Добро пожаловать

На отработку по web программированию

— 01

Академия программирования для детей

procoding

Знакомство

- Как тебя зовут?
- Программировал ли ты ранее игры или сайты?
- Чем любишь заниматься в свободное время?

План сегодняшнего занятия:

1. Повторим для чего используется css, html и javascript
2. Начнем верстать свой сайт портфолио и разбираться в исходниках
3. Научимся работать с figma

На рабочем столе необходимо создать папку для проекта

назовем ее "Проект proCoding"

Получаем техническое задание

Техническое задание от заказчика

Задание на разработку сайта визитки

Объект разработки:

Одностраничный сайт-визита

Требования к дизайну и верстке

Дизайн сайта должен соответствовать макету Vizitka.fig в Figma

Обязательно наличие JavaScript

Обязательно наличие JQuery

Основные разделы сайта и их функционал:

Верхняя часть (необходим логотип и кликабельное меню)

Главная (раздел должен содержать картинку и краткую информацию)

Обо мне (раздел должен содержать краткую информацию)

Мои навыки (раздел должен содержать псевдоэлементы и фотографию)

Как я работаю (раздел должен содержать краткую информацию и видеозапись)

Галерея (раздел должен быть выполнен с применением плагина lightzoom)

Форма (раздел должен содержать форму для отправки сообщения)

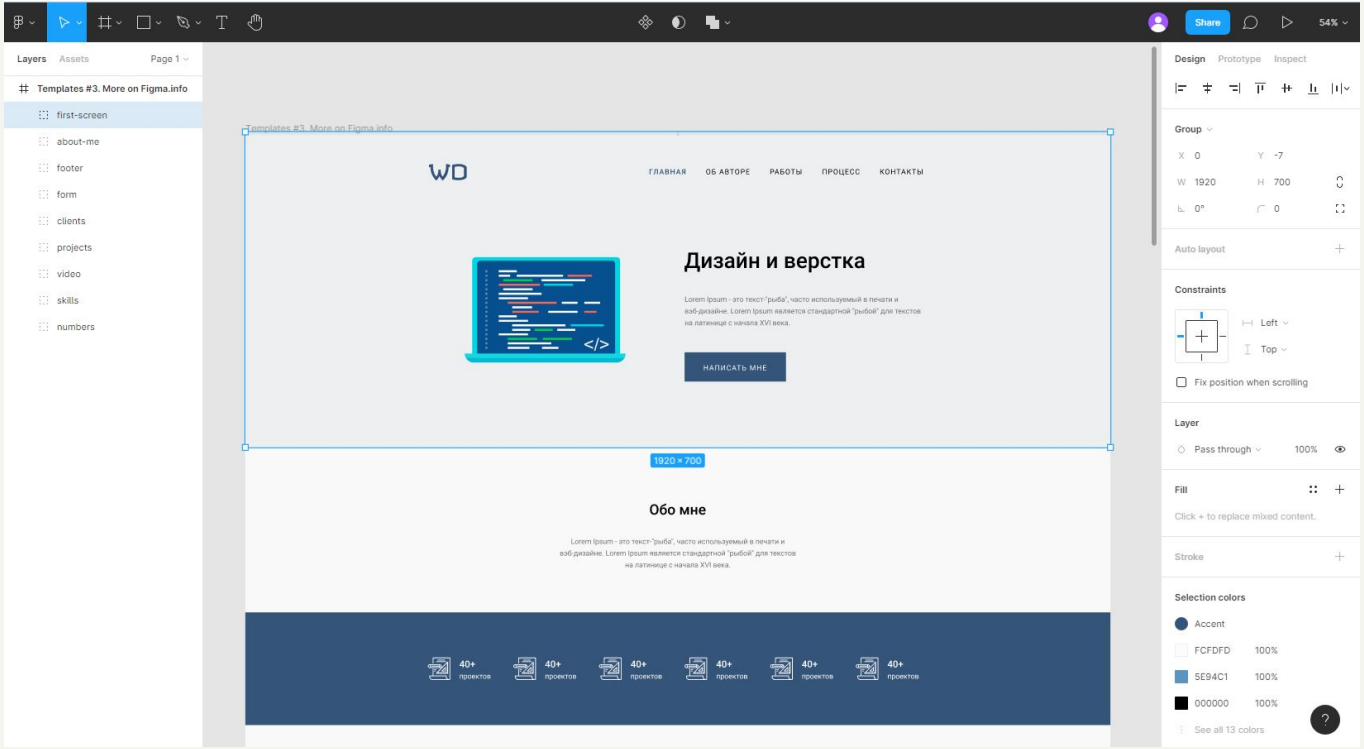
В футере расположить логотипы социальных сетей

Обычно техническое задание (сокращенно ТЗ) от заказчика более объемное.

Оно содержит информацию об основных разделах сайта и пожелания по исполнению (технологии, библиотеки, сроки и изображения)

