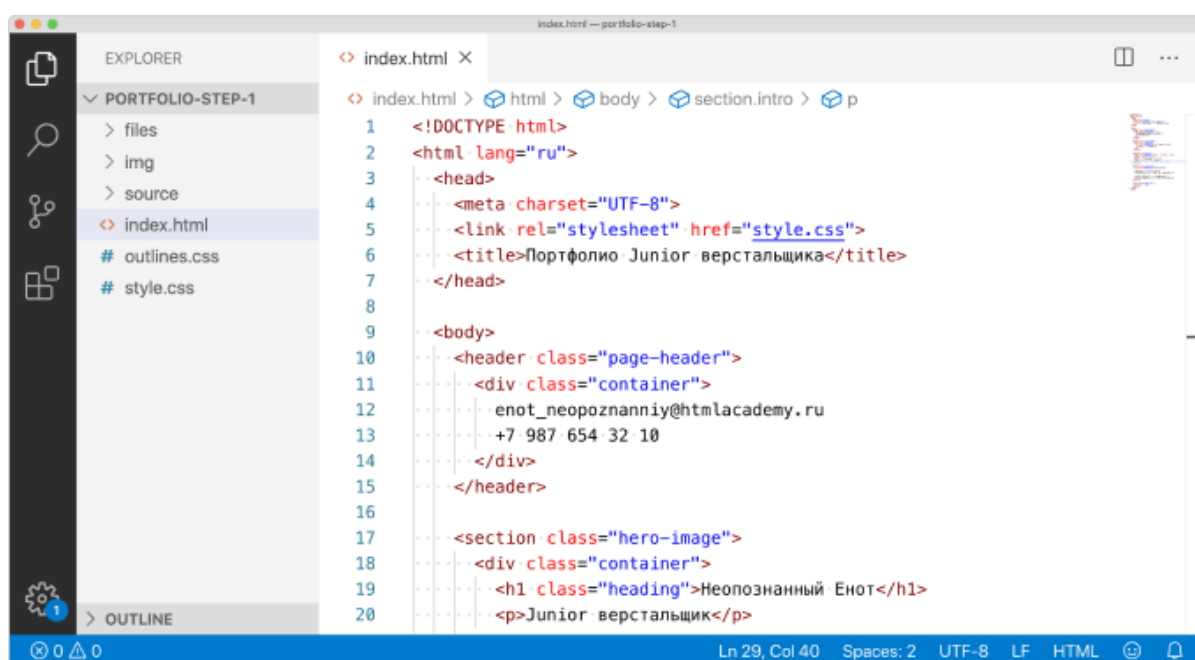


Рисунок 3. – Подключаем «настоящие» стили

Оформление сайта изменилось. Но это только первый шаг.

Доработка разметки сайта

Следующим шагом давайте обернём содержимое каждого смыслового блока (от header до footer) в `div` с классом `container`. Получиться должно примерно вот так:

```
<header class="page-header">
  <div class="container">
    Контакты в шапке
  </div>
</header>
```

Этот служебный «контейнер» отцентрирует наши блоки на экране (рис.4).

