OTUS

JavaScript Developer. Basic

Разработчик JavaScript. Базовый уровень

Длительность курса: 198 академических часов

1 Базовый JavaScript. Работа с GIT и настройка окружения

1 Введение в курс + основы JS и практика

Цели занятия:

описать процессы: как задавать вопросы, как сдавать домашки, какие требования, использование редакторов + типовой подход к работе с домашками (ветка -> пуллреквест -> ревью).

Краткое содержание:

google;
stackoverflow;
github issues;
slack.

Домашние задания

1 Знакомство

Цель: https://github.com/vvscode/otus--javascript-basic/blob/master/lessons/lesson01/ht.md

https://github.com/vvscode/otus--javascript-basic/blob/master/lessons/lesson01/ht.md

---

2 Базовый синтаксис JavaScript: основы, базовые операции, ветвления и базовая алгоритмизация

Цели занятия:

писать базовые скрипты на javascript;
создавать программы обрабатывающие ввод от пользователя и делать вывод на странице.

Краткое содержание:

переменные;
циклы;
условные ветвления;
функции;
объекты;
ввод / вывод.

---

3 **Объектная модель в Javascript**

Цели занятия:

поработать с объектами, их свойствами, понятие прототипа.

Краткое содержание:

объекты как ссылочный тип данных;
proto;
работа с proto;
конструирование объектов с помощью Object.create.

Домашние задания

1 JS-коаны - решение типовых задач на использование базового синтаксиса

Цель: По итогам занятий вы будете свободнее оперировать базовыми операциями на языке JavaScript.

В ходе выполнения задания вы поработаете с двумя репозиториями, которые содержат в себе типичные задачи, с которыми сталкивается любой разработчик каждый день, потренируете навыки поиска ответов (и задавания вопросов) и сдадите свою первую домашнюю работу с кодом

Работать вы будете с двумя репозиториями

https://github.com/mrdavidlaing/javascript-koans

и

https://github.com/liammclennan/JavaScript-Koans

Вам нужно для каждого из репозиториев выполнить следующие шаги
- скачать (склонировать или просто скачать) к себе репозиторий
- разобраться как запустить задачи (которые оформлены в виде тестов)
- внести изменения так, чтобы решения задач проходили тесты
- сделать коммит и пуллреквест с решением
- сделать скриншот всех пройденых тестов

Задание можно сдавать в формате
- скришноты пройденых тестов

или

- скриншоты пройденых тестов и ссылка на пуллреквест с вашими изменениями (работу с гитом мы еще не разбирали, так что это задание со звездочкой)

---

4 **Контекст при работе с функциями**

Цели занятия:

разобраться с понятием контекста;
разобрать, чем определяется значение this;
разобрать способы управления this.

Краткое содержание:

this;
this внутри callback функций;
изменение и фиксация контекста;
стрелочные функции.

---

5 **Прототипное наследование и функции-конструкторы**

Цели занятия:

научиться использовать прототипы для уменьшения дублирования кода;
разобраться с работой конструкторов с точки зрения прототипов.

Краткое содержание:

прототипы;
конструкторы.

Домашние задания

1 Практика кодирования

Цель: Целью домашнего задания является закрепление навыков написания кода, а также получение опыта работы в среде описанной тестами.

1. Зарегистрировать аккаунт на сайте codewars.com (это можно сделать с помощью github аккаунта) - логин должен быть такой же как в вашем github профиле, имя и фамилия такие же как в личном кабинете ОТУС
2. Решить 19 задач из списка ниже
3. Убедиться, что в вашем профиле видно когда, сколько и какие задачи вы решили
3. Ссылку на ваш профиль codewars сбросить в чат по дз

Задачи для решения:
String
8 kyu https://www.codewars.com/kata/palindrome-strings
7 kyu https://www.codewars.com/kata/anagram-detection
7 kyu https://www.codewars.com/kata/disemvowel-trolls/
7 kyu https://www.codewars.com/kata/mumbling/
7 kyu https://www.codewars.com/kata/highest-and-lowest/
7 kyu https://www.codewars.com/kata/isograms/
7 kyu https://www.codewars.com/kata/char-code-calculation
7 kyu https://www.codewars.com/kata/cat-and-mouse-2d-version/

