

**Министерство науки и высшего образования Российской  
Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО  
ITMO University**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**По дисциплине Web-программирование**

**Обучающийся Шестаков Максим Олегович**

**Факультет Инфокоммуникационных технологий**

**Группа K3321**

**Направление подготовки 11.03.02 Инфокоммуникационные  
технологии и системы связи**

**Образовательная программа  
Программирование в инфокоммуникационных  
системах**

<b>Обучающийся</b>	<u>16.11.2024</u> (дата)	<u>                    </u> (подпись)	<u>Шестаков М.О.</u> (Ф.И.О.)
<b>Руководитель</b>	<u>                    </u> (дата)	<u>                    </u> (подпись)	<u>Марченко Е.В.</u> (Ф.И.О.)

Санкт-  
Петербург  
2024 г.

# СОДЕРЖАНИЕ

Стр.

<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>1   Ход работы .....</b>	<b>4</b>
1.1   Задание №1 .....	4
1.2   Задание №2 .....	4
1.3   Задание №3 .....	5
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>7</b>

## ВВЕДЕНИЕ

**Цель работы:** Изучение инструмента gulp

# 1 Ход работы

## 1.1 Задание №1

В первом задании необходимо было установить git. Для сдачи лабораторной работы №1 уже использовался git-репозиторий, поэтому данное задание не будет рассматриваться в отчёте.

## 1.2 Задание №2

Во втором задании требовалось установить gulp. Для этого сначала нужно было установить node, npm и npx. После их установки gulp был успешно загружен.



Рисунок 1.1 — Версия Gulp

После установки gulp была создана первая task. Для создания тасок сначала импортируется модуль в переменную gulp, после чего создаётся сама задача. Первая задача была создана для примера, она выводит приветственное сообщение. Вторая задача, выполняющаяся по умолчанию, создаёт сервер. Для создания сервера сначала импортируется модуль, после чего сразу же используется метод create(), который и иницирует сервер. Данная программа отслеживает изменения в html-странице index.html и стилевом файле second.css, реагируя на них автоматической перезагрузкой страницы.

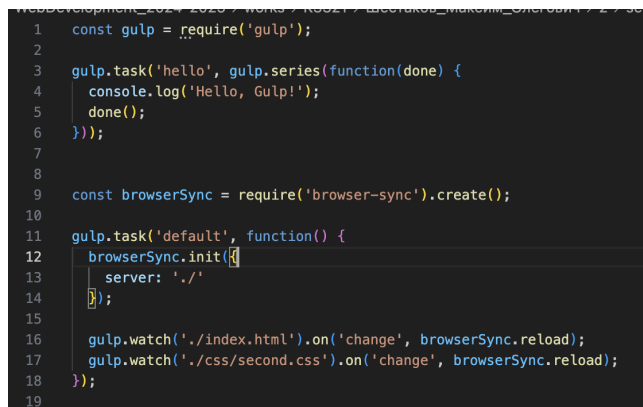


Рисунок 1.2 — Task

### 1.3 Задание №3

В третьем задании необходимо было написать программу-клиент, которая показывала бы с определенной периодичностью web-страницы.

Для выполнения использовались наработки gulpfile из второго задания, к отслеживанию добавились файлы js и json. В файле pages.json содержатся url страниц и интервалы их показа в мс.

```
1  {
2    {
3      "url": "https://ploshadka.net/ustanovka-gulp-na-mac-os/",
4      "interval": 5000
5    },
6    {
7      "url": "https://hcdev.ru/",
8      "interval": 3000
9    },
10   {
11     "url": "https://gulpjs.com/docs/en/getting-started/creating-tasks/",
12     "interval": 4000
13   }
14 }
```

Рисунок 1.3 — Pages

В JS-скрипте с помощью метода fetch данные из файла со страницами преобразуются в нужный формат, после чего производится обработка показа. В исходной html-странице с заданными интервалами меняется src-адрес веб-страницы, это действие повторяется по циклу.

```
1  let pages;
2
3  fetch('pages.json')
4    .then(response => response.json())
5    .then(data => {
6      pages = data;
7      changePage();
8    });
9
10 function changePage() {
11   const frame = document.getElementById('frame');
12   let pageIdx = 0;
13   function showPage() {
14     frame.src = pages[pageIdx].url;
15     setTimeout(() => {
16       pageIdx = (pageIdx + 1) % pages.length;
17       showPage();
18     }, pages[pageIdx].interval);
19   }
20   showPage();
21 }
```

Рисунок 1.4 — JS-скрипт

В html-странице для реализации смены веб-страниц используется элемент `iframe`, по его `id` (`frame`) мы определяем константу `frame` в JS-скрипте для дальнейшей работы с ней.

```
9   <body>
10   <h1>Third sync task: opening pages from the list</h1>
11   <iframe id="frame"></iframe>
12   <script src="js/third.js"></script>
13 </body>
```

Рисунок 1.5 — `iframe`-элемент

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения лабораторной работы был изучен gulp и возможности его применения в веб-разработке.