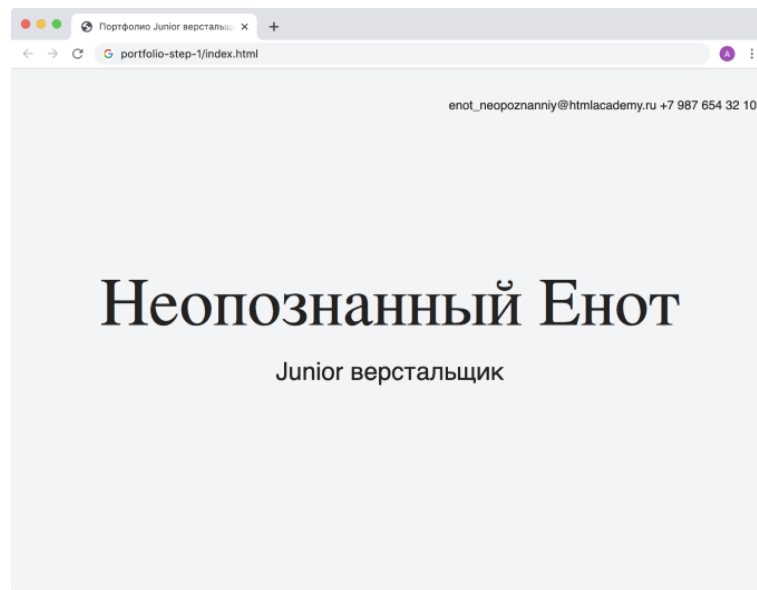


Рисунок 4. – Блоки на странице стали центроваться

Подключение нестандартных шрифтов с Google Fonts

Идём дальше. Чтобы наш сайт выглядел современнее и выделялся из других, подключим на сайт нестандартные шрифты. Проще всего подключить эти шрифты с помощью сервиса Google Fonts. Давайте добавим в head сайта следующую строку (рис.5):

```
<link  
href="https://fonts.googleapis.com/css?family=Montserrat:400,500,700|O1  
d+Standard+TT&display=swap&subset=cyrillic" rel="stylesheet">
```

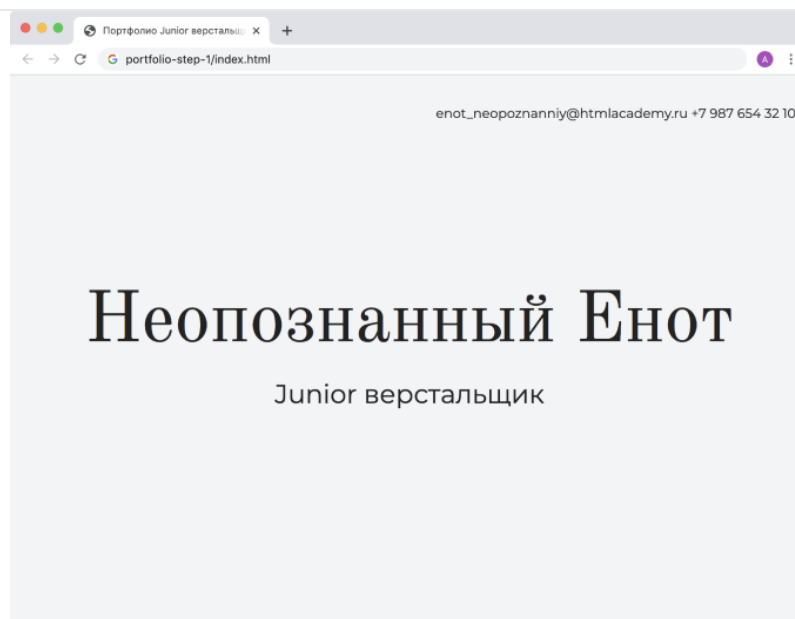


Рисунок 5. – Со шрифтами всё стало выглядеть ещё лучше

Конспект «Оформление текста»

Свойство `font-size`

Управляет размером шрифта. Значение свойства задаёт желаемую высоту символа шрифта. Причём единицы измерения могут быть абсолютными или относительными.

Самая часто используемая единица измерения размера шрифта — пиксели `px`:

```
p {  
  font-size: 20px;  
}
```

Но, чтобы при изменении основного размера шрифта родителя его дочерние элементы пропорционально меняли свои размеры шрифта, есть специальная единица измерения — `em`.

Величина `1em` — это такой же размер шрифта, что и у родителя. Соответственно, если нужно, чтобы шрифт дочернего элемента был всегда в 2 раза больше родительского, то надо задать значение `2em`:

```
h1 {  
  font-size: 2em;  
}
```

Свойство `line-height`

Свойство управляет высотой строки или межстрочным интервалом.

По умолчанию это свойство имеет значение `normal`. Оно указывает браузеру, что межстрочный интервал нужно подобрать автоматически, исходя из размера шрифта. Спецификация рекомендует устанавливать его в пределах 100-120% от размера шрифта. То есть:

```
p {  
  font-size: 10px;  
  line-height: normal; /* значение будет примерно 12px */  
}
```

Значение `normal` позволяет всем нестилизированным текстам выглядеть удобочитаемо. Однако, если есть необходимость отойти от стилизации по умолчанию, `line-height` можно задать фиксированное абсолютное значение в `px`.

```
p {  
  font-size: 16px;  
  line-height: 26px;  
}
```

Если нужно задать `line-height` относительное значение, но не такое, как `normal`, то значение задаётся в процентах или в виде множителя. В таком случае браузер вычисляет значение динамически в зависимости от `font-size`:

```
p {
  font-size: 10px;
  line-height: 150%; /* вычисленное значение: 10px * 150% = 15px */
  line-height: 2;    /* вычисленное значение: 10px * 2 = 20px */
}
```

Относительные значения более гибкие, чем абсолютные. Но для простых сайтов фиксированных `font-size` и `line-height` будет вполне достаточно.