Number
8 kyu https://www.codewars.com/kata/opposite-number
8 kyu https://www.codewars.com/kata/even-or-odd
8 kyu https://www.codewars.com/kata/century-from-year
7 kyu https://www.codewars.com/kata/greatest-common-divisor
7 kyu https://www.codewars.com/kata/factorial
7 kyu https://www.codewars.com/kata/squares-sequence
7 kyu https://www.codewars.com/kata/concatenated-sum
7 kyu https://www.codewars.com/kata/filter-the-number

Array
8 kyu https://www.codewars.com/kata/removing-elements
8 kyu https://www.codewars.com/kata/remove-

duplicates-from-list
8 kyu https://www.codewars.com/kata/sum-of-positive

---

6 **Базовое использование API и JavaScript. Как работать с DOM и другими доступными API (работа с сервером, с хранилищем и т.п)**

Цели занятия:

научиться взаимодействовать со страницей;
разобрать общие принципы и примеры использования API в разработке - загрузить данные со стороннего ресурса и отобразить их на странице.

Краткое содержание:

api;
DOM;
fetch;
localstorage;
createElement.

---

7 **Тестирование кода как часть процесса разработки, пример применения тестирования к домашним заданиям**

Цели занятия:

научиться тестировать код.

Краткое содержание:

testing;
tdd;
jest;
mock;
PR pipeline.

Домашние задания

1 Практика кодирования

Цель: Целью домашнего задания является закрепление навыков написания кода, а также получение опыта работы в среде описанной тестами.

1. Зарегистрировать аккаунт на сайте codewars.com (это можно сделать с помощью github аккаунта) - логин должен быть такой же как в вашем github профиле, имя и фамилия такие же как в личном кабинете ОТУС

2. Решить 16 задач из списка ниже (вместе с прошлыми должно выйти 35)
3. Убедиться, что в вашем профиле видно когда, сколько и какие задачи вы решили
3. Ссылку на ваш профиль codewars сбросить в чат по дз

Задачи для решения:

Array
7 kyu https://www.codewars.com/kata/find-the-capitals
7 kyu https://www.codewars.com/kata/shortest-word
7 kyu https://www.codewars.com/kata/square-every-digit
7 kyu https://www.codewars.com/kata/playing-with-sets-intersection
7 kyu https://www.codewars.com/kata/divide-and-conquer
6 kyu https://www.codewars.com/kata/find-the-odd-int/
6 kyu https://www.codewars.com/kata/find-the-parity-outlier
6 kyu https://www.codewars.com/kata/zipwith

String
6 kyu https://www.codewars.com/kata/duplicate-encoder

Number
6 kyu https://www.codewars.com/kata/n-th-fibonacci

Date
7 kyu https://www.codewars.com/kata/it-is-written-in-the-stars
6 kyu https://www.codewars.com/kata/friday-the-13ths

Object
7 kyu https://www.codewars.com/kata/counting-array-elements
7 kyu https://www.codewars.com/kata/who-is-the-killer-1

Regular expression
8 kyu https://www.codewars.com/kata/simple-validation-of-a-username-with-regex
6 kyu https://www.codewars.com/kata/validate-my-password

8 **Выбор темы и организация проектной работы**

Цели занятия:

выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом.

Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;
требования к результату проекта и итоговой документации.

Домашние задания

1 Проектная работа

Цель: Выбрать тему проекта;
Закрепить тему в чате по дз;
Реализовать собственный проект с применением Javascript, который вы сможете использовать в своем портфолио.

Проект должен включать в себя
- работу со сторонним api
- одностраничное приложение (если речь идет о Frontend)
- должно поддерживаться сохранение пользовательских данных (история, прошлые операции и тп)
- код должен быть покрыт модульными и интеграционными тестами

Проект должен быть:
- выложен на github
- задеплоен на публичный сервис, ссылка прикреплена к репозиторию проекта
- в репозитории должны быть - расширенный README, настроен CI/CD, понятная история коммитов (семантические коммиты)

Для защиты используйте шаблон презентации (https://docs.google.com/presentation/d/1ugjXX1yX400tUXxBIZaLgiwlJwpl3EzQ5fF2sp3wXPY/edit?usp=sharing)
На защиту вашего проекта дается 15 минут, более

подробная информация дана на вебинаре к этому занятию
Проект сдавать через чат с преподавателем

Вопросы задавать руководителю курса.

---

9 **Использование СКВ: проблемы и решения, типовой порядок разработки с использованием Git**

Цели занятия:

работать с git/github;
объяснить и использовать feature-branch подход,
работать с пуллреквестами.