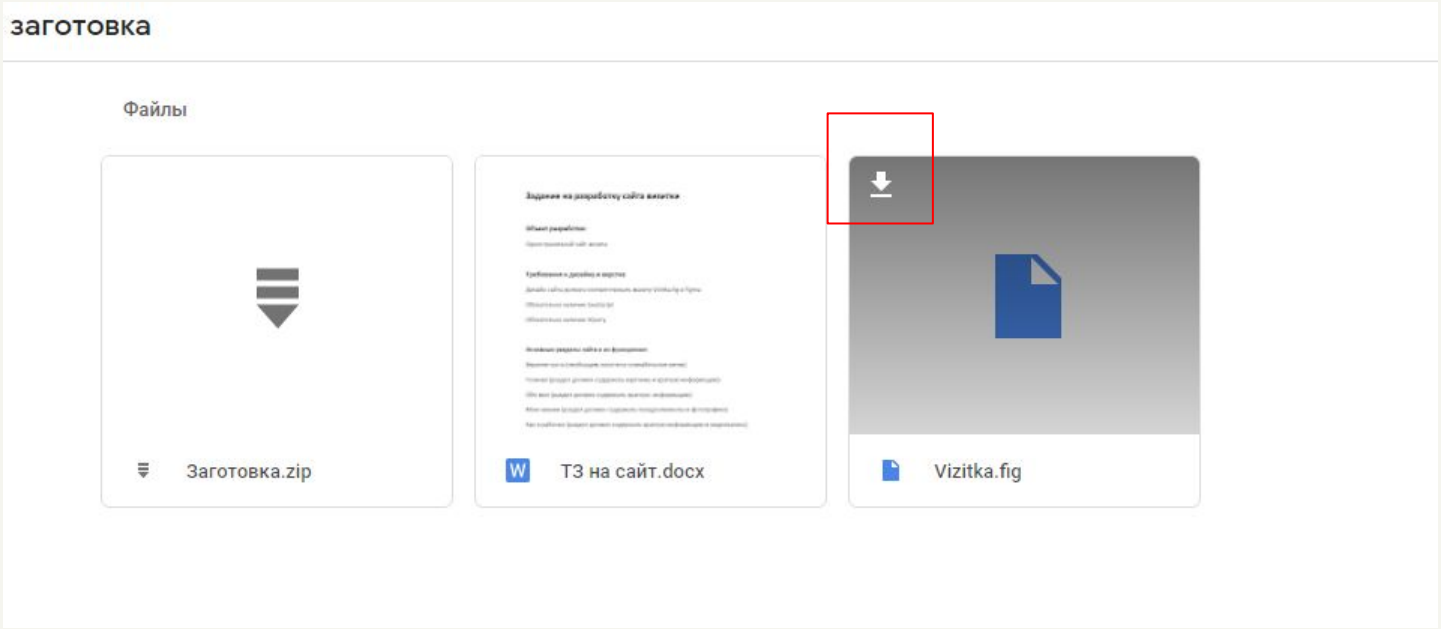
Figma

FIGMA - Программа используемая дизайнерами для разработки WEB интерфейсов



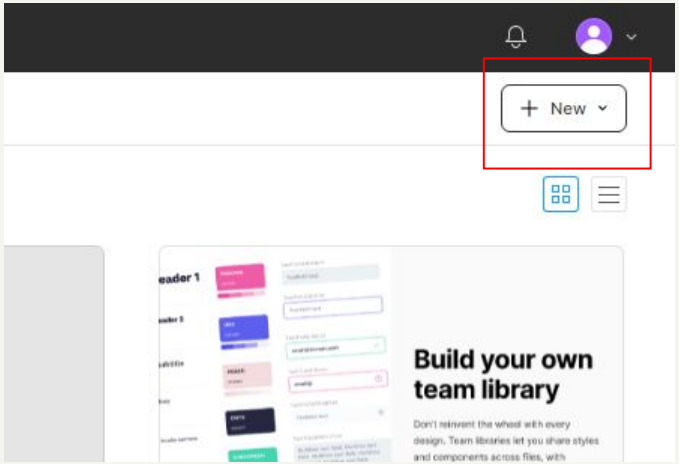
Figma

Скачиваем файл макета от заказчика



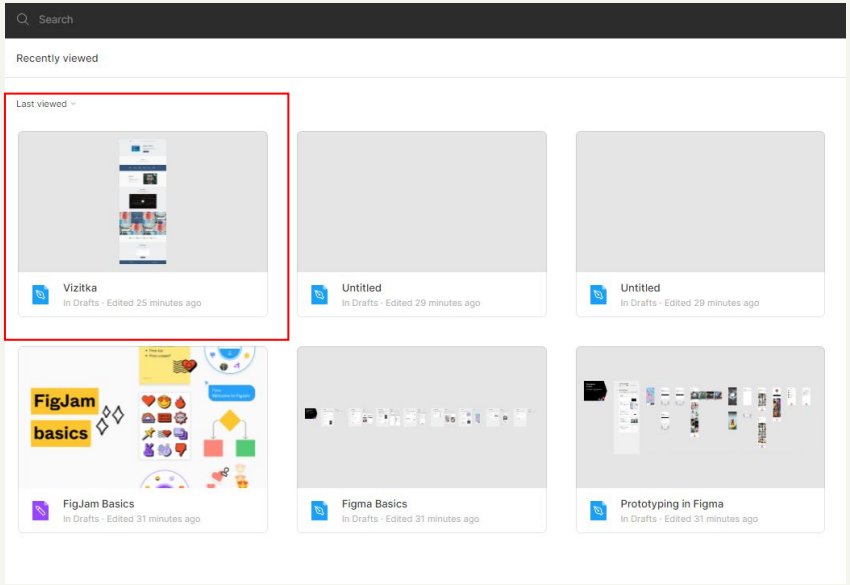
Figma

Загружаем файл макета от заказчика



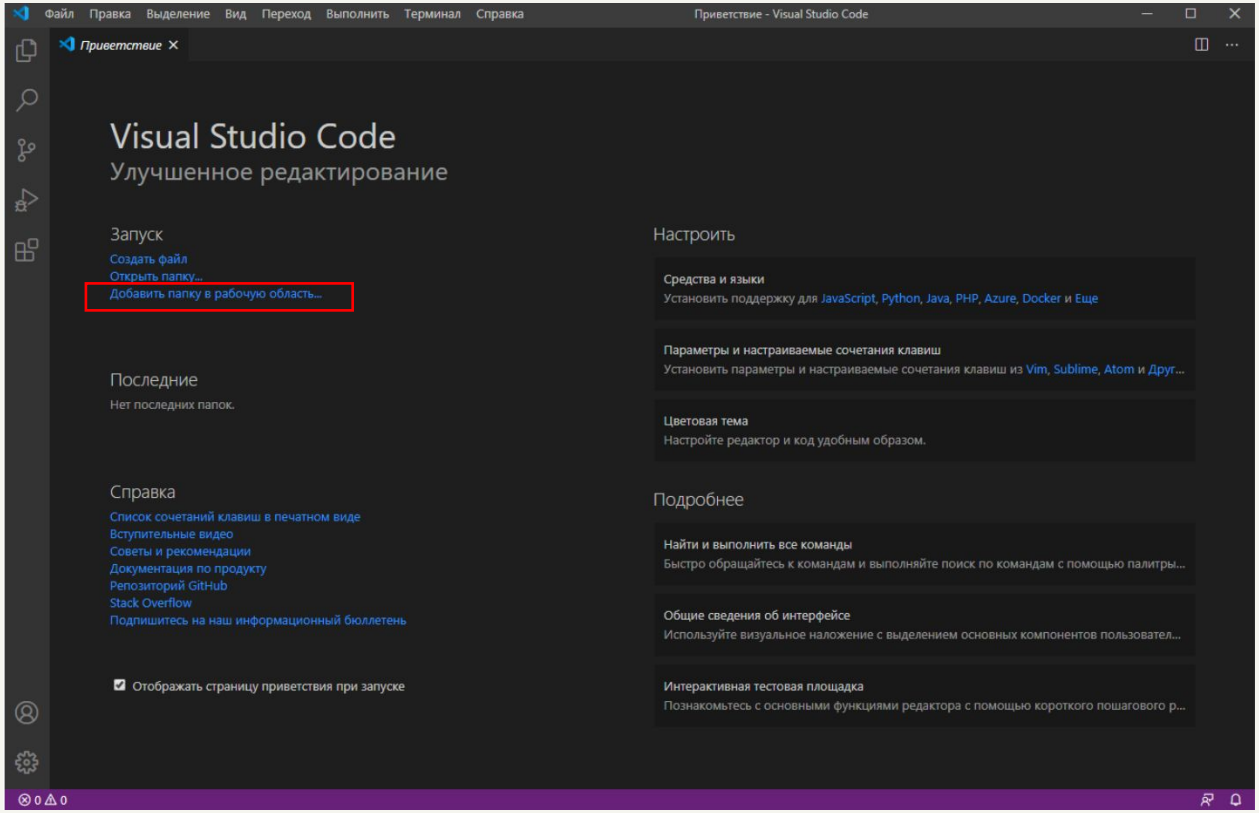
Добавляем наш макет в фигма и открываем

Figma

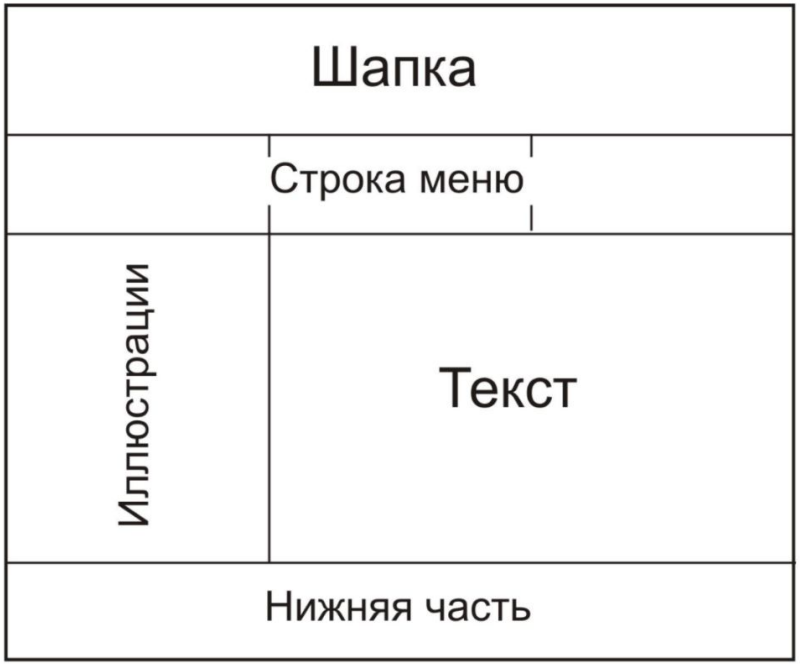


Двойным щелчком открываем на макет

Запускаем программу VS code

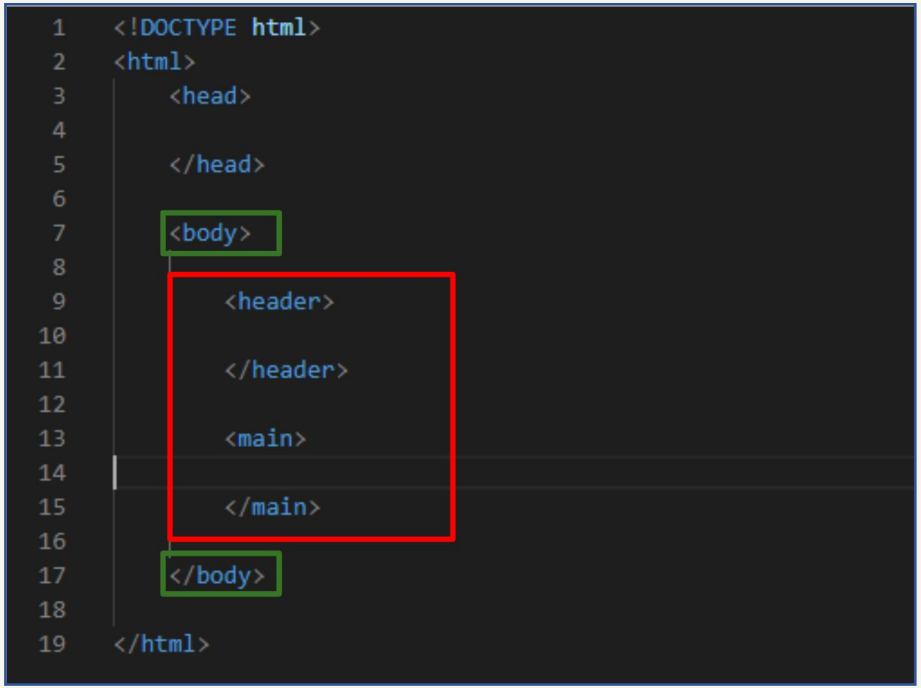


Структура файла index.html



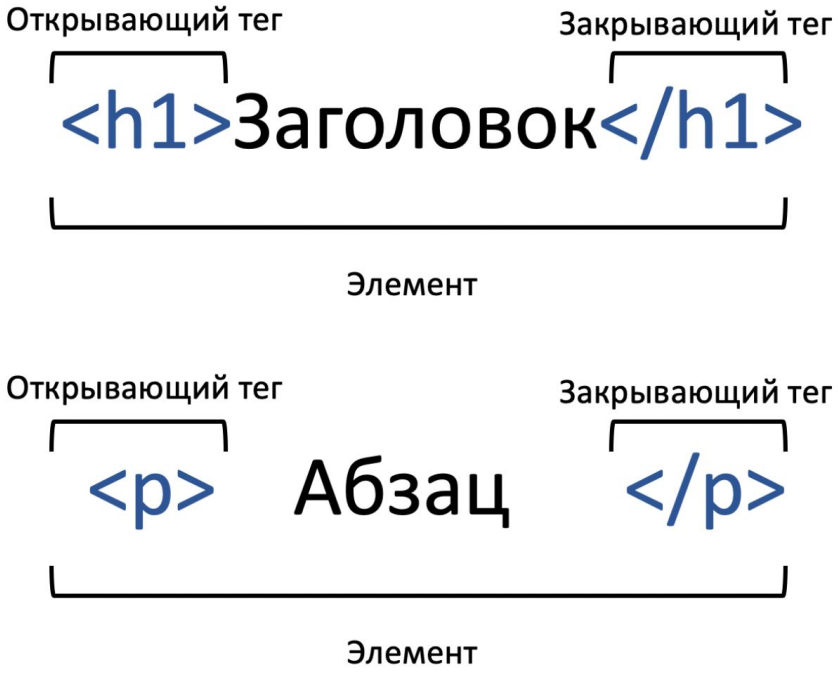

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5      </head>
6
7      <body>
8
9          <header>
10
11         </header>
12
13         <main>
14
15         </main>
16
17     </body>
18
19 </html>
```

Внутрь body помещаем “header” и “main”

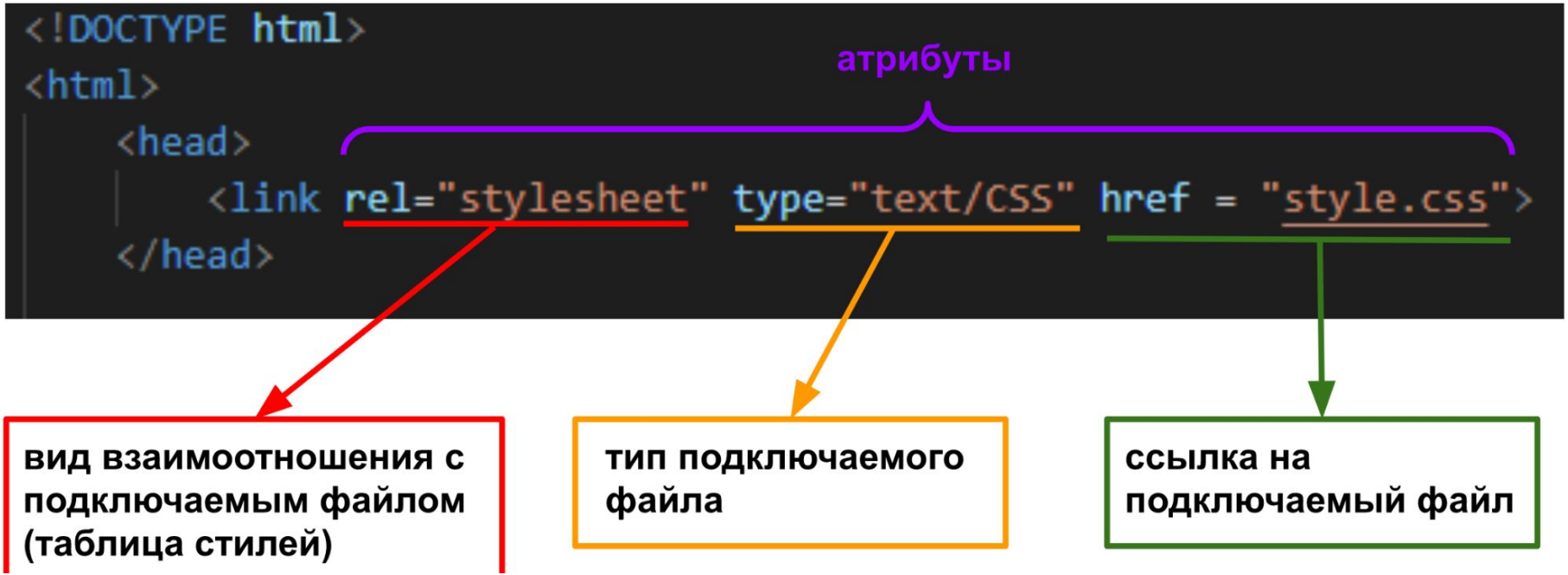


Внутри каждого из блоков будет своя часть проекта

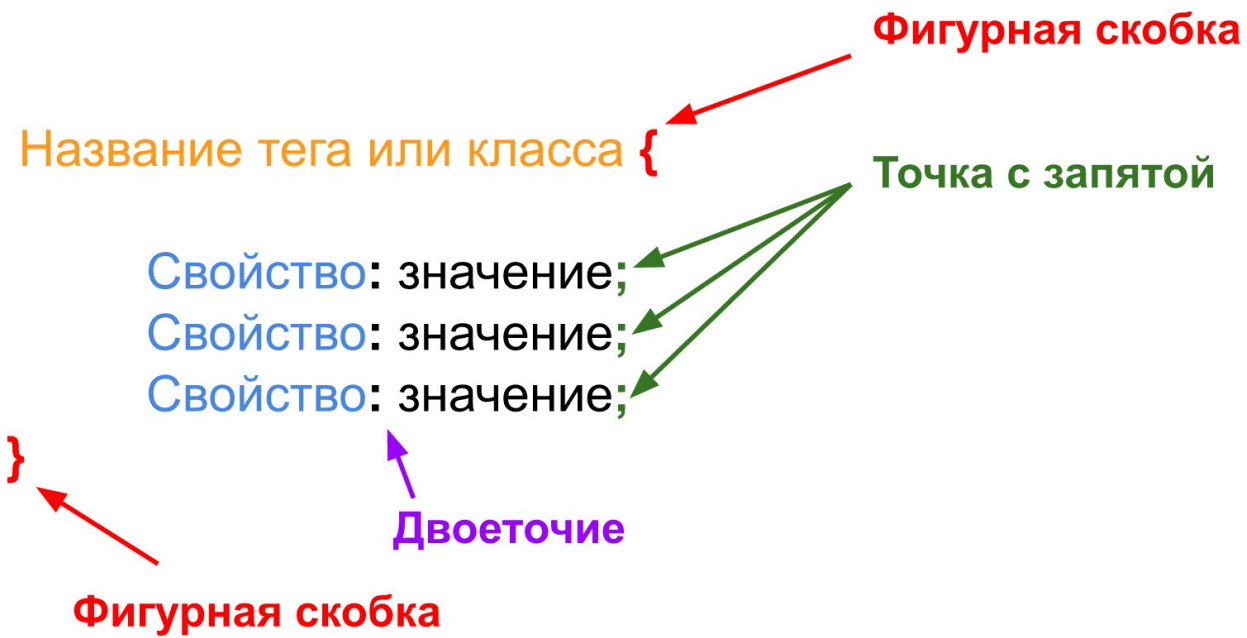
Разберемся с тегами



Подключим style.css в нашем html файле



Правила написания стилей css

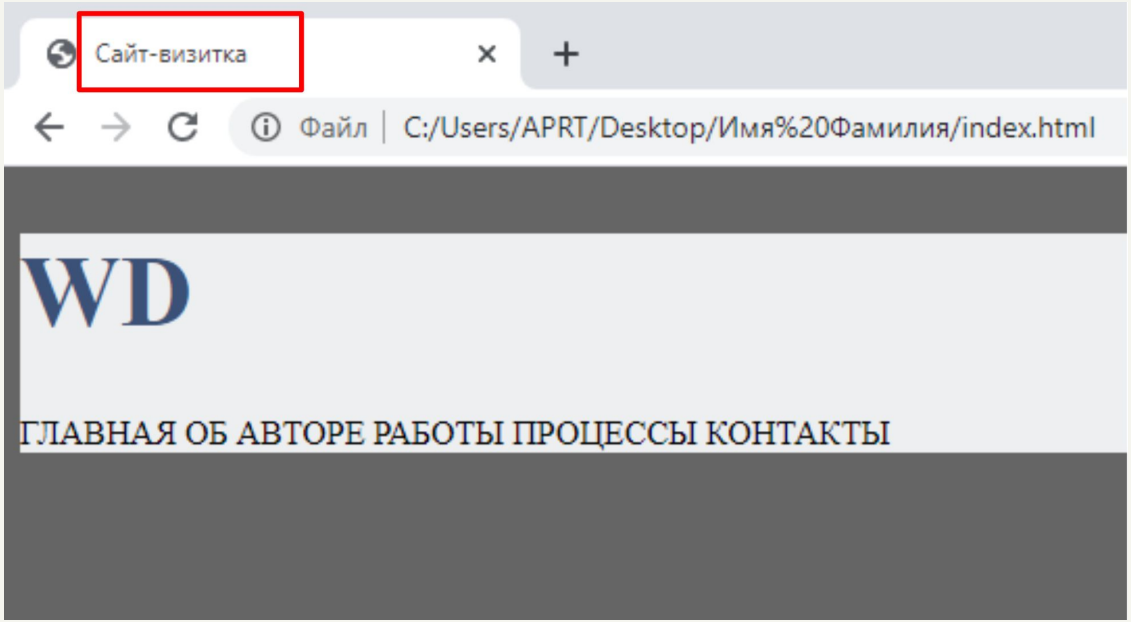


Добавим название нашего сайта

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/CSS" href="style.css">
        <title>Сайт-визитка</title>
    </head>

    <body>

        <header>
            <h1>WD</h1> <p>ГЛАВНАЯ ОБ АВТОРЕ РАБОТЫ ПРОЦЕССЫ КОНТАКТЫ</p>
        </header>

        <main>

        </main>

    </body>

</html>
```