Свойство `font-family`

В значении этого свойства указывается список названий шрифтов, перечисленных через запятую. В начале списка располагают самый желаемый шрифт, затем менее желаемый, а в самом конце списка — общий тип шрифта. Браузер проходит по списку слева направо и использует первый найденный в системе или на сайте шрифт. Если название шрифта состоит из нескольких слов, то его нужно заключать в кавычки.

```
body {
  font-family: "PT Sans", "Arial", sans-serif;
}
```

Свойство `font-weight`

Свойство задаёт насыщенность или толщину шрифта. Шрифт может быть жирнее или тоньше обычного начертания. В качестве значения можно использовать ключевое слово или число. Самые часто встречающиеся значения:

- 400 или `normal` — обычный шрифт, значение по умолчанию;
- 700 или `bold` — жирный шрифт.

Например:

```
h1 {
  font-weight: 400; /* то же самое что и normal */
}

p {
  font-weight: bold; /* то же самое что и 700 */
}
```

Свойство `text-align`

Описывает, как выравнивается текст и другие инлайновые элементы (изображения, инлайн-блоки, инлайн-таблицы и другие) внутри блока по горизонтали.

Свойство может принимать следующие значения:

1. `left` — выравнивание по левому краю блока, это значение по умолчанию;
2. `right` — по правому краю блока;
3. `center` — по центру блока;
4. `justify` — по ширине блока, при этом слова в строке будут размещаться так, чтобы занять равномерно всё пространство строки (пробелы между словами)

в таком случае становятся неравномерными, так как браузер «растягивает» слова в строке).

Важно помнить, что свойство `text-align` применяется именно к самому блоку-контейнеру, внутри которого находится текстовый контент:

```
HTML:
<p>
  Я текст внутри абзаца
</p>

CSS:
p {
  text-align: center;
}
```

Свойство `vertical-align`

Свойством можно выравнивать инлайновые элементы относительно содержащей его строки. Самый простой пример — выровнять картинку `` по вертикали в текстовой строке.

У свойства `vertical-align` много значений, но самые часто используемые:

1. **top** — выравнивание по верхнему краю строки;
2. **middle** — по середине строки;
3. **bottom** — по нижнему краю строки;
4. **baseline** — по базовой линии строки (значение по умолчанию).

В отличие от `text-align` свойство `vertical-align` задаётся самому элементу, а не содержащему его контейнеру:

```
HTML:
<p>
  
  Я текст внутри абзаца
</p>

CSS:
img {
  vertical-align: middle;
}
```

Свойство `color`

Цветом текста можно управлять свойством `color`.

Цвет может быть задан в виде ключевого слова (полный список ключевых слов приводится в спецификации). Например:

```
color: black; /* чёрный цвет */
color: red;   /* красный цвет */
color: white; /* белый цвет */
```

Ещё один вариант указания цвета — в виде шестнадцатеричного значения. В этом случае цвет формируется

из красной, зелёной и синей составляющих, заданных в виде шестнадцатеричного числа от 00 до ff. Помимо шести, цветовой код может содержать три знака, в этом случае второй символ в цветовых составляющих дублируется первым:

```
color: #000000; /* чёрный цвет */
color: #f00;    /* красный цвет, то же что #ff0000 */
color: #fff;    /* белый цвет, то же что #ffffff */
```

Если не хочется иметь дело с шестнадцатеричными значениями, можно воспользоваться специальной функцией `rgb`, в которой указывается цвет в более привычном десятичном виде в диапазоне от 0 до 255 также в виде трёх цветовых составляющих, перечисленных через запятую:

```
color: rgb(0, 0, 0)      /* чёрный, то же что #000000 */
color: rgb(255, 0, 0)    /* красный, то же что #ff0000 */
color: rgb(255, 255, 255) /* белый, то же что #ffffff */
```

У функции `rgb` есть расширенная версия — `rgba`. В этом случае помимо указания цвета последним значением указывается степень непрозрачности цвета — `alpha`. Значение может быть от 0 (полностью прозрачный) до 1 (полностью непрозрачный):

```
color: rgba(0, 0, 0, 0.5) /* чёрный, непрозрачный на 50% */
color: rgba(255, 0, 0, 0.3) /* красный, непрозрачный на 30% */
color: rgba(255, 255, 255, 0.9) /* белый, непрозрачный на 90% */
```