Краткое содержание:

git;
gitignore;
PR;
review;
resolve conflicts;
github actions;
настройки репозитория;
автодеплой (с примером codesandbox).

Домашние задания

1 Закрепление базового синтаксиса языка

Цель: В задании тренируются навыки:
- использование базовых аспектов языка (использование переменных, функций и тп)
- выполнения заданий на курсе

В задании вы сможете:
- познакомиться с первичной настройкой окружения
- отточить базовые конструкции языка

Вам нужно будет:
- создать репозиторий на гитхабе
- инициировать проект в репозитории
- решить предложенные задачи
- сделать коммит (а лучше несколько - по одному на задание)
- открыть пуллреквест
- прислать ссылку на пуллревест в чат по дз

10	**Код как «проект» - артефакты работы разработчика. Зависимости и утилиты в стеке JavaScript**	**Цели занятия:** создавать nodejs проект; устанавливать зависимости, различать обычные и dev-зависимости, использовать cli утилиты из npmjs. **Краткое содержание:** npm project; package.json; dependencies/ devdependencies; npm vs npx.
11	**Консультация. Разбор сложных моментов в выполнении домашних заданий. Пример применения TDD в практике разработки**	**Цели занятия:** получить ответы на вопросы. **Краткое содержание:** tdd; rgr цикл.
12	**Итоги по синтаксису JS. Что делать, если что-то не работает? Где искать документацию и помощь. Решение вопросов по текущим домашним заданиям**	**Цели занятия:** проверить знание js на основе списка задач с собеседований; ответить на возникающие вопросы. **Краткое содержание:** debug; console / dev tools; caniuse.

13 **Работа с асинхронным кодом**

Цели занятия:

разобрать тему асинхронности и неблокирующего выполнения кода.

Краткое содержание:

stack;
queue;
callbacks;
promise;
async / await.

---

14 **Современный инструментарий при разработке клиентских (и не только приложений)**

Цели занятия:

разобрать и посмотреть на примере какие инструменты упрощают жизнь разработчика:
- дев-сервер;
- хот-релоад;
- дебаггер;
- browsersync;
- github pages.

Краткое содержание:

webpack;
reload;
browsersync;
githubpages.

Домашние задания

1 Приложение "Прогноз погоды"

Цель: В задании тренируются навыки:
- работы с тестовыми системами
- использование базового синтаксиса js
- публикация кода с помощью сервиса githubpages

Во время выполнения задания вы:
- примените знания по созданию базовых страниц для отображения информации
- познакомитесь с основными инструментами, упрощающими жизнь разработчика

1. Создайте страницу:

1.1 при открытии страницы пользователь видит погоду (город, температуру и иконку) в своей местности (для получения прогноза погоды используйте https://openweathermap.org/current)
1.2 он может ввести имя города в поле ввода и увидеть погоду в выбранном городе
1.3 введенные города сохраняются у пользователя в браузере, так что он видит последние 10 городов, где он смотрел погоду
1.4 при клике по строчке города в списке он видит погоду в выбранном городе
1.5 кроме информации о погоде покажите в центре страницы карту для введенного адреса (используйте Google Maps Statis API https://developers.google.com/maps/documentation/maps-static/start)

2. Проверить покрытие кода тестами, и добавить проверку покрытия при запуске `test` скрипта. Покрытие должно быть не ниже 60%

---

15 **Консультация**

Цели занятия:

получить ответы на вопросы.

Краткое содержание:

решение задач.

2 Создание и стилизация HTML страниц

1 Структура HTML документа, семантика и основы разметки страницы

Цели занятия:

создавать новые документы в соответсвии с семантическими правилами;
объяснить основные типы тегов и атрибутов.

Краткое содержание:

semantic markup;
html attributes.

2 **Стилизация страницы, позиционирование элементов**

Цели занятия:

рассмотреть способы стилизации элементов, особенности использования стилевых таблиц; поработать с селекторами и стилевыми правилам.

Краткое содержание:

css;
layout technics;
selector;
selector priority.