В тег <title></title>
добавьте название вашего сайта

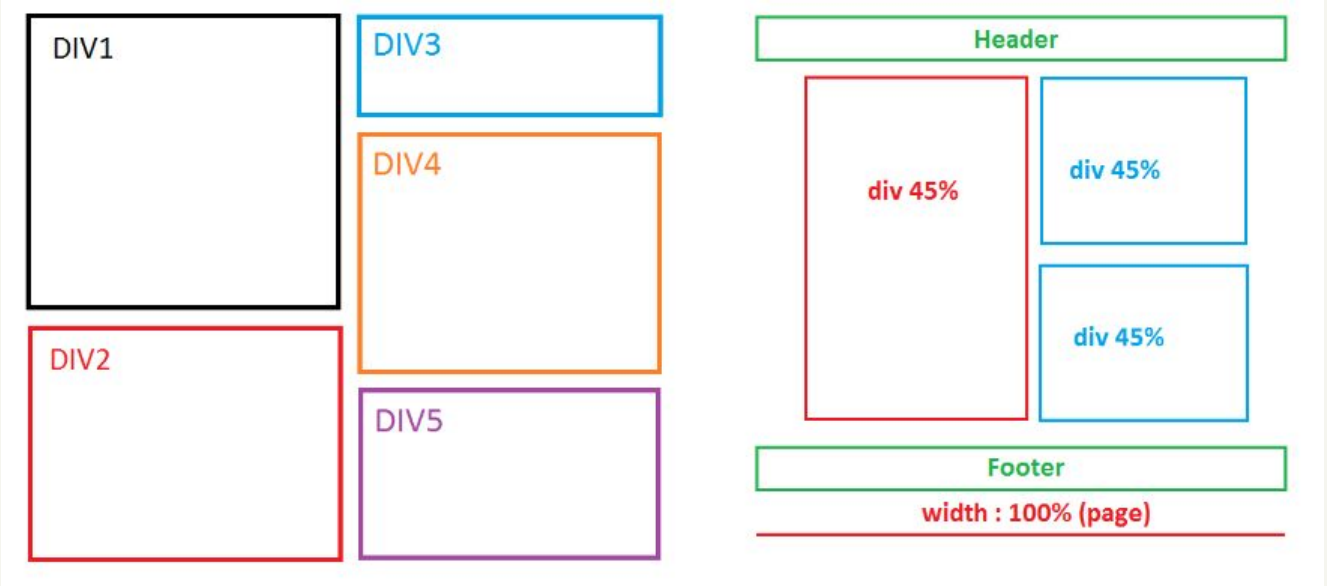
Добавим название нашего сайта



Добавленное название появится во вкладке браузера после обновления страницы

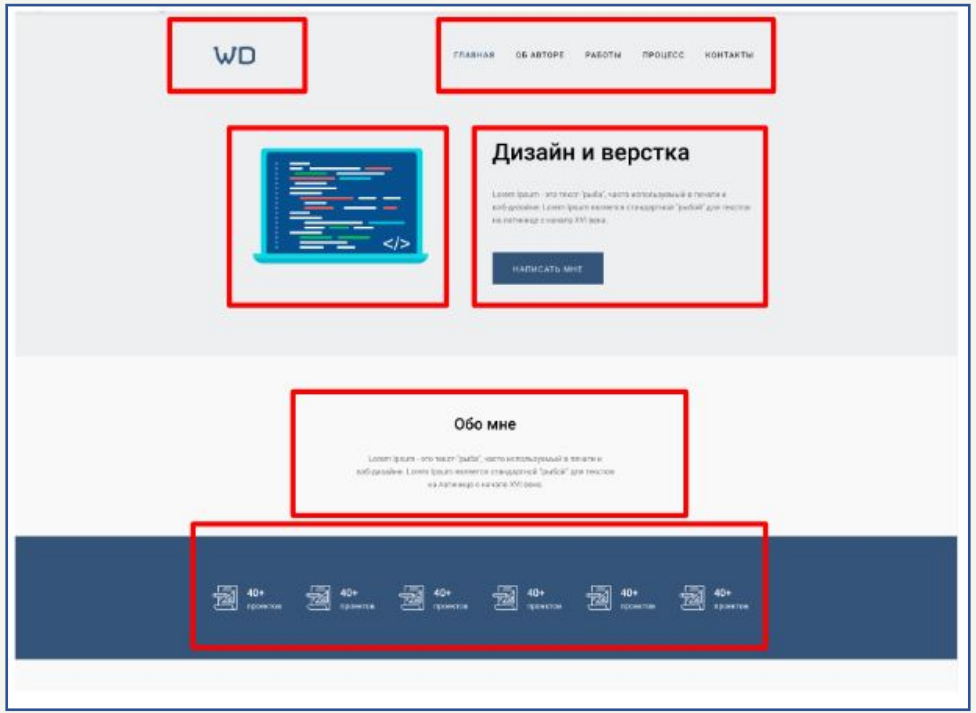
Div элементы

Блоки, контейнеры для контейнера



Div элементы

На примере нашего сайта

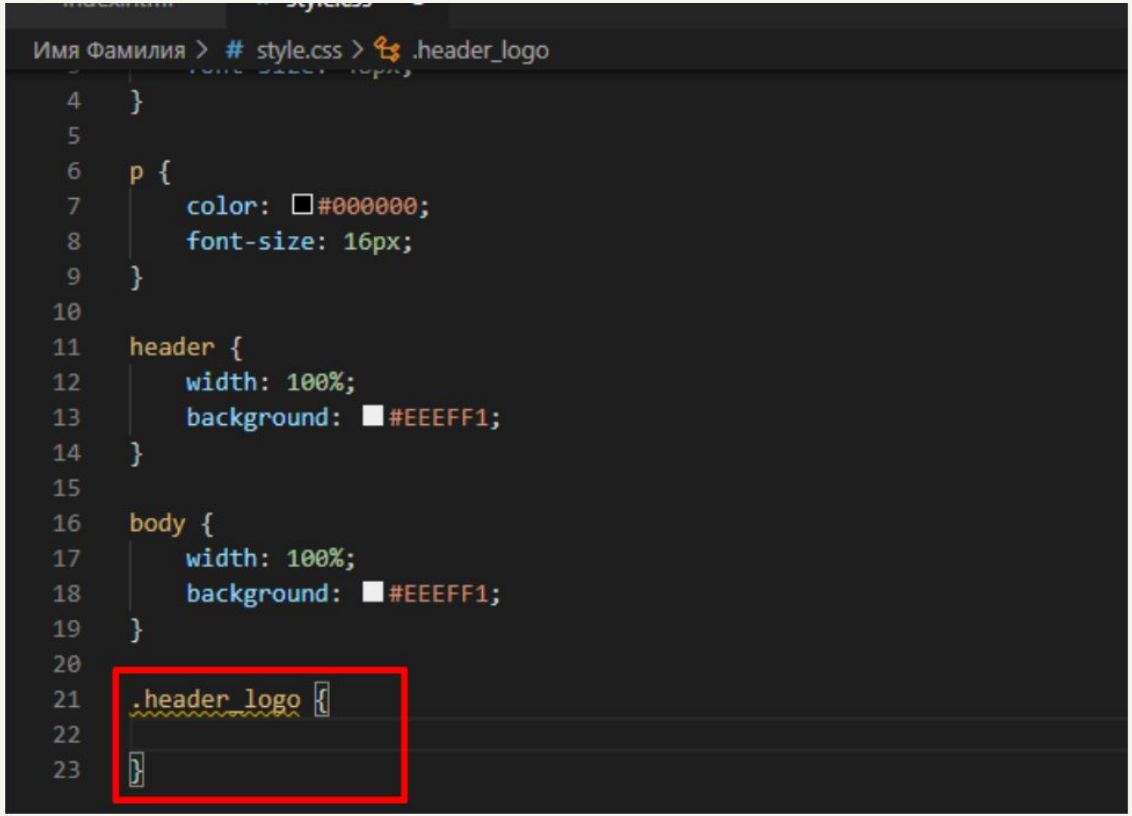


Классы

```
<body>

    <header>
        <div class="header_logo">
            <h1>WD</h1>
        </div>

        <div>
            <p>ГЛАВНАЯ ОБ АВТОРЕ РАБОТЫ ПРОЦЕССЫ КОНТАКТЫ</p>
        </div>

    </header>

    <main>

    </main>

</body>
```

Добавим класс header_logo
к div-элементу логотипа

Классы в CSS



Чтобы обратиться к классу через CSS, ставим перед его названием ТОЧКУ

```
.header_logo{

}
```

Классы в CSS

Смотрим в figma на свойства логотипа и применяем font-family, font-style и font-weight

Дополнительно пишем свой свойство cursor: pointer

```
position: absolute;
width: 96px;
height: 60px;
left: 402px;
top: 50px;

font-family: Revalia;
font-style: normal;
font-weight: normal;
font-size: 48px;
line-height: 60px;
/* identical to box height */
text-align: center;

color: #34547A;
```

```
.header_logo {
    font-family: Revalia;
    font-style: normal;
    font-weight: normal;
    cursor: pointer;
}
```

Работа со шрифтами

font-family - Устанавливает семейство шрифта, которое будет использоваться для оформления текста содержимого

Универсальные семейства шрифтов:

- serif — шрифты с засечками (антиквенные), типа Times;
- sans-serif — рубленные шрифты (шрифты без засечек или гротески), типичный представитель — Arial;
- cursive — курсивные шрифты;
- fantasy — декоративные шрифты;
- monospace — моноширинные шрифты, ширина каждого символа в таком семействе одинакова (шрифт Courier).