Контраст цвета текста и фона

Фоновое изображение и фоновый цвет блока всегда должен достаточно сильно контрастировать с цветом текста в этом блоке. Чем больше контраст, тем удобнее этот текст читать в разных условиях освещённости и на разных устройствах. Поэтому если вы задаёте блоку фоновое изображение, нужно обязательно дополнительно задавать подходящий фоновый цвет. В этом случае, пока изображение загружается, или в случае, если оно совсем не загрузится, текст всё равно можно будет прочитать:

```
p {
  /* идеальный контраст: цвет текста белый, цвет фона — чёрный */
  background-color: #000000;
  color: #ffffff;
}

span {
  /* плохой контраст: цвет текста и фона — серые */
  background-color: #cccccc;
  color: #dddddd;
}
```

Свойство white-space, управление пробелами

Браузер игнорирует множественные пробелы и переносы строк в HTML-коде. С помощью свойства `white-space` можно управлять пробелами и переносами строк. Свойство принимает значения:

- `nowrap` — схлопывает лишние пробелы и отображает весь текст одной строкой без переносов;
- `pre` — сохраняет пробелы и переносы как в исходном коде аналогично тегу `<pre>`;
- `pre-wrap` — работает как значение `pre`, но добавляет автоматические переносы, если текст не помещается в контейнер;
- `normal` — режим по умолчанию: лишние пробелы и переносы строк схлопываются, текст переносится, пробелы в конце строк удаляются.

Свойство text-decoration

Задаёт дополнительное оформление текста. Значения свойства:

1. `underline` — подчёркивание;
2. `line-through` — зачёркивание;
3. `overline` — надчёркивание;
4. `none` — убирает вышеперечисленные эффекты.

К тексту можно одновременно применить несколько эффектов, если перечислить значения через пробел:

```
p {
  text-decoration: underline; /* подчёркнутый текст */
}

span {
  /* подчёркнутый и зачёркнутый текст */
  text-decoration: underline line-through;
}
```

Свойство `text-decoration` — составное. Оно раскладывается на отдельные свойства:

- `text-decoration-line` — вид линии: зачёркивание, подчёркивание или надчёркивание;
- `text-decoration-style` — стиль линии, может принимать значения:
 - `solid` — сплошная линия;
 - `double` — двойная линия;
 - `dotted` — точечная линия;
 - `dashed` — пунктирная линия;

- `wavy` — волнистая линия.
- `text-decoration-color` — цвет линии.

Свойство `font-style`

Свойством можно задать начертание текста. Его основные значения:

1. `normal` — обычное начертание;
2. `italic` — курсивное начертание;
3. `oblique` — наклонное начертание.

Если задано значение `italic`, браузер будет пытаться найти в заданном шрифте отдельное курсивное начертание символов. В некоторых шрифтах отдельный курсив предусмотрен.

Если отдельного курсивного начертания в шрифте не предусмотрено, то браузер сделает текст наклонным, то есть симитирует курсив. Что равноценно заданию тексту значения `font-style: oblique`.

Свойство `text-transform`

С его помощью можно управлять регистром символов: делать буквы строчными (маленькими) или заглавными (большими). Значения свойства:

1. `lowercase` — все строчные;
2. `uppercase` — все заглавные;
3. `capitalize` — каждое слово начинается с большой буквы;
4. `none` — отменяет изменение регистра.

Отступы

Важный фактор того, что текст в блоке будет удобочитаемым, это наличие свободного пространства в блоке для этого текста. Вокруг текста должно быть достаточно «воздуха», он не должен «прилипнуть» к краям, ему не должно быть «тесно».

За отступы в CSS отвечают два свойства: `padding` задаёт внутренние отступы в блоке, а `margin` задаёт внешние отступы.

Задание 1.

Выполнить задания по темам

«Основы CSS» (<https://htmlacademy.ru/courses/307>),

«Оформление текста» (<https://htmlacademy.ru/courses/309>)

Результат работы разместить в своей рабочей директории с ЛР№3 в виде скриншота со статистикой.