Домашние задания

1 Создание сложной страницы с использованием семантических элементов html5, стилизация ее в 2/3 - колоночный макет

Цель: HTML является неотъемлемой частью работы JS-разработчика. В этом задании вы закрепите знания о структуре html документа и создадите один из самых часто-используемых шаблонов - трехколоночный макет. (https://upload.wikimedia.org/wikipedia/commons/5/50/Трёхколоночный_типовой_веб-макет.png)

В задании тренируются навыки:
- создание html страниц
- стилизация html элементов

Вам нужно:
- Создать новый github репозиторий
- подключить к нему линтеры и actions разобранные ранее
- сверстать трехколоночный макет, с использованием семантической разметки
- сделать пуллреквест для отображения изменений
- к описанию пуллреквеста приложить ссылку на просмотр страницы (с использованием rawgithack)"

3 **Инструменты разработки. Тестирование верстки, использование медиазапросов**

Цели занятия:

настроить локальный HTTP сервер, browsersync;
использовать puppeteer для тестирования скриншотами;
использовать медиазапросы для адаптивной верстки.

Краткое содержание:

node-static;
browser-sync;
adaptive design;
mobile first.

4 **Основные подходы к позиционированию элементов. Основные проблемы и решения при работе со стилями, модульность**

Цели занятия:

разобраться в том, какие есть варианты для создания раскладок - от таблиц для flexbox и гридов;
объяснить про назначение reset/normalize, понимать причины возникновения BEM/css in js.

Краткое содержание:

tables;
float;
flexbox;
grid.

Домашние задания

1 Использовать препроцессор для создания страниц приложения "блог" - сверстанные страницы должны быть доступны на githubpages

Цель: В результате выполнения ДЗ вы создадите верстку для вашего личного блога с использование препроцессора SASS(/LESS)

В данном задании тренируются навыки:
- верстки страниц
- использование принципов DRY при верстке
- использование препроцессоров
- деплой страниц на сервис github pages

Необходимо:
- создать новый репозиторий
- настроить в нем линтеры, github actions
- включить проверку линтером при PR
- включить деплой приложения при PR
- создать сайт-блог с минимум 4мя страницами (главная, обратная связь, список записей, страница записи)
- создать пуллреквест с изменениями и
- сбросить ссылку на PR и ссылку на задеплоенные страинцы в част с преподавателем

3 Применение JavaScript/Typescript для создания интерактивных страниц

1 Создание подключаемых плагинов

Цели занятия:

объяснить, как устроены подключаемые плагины (например. bootstrap);
разобрать подходы к разработке улучшателей пользовательского опыта.

Краткое содержание:

подключение js на страницу;
декларативное и императивное использование плагинов;
плагины / сторонние скрипты;
интеграция и использование плагинов.

Домашние задания

1. Разработать слайдер, который подключается к странице и позволяет получить аналог boostrap carusel

   Цель: В результате выполнения ДЗ вы создадите свою подключаемую библиотеку, которую в дальнейшем сможете использовать на своих страницах

   В задании тренируются навыки
   - работы с DOM элементами из javascript
   - стилизация DOM-элементов

   Необходимо
   - в репозитории прошлого проекта создать новую ветку
   - на главную страницу добавить разметку для слайдера ul > li > img, добавить на корневой ul элемент класс для виджета карусели
   - добавить файл js и написать в нем код, который вместо отображения списка картинок позволит работать с каруселью (работать должно как https://getbootstrap.com/docs/4.0/components/carousel/#with-controls)
   - проверить работу карусели на мобильном

браузере
- сделать пуллреквест
- ссылку на пуллреквест и на задеплоенную страницу сбросить в чат по дз

---

2 **Различие между стандартами языка, инструменты транспиляции, проблемы типизации**

Цели занятия:

объяснить разницу ES5/ES9/TS;
объяснить зачем нужен babel / webpack.

Краткое содержание:

typescript;
basic types;
tsc и настройка tsconfig;
types;
babel.

---

3 **Настройка окружения для современной разработки на TS, использование документации, настройка тестового окружения**

Цели занятия:

настроить webpack для разработки на TS;
объяснить, как typescript помогает при работе с документацией.

Краткое содержание:

webpack + ts;
read ts typings;
editor + typings.