Работа со шрифтами

font-style - Определяет начертание шрифта: обычное, курсивное или наклонное.

Значения

- normal - Обычное начертание текста.
- italic - Курсивное начертание.
- oblique - **Наклонное** начертание. Курсив и наклонный шрифт при всей их похожести не одно и то же. Курсив это специальный шрифт имитирующий рукописный, наклонный же образуется путем наклона обычных знаков вправо.
- inherit - Наследует значение родителя.

Работа со шрифтами

font-weight - Устанавливает насыщенность шрифта. Значение устанавливается от 100 до 900 с шагом 100

Значения

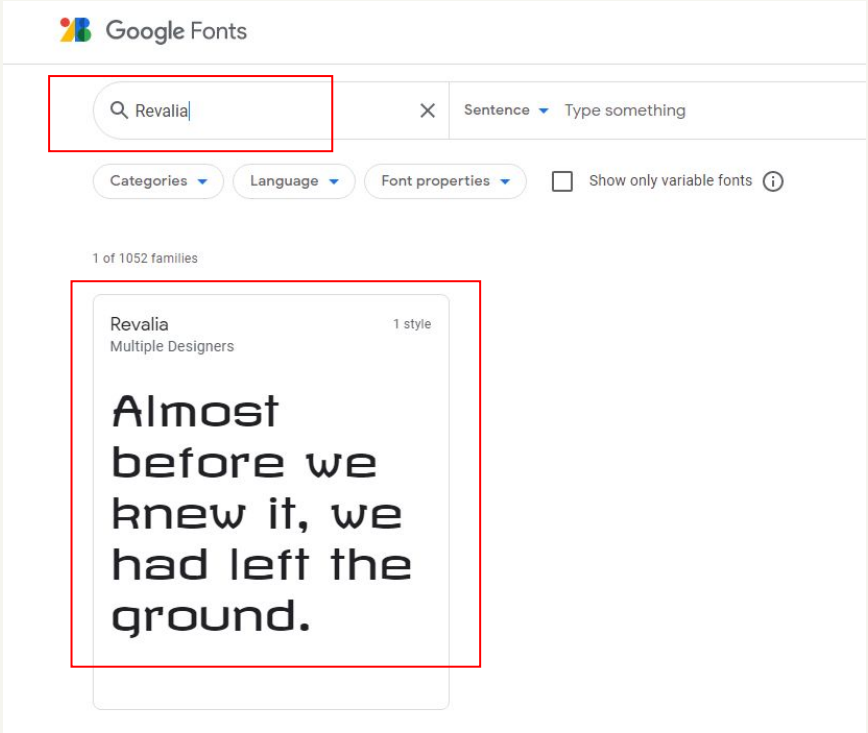
- bold — полужирное начертание
- normal — нормальное начертание.
- Также допустимо использовать условные единицы от 100 до 900

Работа со шрифтами

cursor - устанавливает форму курсора, когда он находится в пределах элемента.

Вид	Значение	Тест	Пример
	default		P {cursor: default}
	crosshair		P {cursor: crosshair}
	help		P {cursor: help}
	move		P {cursor: move}
	pointer		P {cursor: pointer}
	progress		P {cursor: progress}
	text		P {cursor: text}
	wait		P {cursor: wait}
	n-resize		P {cursor: n-resize}
	ne-resize		P {cursor: ne-resize}
	e-resize		P {cursor :e-resize}
	se-resize		P {cursor: se-resize}
	s-resize		P {cursor: s-resize}
	sw-resize		P {cursor: sw-resize}
	w-resize		P {cursor: w-resize}
	nw-resize		P {cursor :nw-resize}

Работа со шрифтами

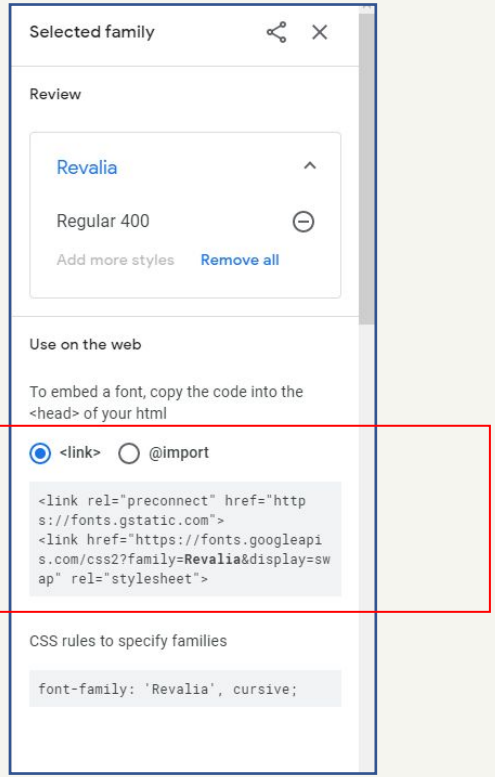


Ищем на сайте Google Fonts шрифт Revalia

Работа со шрифтами

Выбираем regular 400, нажимаем **select this style**

Работа со шрифтами



Копируем ссылку на шрифт и переносим в проект

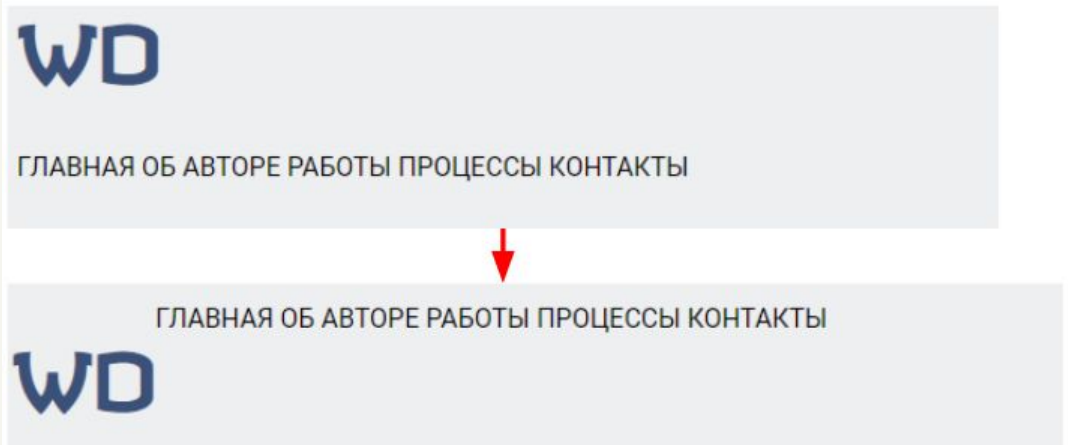
Работа со шрифтами

Добавляем скопированный фрагмент кода в head сайта, сохраняем проект и проверяем на сайте

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/CSS" href="style.css">
        <link href="https://fonts.googleapis.com/css2?family=Revalia&display=swap" rel="stylesheet">
        <title>Сайт-визитка</title>
    </head>

    <body>

        <header>
            <div class="header_logo">
                <h1>WD</h1>
            </div>
```

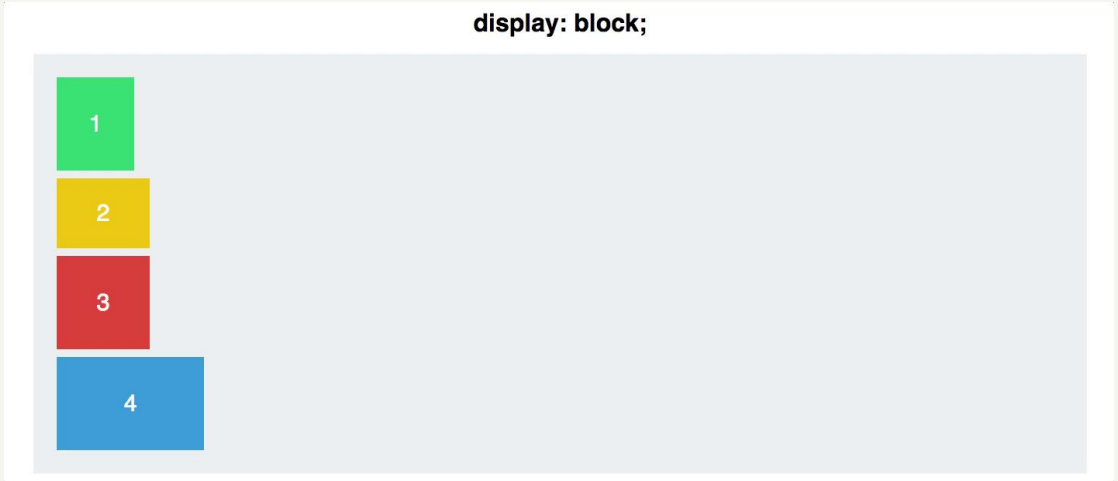
Расположение div элементов

Наше меню должно идти в строчку, после логотипа



Flexbox

Привязка элементов на сайте к линии (горизонтальной \ вертикальной)



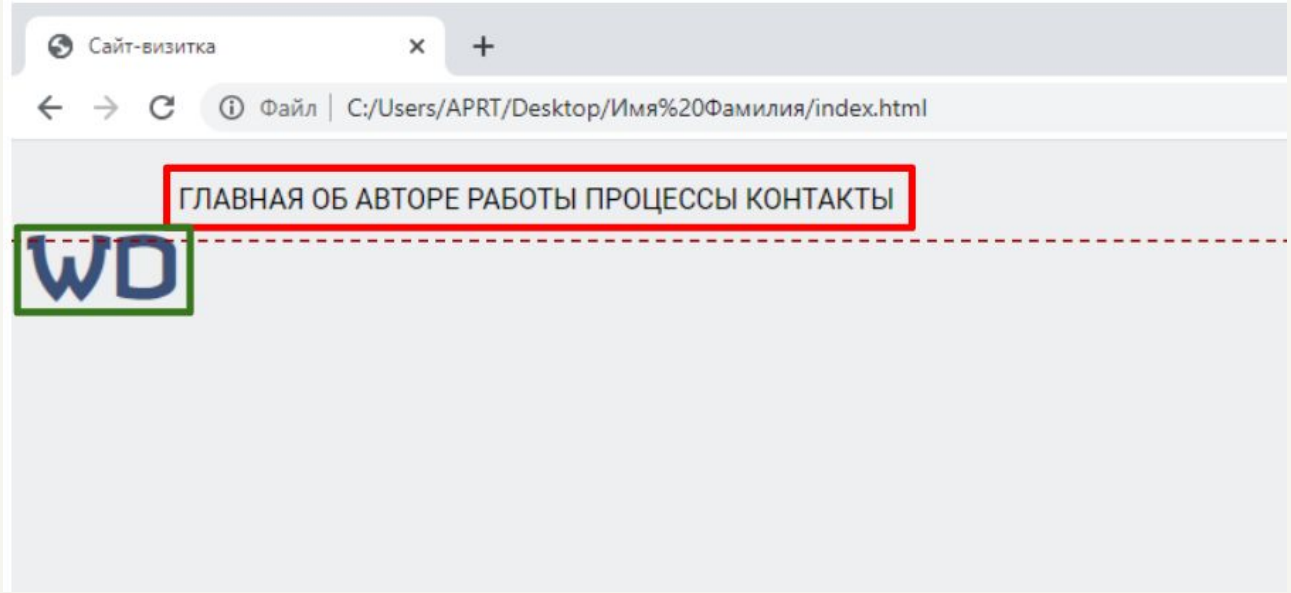
Flexbox

Добавим свойство display в CSS, и установим значение flex - чтобы привязать все элементы к линии (по умолчанию горизонтальной)