Как добавить на сайт блок с навыками

Подключение на страницу блока с навыками

Мы продолжаем работать с наполнением нашего сайта-портфолио. На этом шаге мы добавим блок, в котором вы сможете указать, какими технологиями владеете и в какой степени. Приступим.

Список навыков будет располагаться внутри блока intro. Там уже есть ваша фотография, краткая биография и рассказ о том, чем вы можете быть полезны работодателю. Блок навыков мы вставим под ними.

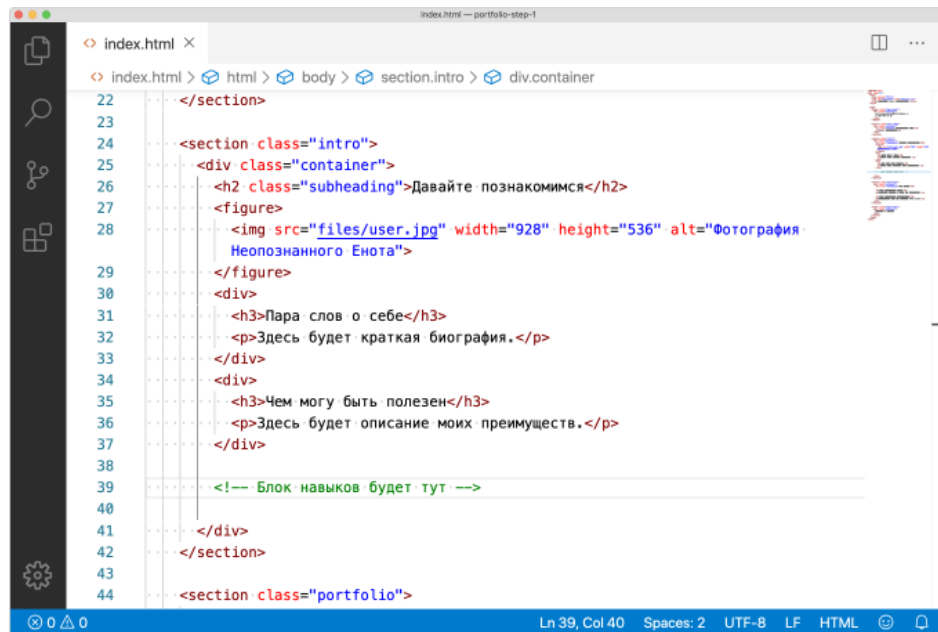


Рисунок 6. Сюда мы добавим блок навыков

Новый блок мы создадим с помощью тега `<div>` с классом `skills`. Внутри добавим заголовок третьего уровня «Мои навыки».