Домашние задания

1 Настроить webpack для работы с typescript, добавить конфигурацию к eslint, решить задачи по типам/TS

Цель: В результате выполнения ДЗ вы создадите базовый скелет для дальнейшей разработки. Он будет включать в себя настройку babel, webpack, typescript

В задании тренируются навыки
- настройки окружения
- создания проекта с 0

Необходимо
- создать новый репозиторий
- инициализировать его с файлом .gitignore
- создать новую ветку (чтобы можно было создать PR)
- настроить линтинг и actions, настроить автодеплой из PR
- сконфигурировать webpack
- добавить поддержку ts файлов
- добавить поддержку импорта css файлов
- реализовать приложение "Игра Жизнь" на языке Typescript https://ru.wikipedia.org/wiki/Игра_«Жизнь»
- ссылку на задеплоенную страницу и на пуллреквест сбросить в чат по дз
- настроить jest и написать тесты на приложение

4 **ООП в JavaScript**

Цели занятия:

разобраться для чего используется ООП;
рассказать про основные принципы, используемые при работе с объектно-ориентированным подходом.

Краткое содержание:

абстракция;
классы и наследование классов.

---

5 **Основы функциональной разработки**

Цели занятия:

рассмотреть очередь задач в JS;
использовать функции как First Level Citizens, отличать чистые функции и понимать их преимущества.

Краткое содержание:

очередь задач и стек вызовов;
чистые функции;
functions as first citizens.

Домашние задания

1 Реализация типовых задач с использованием ООП / ФП с покрытием кода тестами

Цель: В этом задании мы попрактикуемся в работе с typescript и jest

В ходе работы тренируются
- работа с типами
- решение типовых задач разработки
- написание тестов

- создать новый репозиторий
- настроить в нем проект (typescript, jest, линтеры, github actions)
- в новой ветке решить предложенные задачи
- сделать пуллреквест
- сбросить пуллреквест в чат с ментором

Задачи:

1. Написать функцию для каррирования (https://ru.wikipedia.org/wiki/Каррирование)

Пример использования функции

```ts
const func = (a, b, c, d, e) => a + b + c + d + e;

const hof = yourFunction(func);

console.log(hof(1, 2, 3, 4, 5)); // 15
console.log(hof(2, 3, 4)(5, 6)); // 20
console.log(hof(3, 4)(5, 6)(7)); // 25
console.log(hof(4, 5)(6)(7, 8)); // 30
console.log(hof(5)(6)(7)(8)(9)); // 35
```

2. Написать функцию сумматор. При вызове функции с аргументами она суммирует переданные значения (полезно прочитать про методы .valueOf и .toString)

```ts
const sum = function() { /* put your code here */};

alert(sum()); // 0;

const s = sum();
alert(s(1)); // 1
alert(s(1)(2)); //3
alert(s(3)(4)(5)); // 12

const s3 = sum(3);

alert(s3(5)); // 8
alert(s3(6)); // 9
```

3. Реализовать функцию параллельной потоковой обработки данных. В конструктор передается число парралельных "потоков", которое указывает сколько данных обрабатывается в конкретный момент времени

```ts
const runner = new Parallel(2);

console.log(await runner
.jobs(
```

() => new Promise((resolve) => setTimeout(resolve, 10, 1)),
() => new Promise((resolve) => setTimeout(resolve, 50, 2)),
() => new Promise((resolve) => setTimeout(resolve, 20, 3)),
() => new Promise((resolve) => setTimeout(resolve, 90, 4)),
() => new Promise((resolve) => setTimeout(resolve, 30, 5)),
)) // [1, 3, 2, 4, 5];
```

4. Реализовать фукнцию, возвращающую развернутую по спирали матрицу (любой размерности)

```ts
spiral([
[0, 1, 2, 3, 4],
[5, 6, 7, 8, 9],
[10, 11, 12, 13, 14],
[15, 16, 17, 18, 19]
]); // [0, 1, 2, 3, 4, 9, 14, 19, 18, 17, 16, 15, 10, 5, 6, 7, 8, 13, 12, 11]
```

5. Реализовать функцию, реализующую сортировку с учетом правил semver

```ts

semverSort([ "1.0.5", "2.5.0", "0.12.0", "1", "1.23.45", "1.4.50", "1.2.3.4.5.6.7"]); // [ "0.12.0", "1", "1.0.5", "1.2.3.4.5.6.7", "1.4.50", "1.23.45", "2.5.0" ]
```

3 балла за каждую, задание считается принятым от 12 баллов

Если в пуллреквесте не включены/не отработали линтинг и тесты из пуллреквеста, нет ссылки от github actions на codesandbox - задание разворачивается на доработку
```

6	**Разбор базовых приемов разработки на Typescript - типы, интерфейсы, перечисления и прочее**	**Цели занятия:** объяснить какие инструменты Typescript покрывают 80% задач, и как это спасает при разработке проектов. **Краткое содержание:** types; interfaces; function types; nominal types typescript symbol.