```
header {
    width: 100%;
    background: #EEEFF1;
    display: flex;
}
```

Тестируем

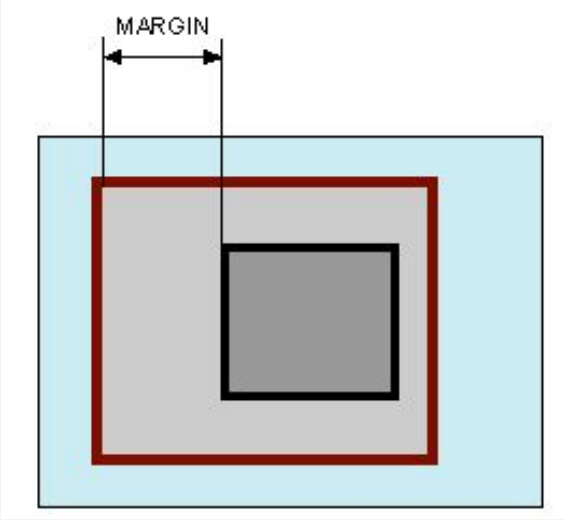
Теперь и логотип и меню привязаны к горизонтальной линии



Свойство margin

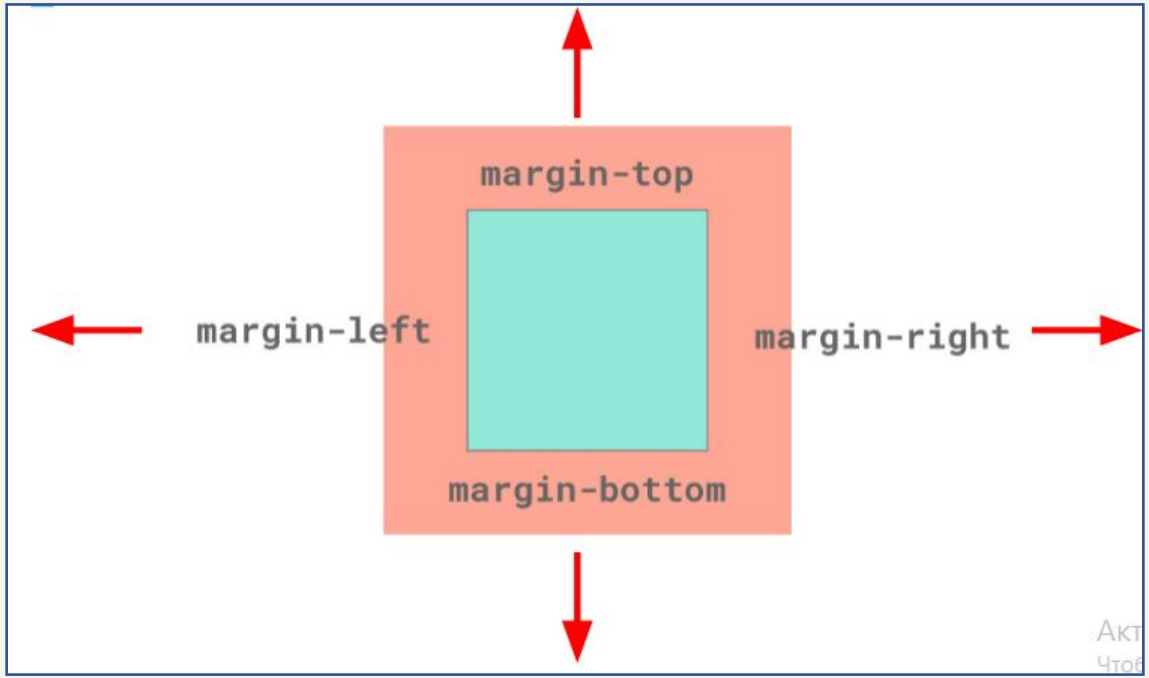
margin - устанавливает величину отступа от каждого края элемента.

Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента



Отступы

Название отступов с каждой из сторон



Задаем отступы для нашего логотипа

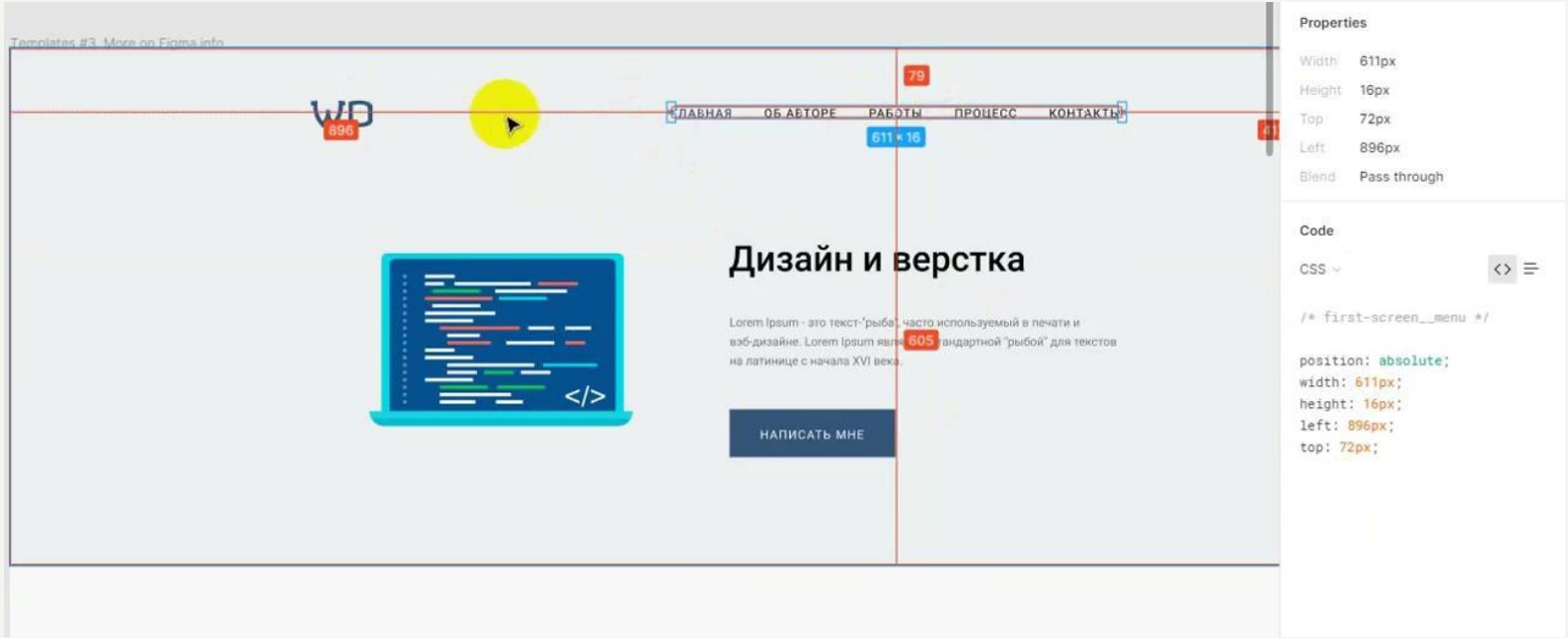
```
.header_logo {
    font-family: Revalia;
    font-style: normal;
    font-weight: normal;
    cursor: pointer;
    margin-left: 402px;
    margin-top: 35px;
}
```

Отступы берем из макета figma

Самостоятельное задание 2

Необходимо добавить отступы для блока меню **header_menu.**

Отступ слева равен 398 px, сверху 79 px



Проверяем

```
.header_menu {
    font-family: Roboto;
    font-style: normal;
    font-weight: normal;
    cursor: pointer;
    margin-left: 398px;
    margin-top: 79px;
}
```

Добавлены два отступа для блока меню

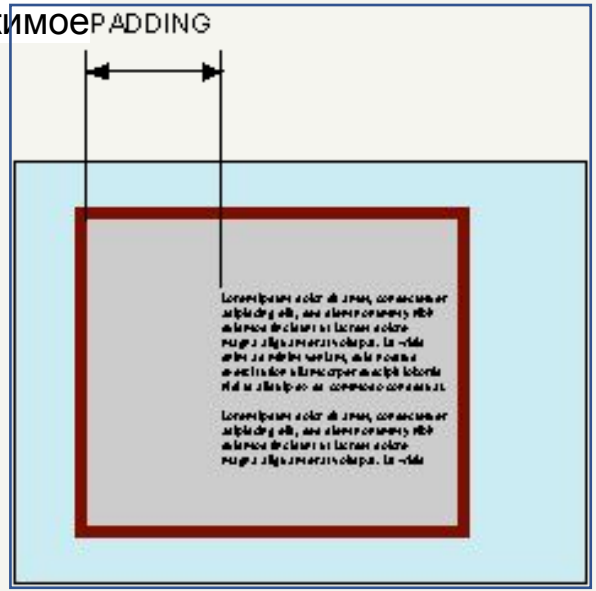
Тег ссылки - <a>Ссылка

```
<header>
    <div class="header_logo">
        <h1>WD</h1>
    </div>

    <div class="header_menu">
        <a>ГЛАВНАЯ</a>
        <a>ОБ АВТОРЕ</a>
        <a>РАБОТЫ</a>
        <a>ПРОЦЕССЫ</a>
        <a>КОНТАКТЫ</a>
    </div>

</header>
```

Обернем наши пункты меню в тег <a>, и сделаем их отдельными ссылками

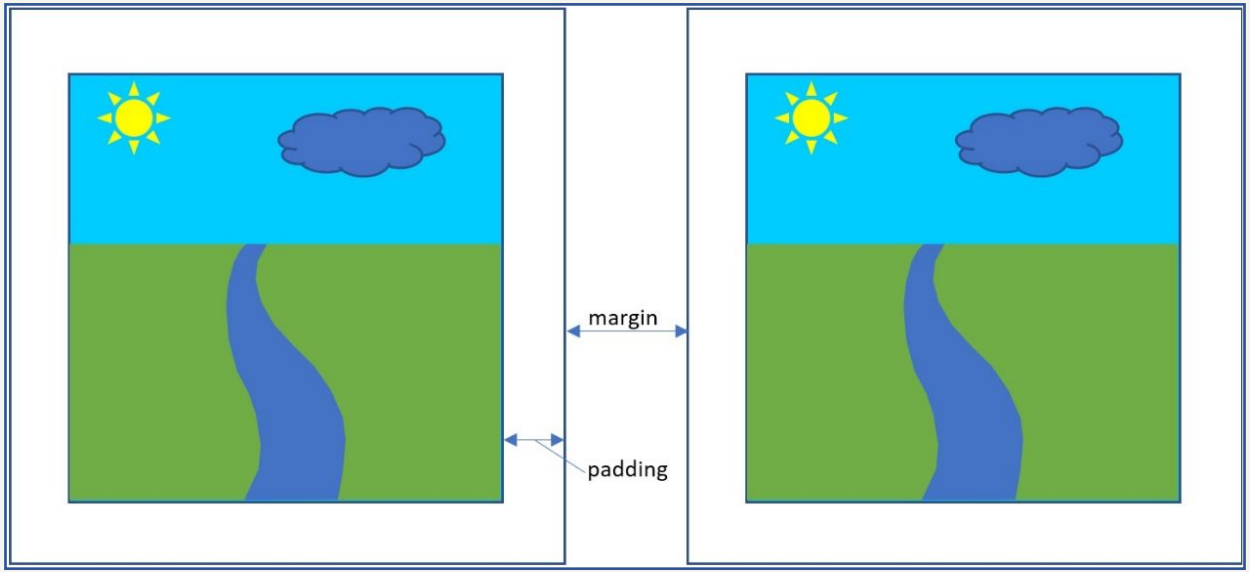
Свойство padding

padding - устанавливает значение полей вокруг содержимого элемента. Полем называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое

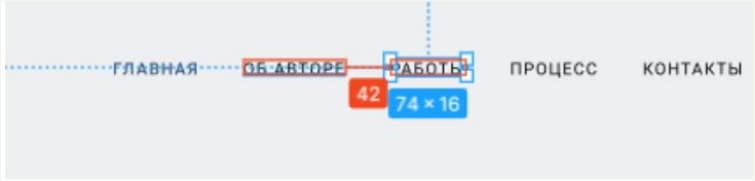


Свойство padding

padding можно сравнить с рамкой у картины



Свойство padding