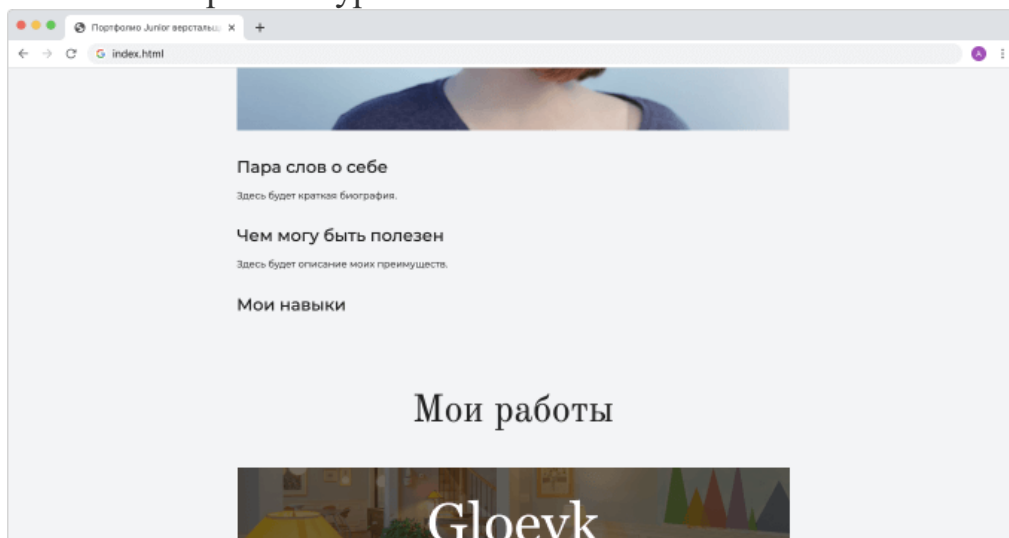


Рисунок 7. Создаём блок и добавляем заголовок

Под заголовком создадим список определений с классом `skills-list`. В нём перечислим освоенные технологии, например, HTML и CSS. В качестве «терминов» будут выступать сами технологии, а в «определениях» мы запишем уровень навыка и обернём значение в дополнительный `<div>`. Вы уже создавали похожий список в части «[Основы CSS](#)».

```
<div class="skills">
  <h3>Мои навыки</h3>
  <dl class="skills-list">
    <dt>HTML</dt>
    <dd><div>60%</div></dd>
    <dt>CSS</dt>
    <dd><div>40%</div></dd>
  </dl>
</div>
```

Добавим классы тегам `<dt>`. Для навыка HTML это будет класс `skill-html`, а для CSS — класс `skill-css`. Тегам `<dd>` добавим класс `level`.

```
<div class="skills">
  <h3>Мои навыки</h3>
  <dl class="skills-list">
    <dt class="skill-html">HTML</dt>
    <dd class="level"><div>60%</div></dd>
    <dt class="skill-css">CSS</dt>
    <dd class="level"><div>40%</div></dd>
  </dl>
</div>
```

Теперь осталось сделать так, чтобы полоска, указывающая на уровень навыка, была той же ширины, что и указанный в качестве значения процент. Для этого используем атрибут `style`.

```
<div class="skills">
  <h3>Мои навыки</h3>
  <dl class="skills-list">
    <dt class="skill-html">HTML</dt>
    <dd class="level"><div style="width: 60%;">60%</div></dd>
    <dt class="skill-css">CSS</dt>
    <dd class="level"><div style="width: 40%;">40%</div></dd>
  </dl>
</div>
```

Готово!

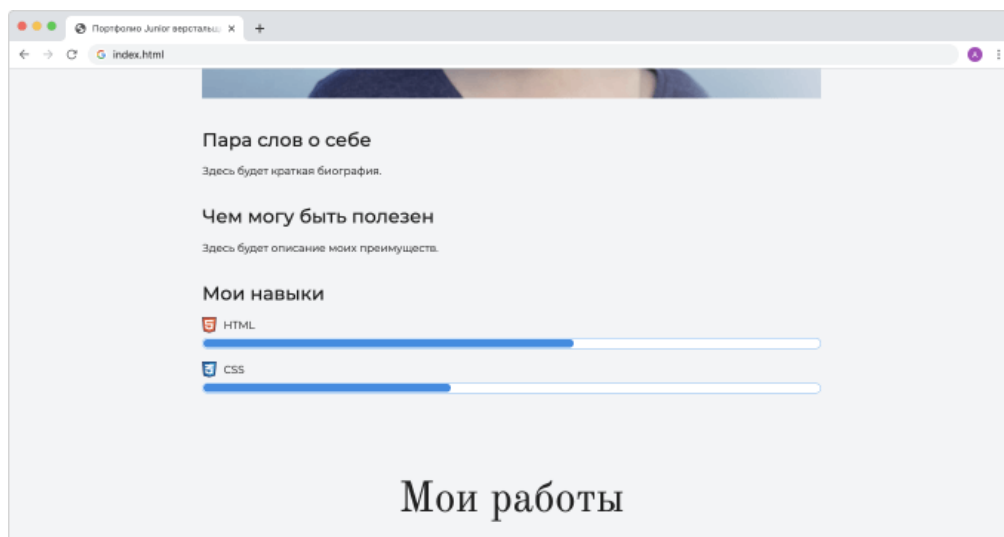


Рисунок 8. Блок навыков готов!

Не забудьте изменить процент владения технологиями, когда научитесь чему-то новому!

Подключение тёмной темы сайта

Мы добавили на сайт-портфолио все блоки и элементы, теперь вернёмся к его оформлению. Многие сайты могут иметь не один дизайн, или тему, а несколько: базовый, облегчённый, тёмный, новогодний и так далее. У вашего сайта-портфолио тоже может быть несколько тем. По умолчанию включена светлая, давайте разберём, как сменить её на тёмную.

Сначала скачайте и распакуйте [архив](#). В нём находятся файл со стилями `style.css` и папка с интерфейсными картинками `img`.

Чтобы поменять тему на сайте, нам не придётся изменять разметку. Обновятся только стили и картинки, поэтому папка `files` с контентными изображениями останется в целости и сохранности. Приступим.

Перейдите в папку с сайтом-портфолио и удалите файл `style.css` и папку `img`.

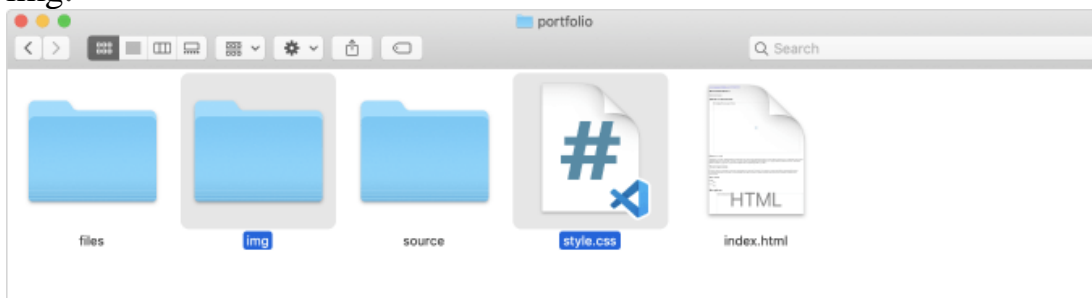


Рисунок 9. Удаляем файл `style.css` и папку `img`

Затем вместо них переместите в папку с сайтом-портфолио соответствующие файлы из скачанного архива.

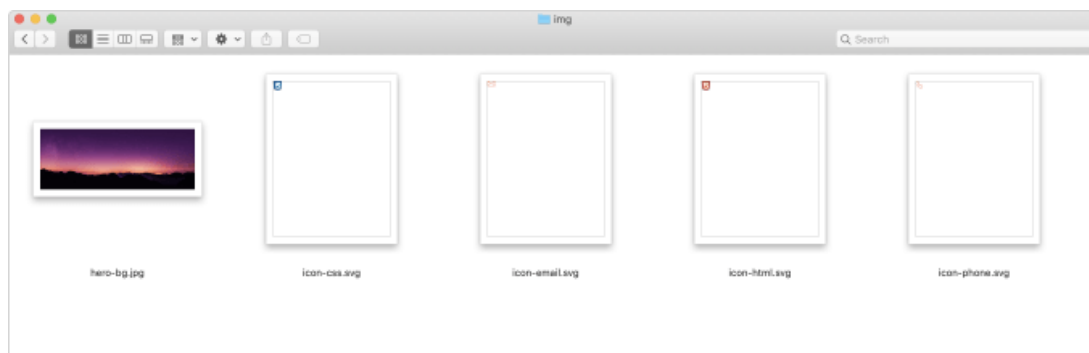


Рисунок 10. Картинки для тёмной темы

Откройте в браузере главную страницу сайта. Теперь она выглядит совсем иначе, при этом разметка не изменилась и всё содержимое осталось на месте. Вот она, сила CSS!