```
a {
    padding-left: 42px;
}
```

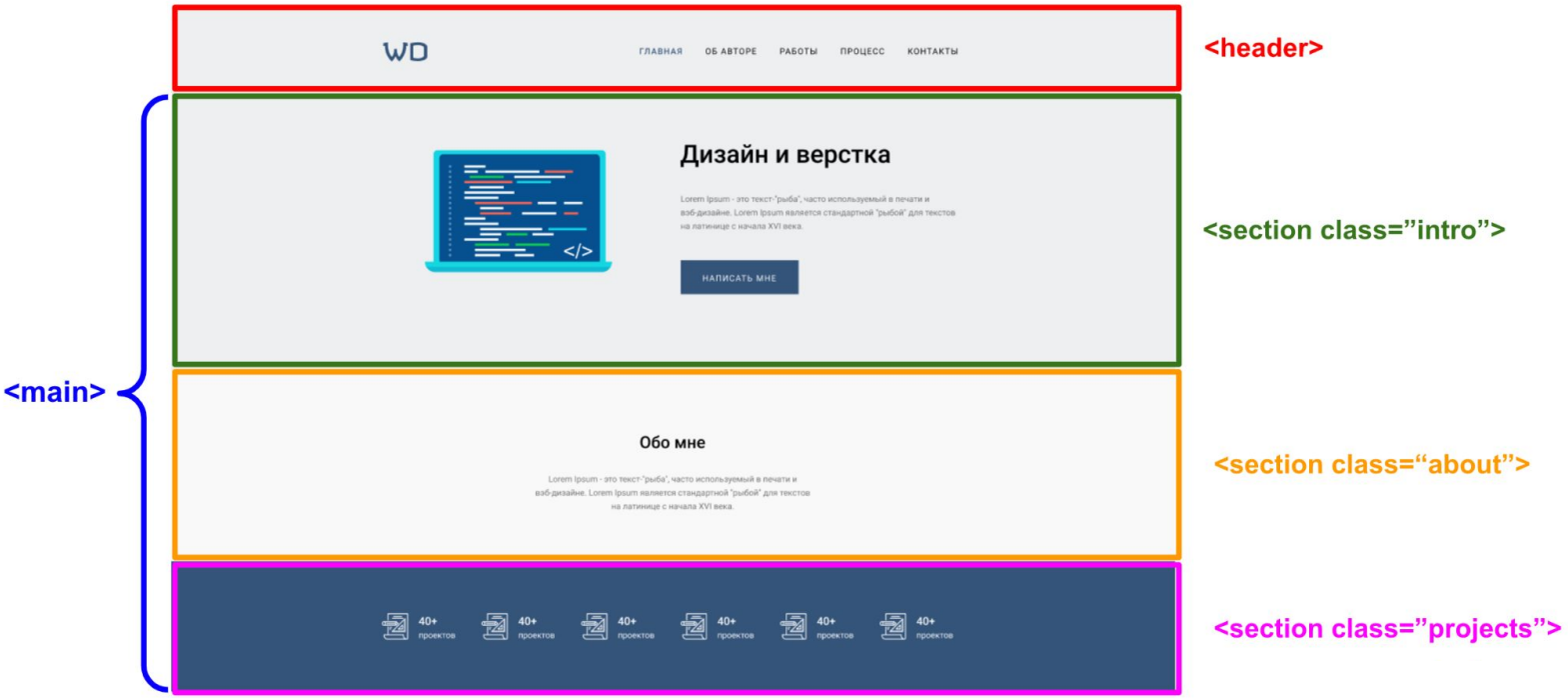
Зададим отступы тегу <a>, в соответствии с макетом

Тестируем на сайте

Теперь у каждого элемента <a> внутренний отступ слева **42 пикселя**

Секции

<section> отвечает за деление страницы на секции

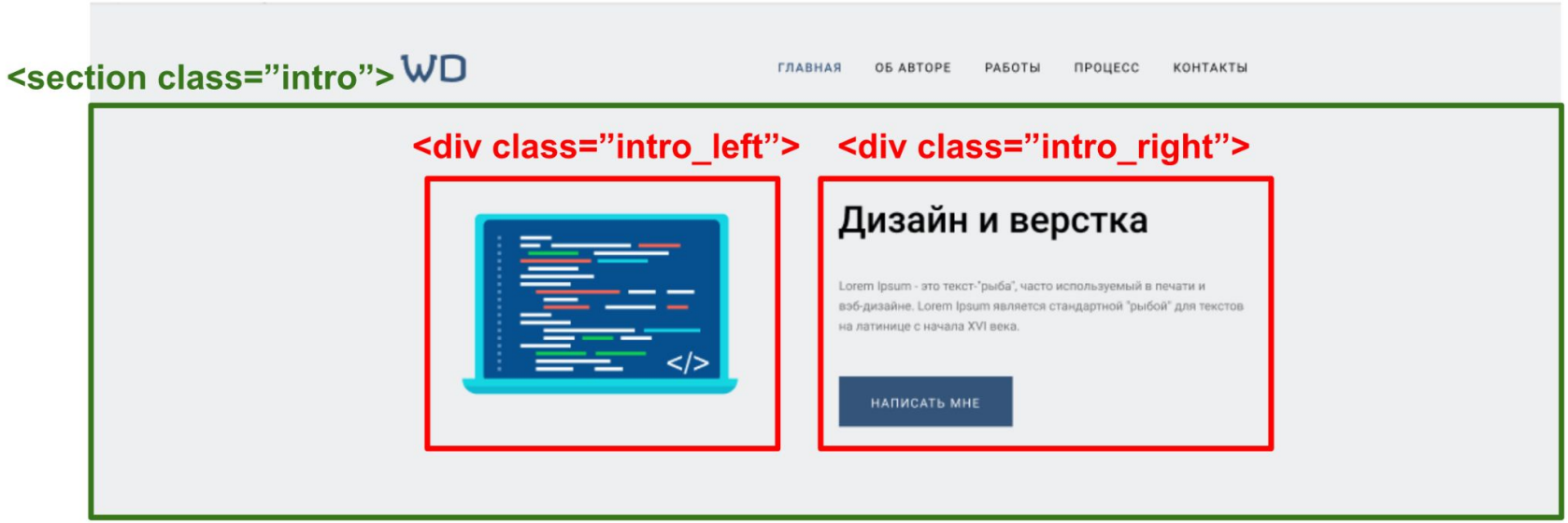


Секции

Внутри main необходимо создать <section> с классом intro.
Самый быстрый способ это сделать - написать section.intro и нажать два раза enter

Секции

Если обратимся к макету, то увидим что секция делится на две части левую и правую (left / right)



Секции

Создадим два контейнера DIV с классом intro_left и intro_right.

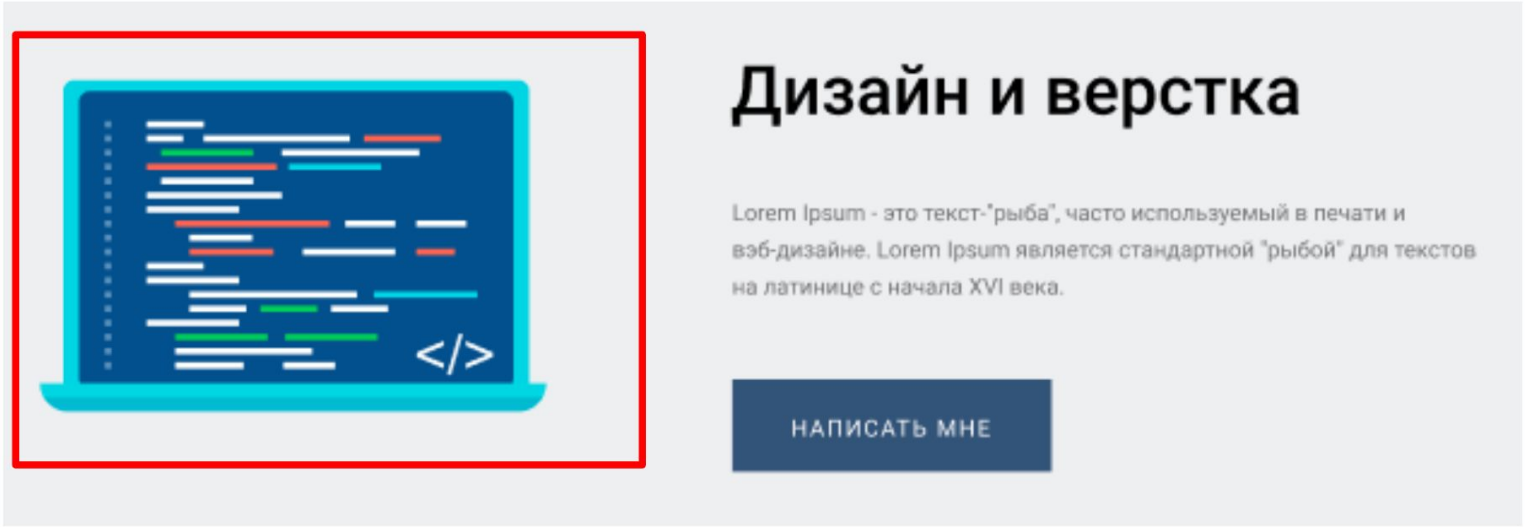
Команда div.intro_left и div.intro_right

После два раза enter

```
<main>
    <section class="intro">
        <div class="intro_left">

        </div>

        <div class="intro_right">

        </div>
    </section>
</main>
```

Секции

В левой части будет располагаться картинка - это уже знакомый нам тег <img/>



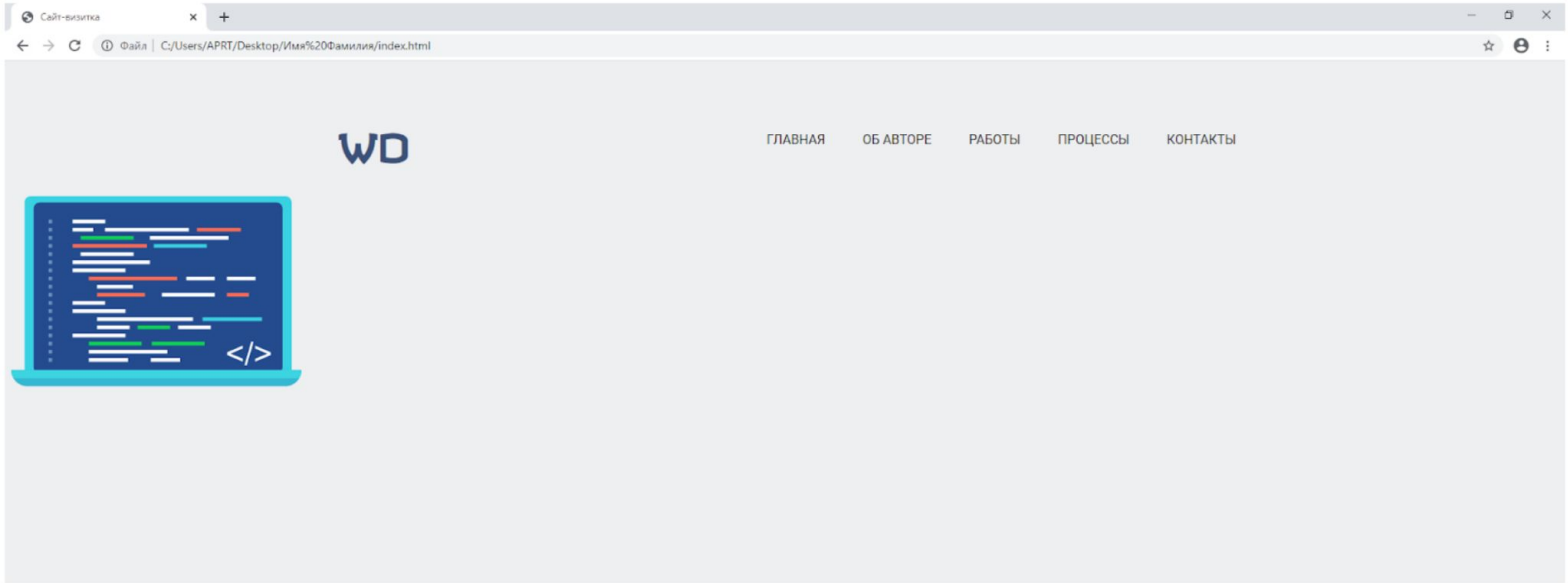

```
<img class="intro_img" src="intro_logo.svg" alt="INTRO">
```

Добавляем картинку на сайт

Внутри контейнера intro_left создаем тег img с классом intro_img

Добавляем картинку на сайт

На сайте должна появиться картинка



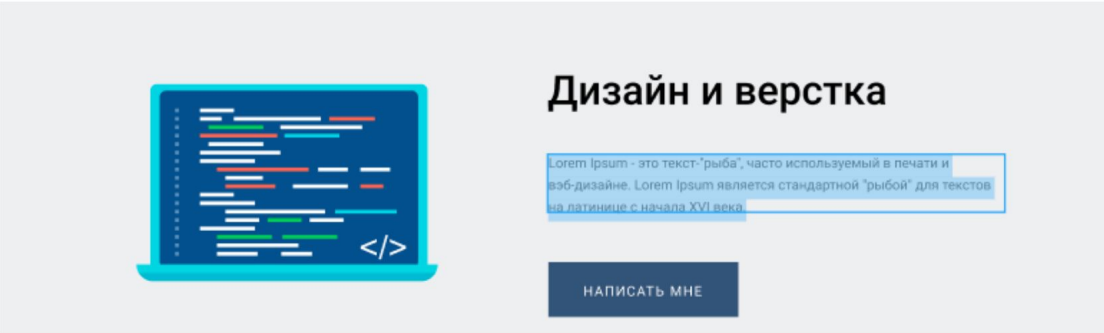
Правая часть intro

**Делаем правую часть сайта, которая состоит из заголовка <h1> </h1> и параграфа **

```
<div class="intro_right">
    <h1 class="intro_title">

    </h1>
    <p class="intro_text">

    </p>
</div>
```

Правая часть intro

Копируем текст из макета фигмы и вставляем на сайт




```
<h1 class="intro_title">
    Дизайн и верстка
</h1>
<p class="intro_text">
    Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.
    Lorem Ipsum является стандартной "рыбой" для текстов на латинице с начала XVI века.
</p>
```

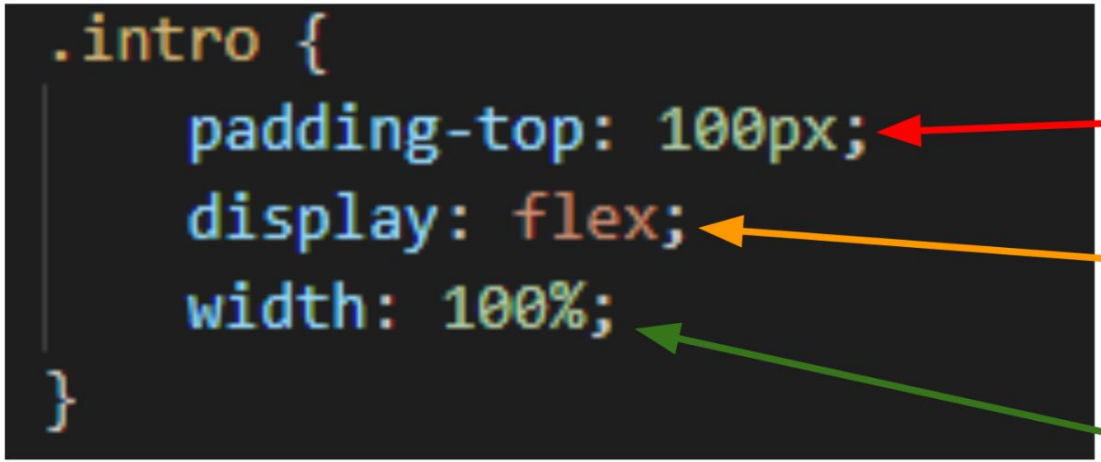
Правая часть intro

Получится примерно так:

Блоки расположились один ниже другого

Работаем с CSS intro

Перейдем в css и напишем следующие стили для класса intro:



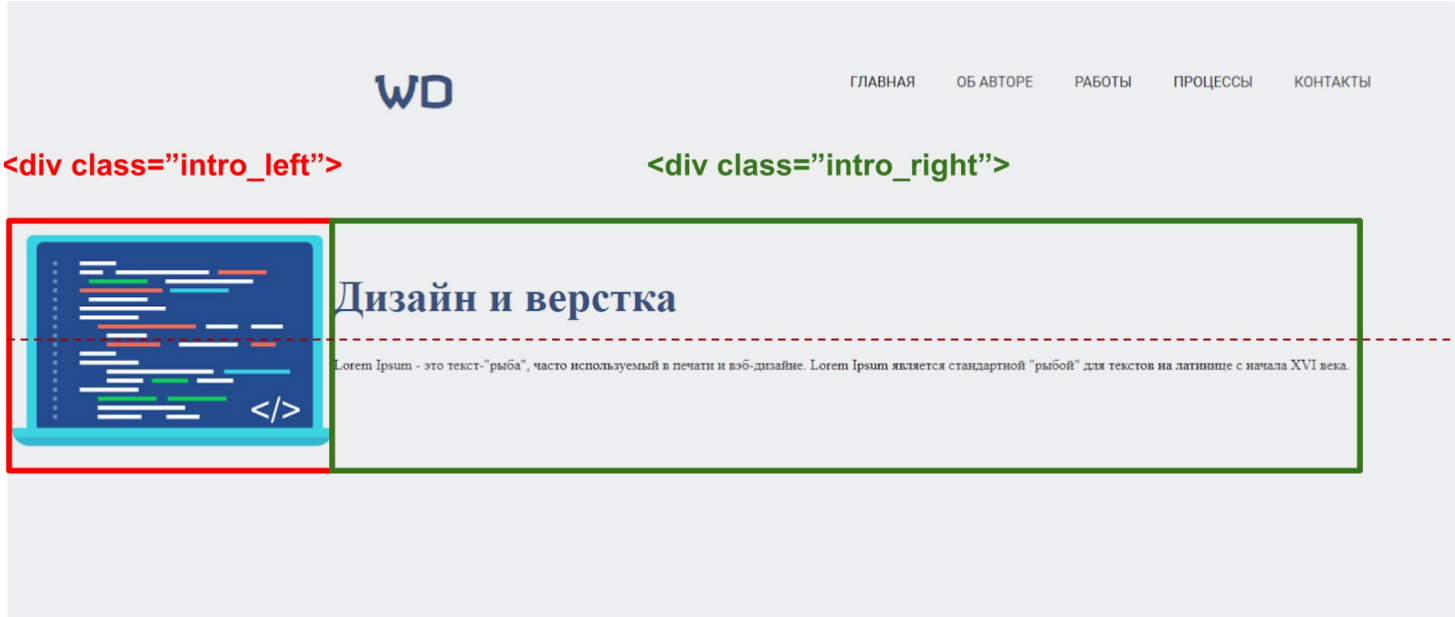
внутренний отступ сверху

расположить блоки вдоль горизонтальной линии

ширина секции - 100%, по всей ширине сайта

Работаем с CSS intro

Блоки расположились друг за другом вдоль оси, но привязаны к левому краю



Кто вспомнит, каким образом мы можем сделать отступ слева?

Работаем с CSS intro

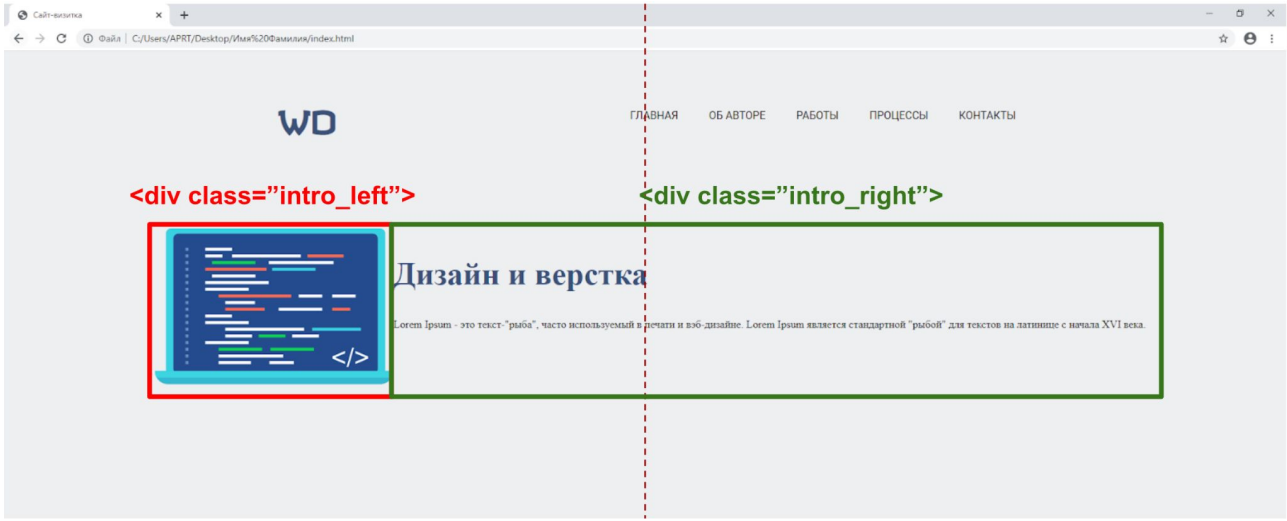
Правильно, мы могли бы сделать отступ, при помощи margin-left, но сегодня изучим следующее свойство:

```
.intro {
    padding-top: 100px;
    display: flex;
    width: 100%;
    justify-content: center;
}
```

Оно центрует содержимое блока (т.е. располагает его по центру экрана)

Работаем с CSS intro

Теперь содержимое блока выровнялось по середине экрана:



Работаем с CSS intro

Зададим ширину и внутренний отступ для левой секции (ширину посмотрим в figma)

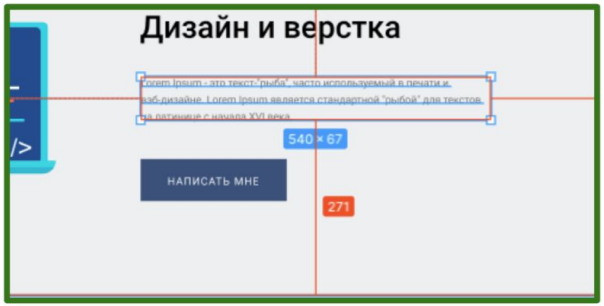
```css
.intro_left {
    width: 357px;
    padding-top: 50px;
}
```

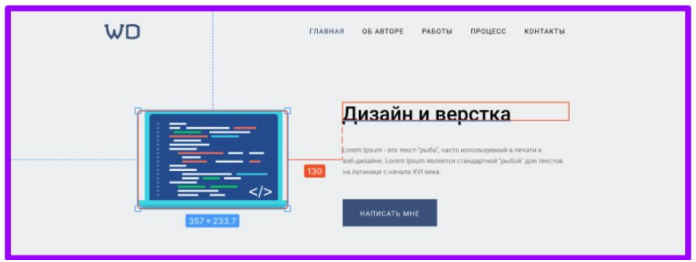
ширина картинки

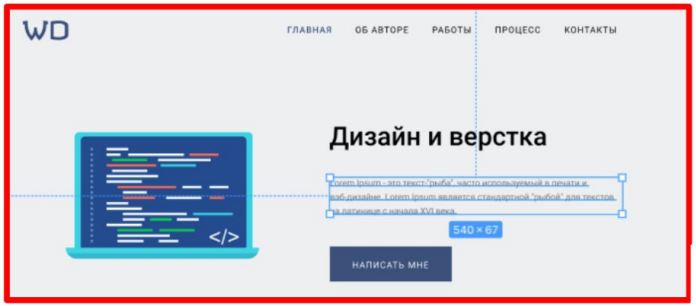
Работаем с CSS intro

Настраиваем отступы для правой части в соответствии с макетом figma