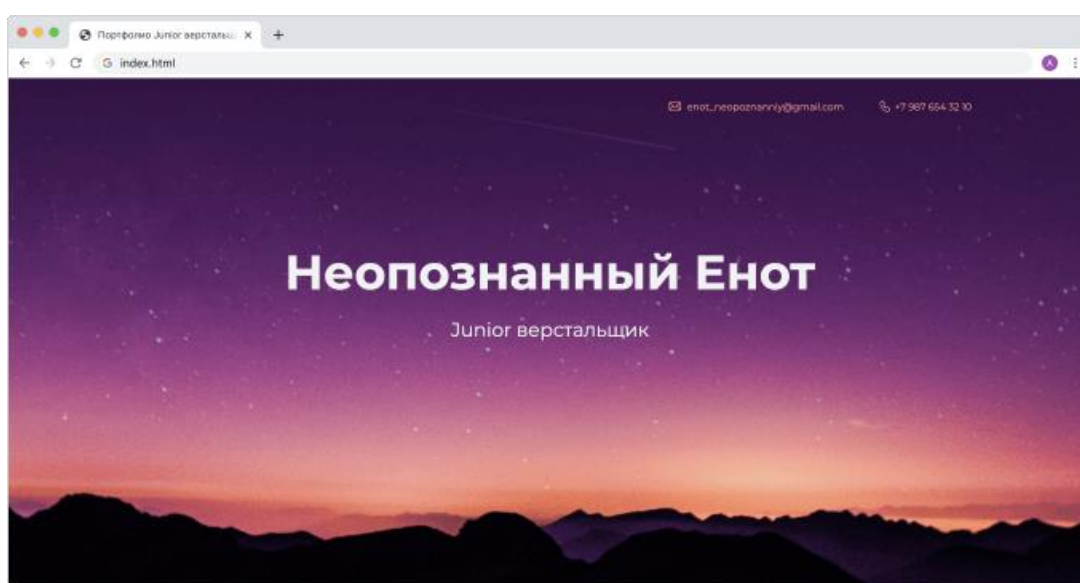


Рисунок 11. Подключаем тёмную тему

В следующей домашней работе вы сможете подключить к своему сайту-портфолио и другие темы.

Задание 2. Доработать задание первого урока, придавая ему стили CSS

1. Создать файл `style.css`, в котором будут храниться все стили вашей работы. Подключить этот файл ко всем страницам.
2. В качестве фона на всех страницах установить цвет `#f8f8f8`.
3. Меню сайта:
 - а. Для всех ссылок меню задать определённый стиль. (Цвет текста, размер шрифта, начертание шрифта и т.д.)
 - б. Убрать маркеры списка.
4. Страница “Подробное описание товара”
 - а. Заголовки (Краткое описание товара, Характеристики, Подробное описание товара)

- i. Цвет текста черный.
 - ii. Размер шрифта 18px;
 - iii. Насыщенность шрифта 400. (font-weight);
 - iv. установите цвет фона #eaeaea;
- b. Для текста краткого описания товар.
 - i. Цвет текста #707070;
 - ii. Размер шрифта 14px.
 - iii. Начертание шрифта italic. (font-style)
 - iv. Высоту текста 16px (line-height);
- c. Для текста подробного описания товара:
 - i. Цвет текста #484343.
 - ii. Размер шрифта 16px.
 - iii. Насыщенность шрифта 400. (font-weight);
 - iv. Высоту текста 24px (line-height);
 - v. Расположение текста по левому краю (text-align)
- d. Для списка внутри подраздела Характеристики товара
 - i. задайте списку стили, отличные от всего остального текста.
 - ii. *установите в качестве маркеров произвольные изображения
- 5. Странице “Контакты”:
 - a. Задать значения ширины и высоты для полей ввода.
 - b. Задать стили для текста, внутри полей input. (Цвет текста, размер шрифта и т.д.).
 - c. Следите за тем, чтобы на странице всё выглядело гармонично, не выбирайте слишком резких цветов.
- 6. *Для изображений, размещенных на странице подробного описания товара, задать рамку произвольным цветом.
- 7. *Приветствуются самостоятельные дополнения ваших работ в пределах пройденной темы.
- 8.* Опубликовать полученную работу на бесплатном хостинге, используя платформу GitHub
 - Результат работы разместить в своей рабочей директории с ПР№3 в виде папки с проектом сайта.
 - Сдать в виде ссылки на директорию с ПР№3.

Вспомогательные материалы

1. Архив с базовыми стилями для сайта-портфолио
<https://github.com/Irina-64/web-interfaces/blob/main/%D0%9C%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B%20%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F%D0%BC/style.css>
2. Архив с base-dark темой для сайта-портфолио
<https://github.com/Irina-64/web-interfaces/blob/main/%D0%9C%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B%20%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F%D0%BC/style.css>

[B8%D0%B0%D0%BB%D1%8B%20%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F%D0%BC/basic-dark-theme.zip](https://github.com/Irina-64/web-interfaces/blob/main/%D0%9C%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B%20%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F%D0%BC/basic-dark-theme.zip)

3. Архив с pro-dark темой для сайта-портфолио

[https://github.com/Irina-64/web-](https://github.com/Irina-64/web-interfaces/blob/main/%D0%9C%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B%20%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F%D0%BC/pro-dark-theme.zip)

[interfaces/blob/main/%D0%9C%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B%20%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F%D0%BC/pro-dark-theme.zip](https://github.com/Irina-64/web-interfaces/blob/main/%D0%9C%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B%20%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F%D0%BC/pro-dark-theme.zip)

4. Как опубликовать свой сайт на GitHub Pages

<https://htmlacademy.ru/courses/basic-html-css/homeworks/29/steps/1>

5. <https://htmlacademy.ru/courses/307>

6. <https://htmlacademy.ru/courses/309>