```css
.intro_right {
    max-width: 541px;
    margin-left: 130px;
    margin-bottom: 271px;
}
```

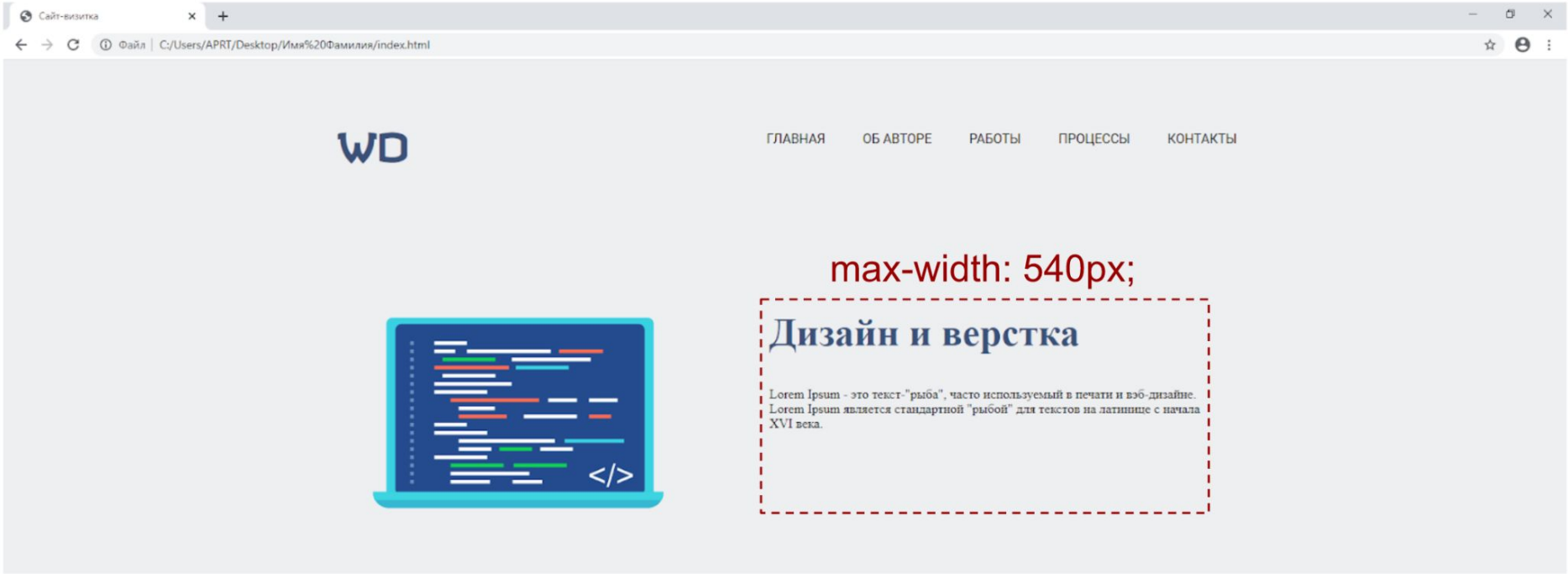






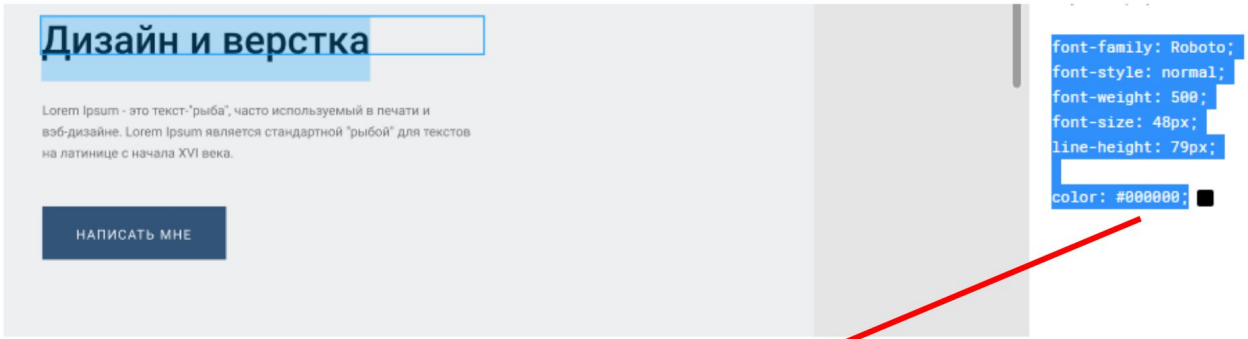
Работаем с CSS intro

Тест сжался до указанной нами максимальной ширины блока, отступы заработали



Работаем с CSS intro

Настраиваем свойство из макета figma для нашего заголовка




```
.intro_title {
    font-family: Roboto;
font-style: normal;
font-weight: 500;
font-size: 48px;
line-height: 79px;

color: #000000;
}
```

Новые свойства

line-height - устанавливает интерлиньяж (межстрочный интервал) текста

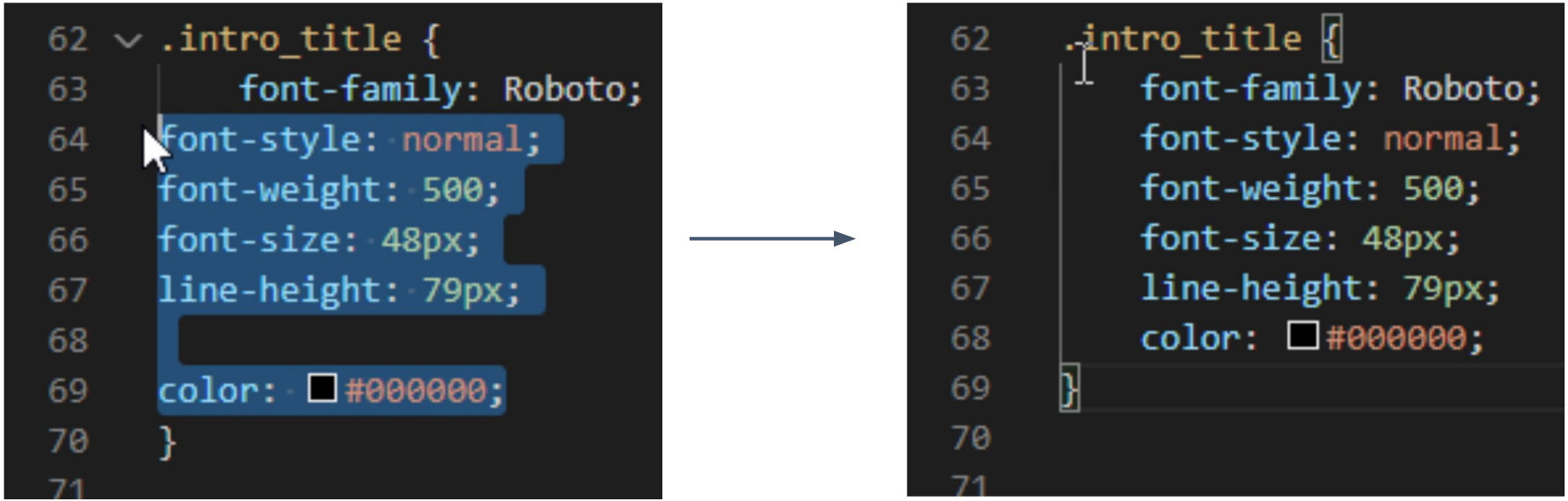
текст текст текст текст текст текст текст текст текст
текст текст текст текст текст текст текст текст текст

текст текст текст текст текст текст текст текст текст

текст текст текст текст текст текст текст текст текст

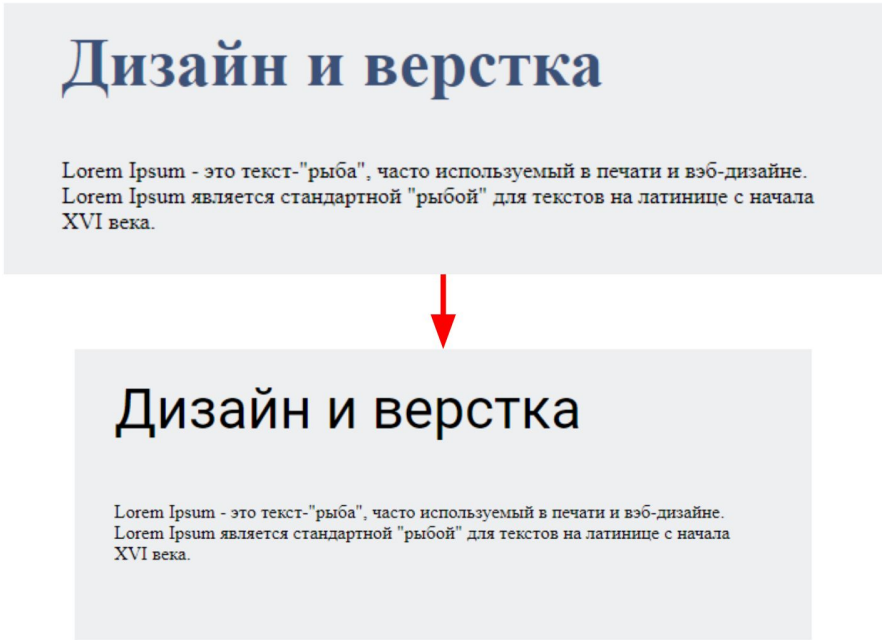
Приводим в порядок

Добавить отступ свойствам можно выделив строки и нажав TAB. Код лучше держать в чистоте



Заголовок готов

Дизайн нашего текст изменился и соответствует макету на 100%



Самостоятельное задание 1

Необходимо установить все свойства из макета для параграфа с текстом

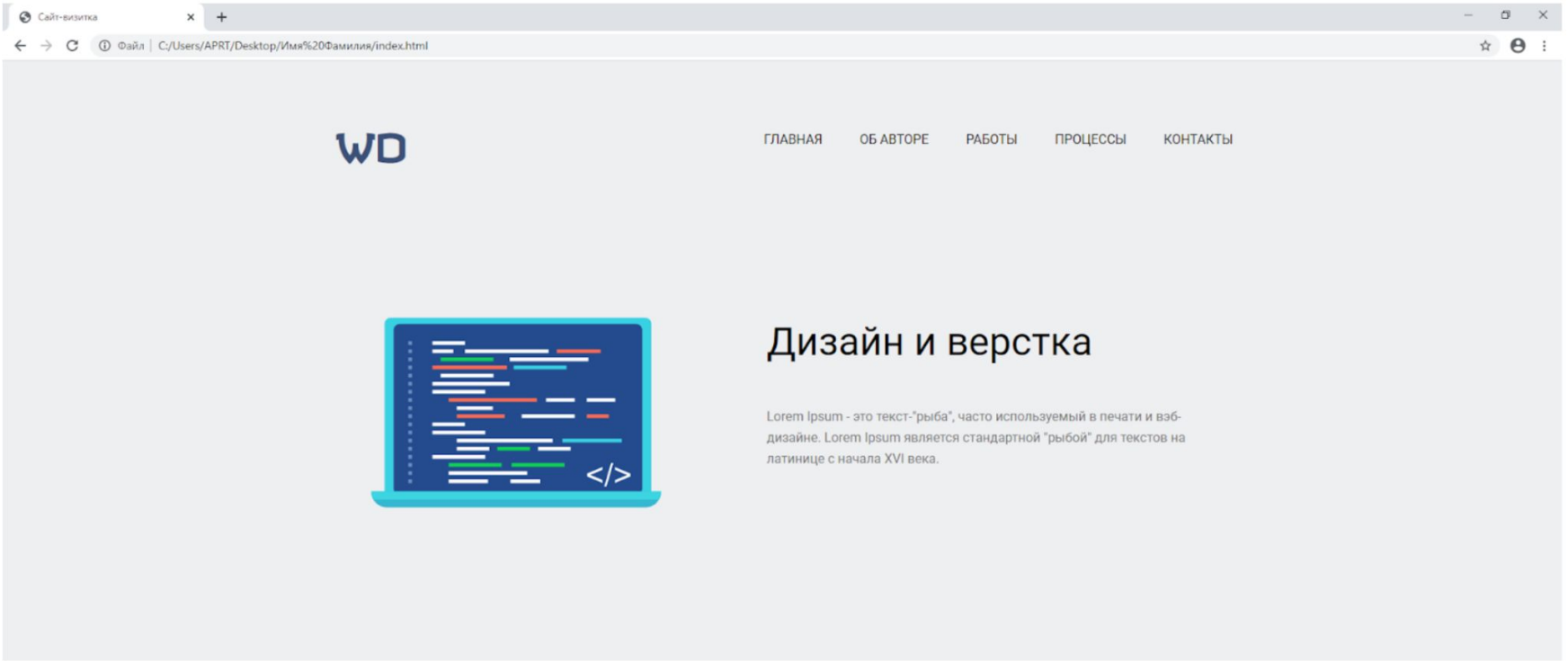
Проверка самостоятельного задания

```css
.intro_text {
    font-family: Roboto;
    font-style: normal;
    font-weight: normal;
    font-size: 16px;
    line-height: 26px;
    color: #727272;
}
```

Дизайн и верстка

Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне. Lorem Ipsum является стандартной "рыбой" для текстов на латинице с начала XVI века.

Обновим страницу



До встречи на следующем уроке