7 **Расширенные возможности типизации**

Цели занятия:

разобраться с
- дженериками;
- утилитарными типами;
- созданием typeguards.

Краткое содержание:

generics;
utility types;
conditional types;
typeguards.

Домашние задания

1 Решить задачи на типизацию + типизировать код с прошлого шага

Цель: В результате выполнения ДЗ вы
- потренеруете работу с типами TS
- посмотрите, где тебе могут пригодиться utility types

Необходимо
- создать новый репозиторий
- настроить в нем линтеры, typescript, проверки и pipelines
- добавить в команды package.json команду на проверку валидности TS
- решить задачи на типы (заменить FIXME на реальные типы с использованием utility types/generics, чтобы код компилировался)
- сбросить ссылку на пуллреквест в чат по дз

Задания можно найти здесь
https://gist.github.com/vvscode/8b60049bc335bbc52a4c363f92820956

8 **Консультация**

Цели занятия:

научиться использовать паттерны для структурирования кода.

Краткое содержание:

single responsibility;
service vs component;
abstraction;
incapsulation.

9 **Разработка собственного API**

Цели занятия:

объяснить правила использования асинхронного API;
научиться документировать свой API, следовать соглашениям по разработке.

Краткое содержание:

async API (когда и как?);
callback vs promise interface;
соглашение error-first.

Домашние задания

1 Разработка API для работы с календарем задач - CRUD, фильтрация и поиск

Цель: В этом задании вы потренируете в разработке собственного API для своих библиотек

Необходимо
- в репозитории прошлого задания создать новую ветку
- описать интерфейс для библиотеки для работы с календарем (закладывая работу с произвольным хранилищем)
Должны поддерживаться следующие операции
- CRUD
- фильтрация задач (по тексту, дате, статусу, тегам)

-Реализовать преложенный интерфейс с использованием в качестве хранилища locasStorage

-Покрыть реализацию тестами
- ссылку на пуллреквест сбросить в чат по дз

10 **Разработка шаблонизатора**

Цели занятия:

рассмотреть задачи, стоящие перед шаблонизаторами;
объяснить, как организовать основной функционал.

Краткое содержание:

template engine;
regular expressions;
handlebars / lodash templates.

11 **Консультация**

Цели занятия:

обсудить обновление представлений, подходы к реализации, реактивность;
объяснить, какие есть подходы к организации обновления шаблонов;
научиться реализовать некоторые из них.

Краткое содержание:

императивноие и реактивное программирование;
способы передачи параметров через атрибуты;
принципы устройства virtual dom;
api классовых компонентов.

Домашние задания

1 Реализовать и применить компонент для существующего проекта

Цель: Тренируем работы с презентационным слоем. Добавим к приложению, которое вы уже разработали немножко изящества

Необходимо
- в репозитории прошлого задания "Прогноз погоды" создать новую ветку
- реализовать в ней реализовать задание https://gist.github.com/vvscode/d0bde55073748ee472c8c97b793db5eb
- покрыть реализацию тестами
- сделать пуллреквест
- в общий чат сбросить ссылку на пуллреквест и на страницу с задеплоенным приложением + страницу к которой подключен виджет

- сбросить в чат по дз ссылку на пуллреквест и на страницу с задеплоенным приложением

4 Структурирование программ на JavaScript

1 Mediator и EventEmitter как инструмент организации кода

Цели занятия:

рассмотреть важность низкой связанности кода;
объяснить, как ее можно обеспечить;
объяснить, как EventEmitter может помочь в организации модульного приложения.

Краткое содержание:

связность и связанность кода;
EventEmitter и EventBus.

2 **Управление состоянием приложения, разработка redux**

Цели занятия:

объяснить назначение систем state management;
объяснить api redux, реализовать функционал redux.

Краткое содержание:

state management;
redux pattern;
redux api.

Домашние задания

1 Разработать и покрыть тестами redux-api, реализовать combineReducer

Цель: В результате выполнения ДЗ
- лучше изучите API redux
- разберетесь для чего нужны те или иные его функции
- потренируетесь писать обобщенный код (предназначенный для решения типовых задач)

Необходимо
- создать новый проект, инициализировать его с .gitignore, настройкой линтеров, actions, webpack
- создать новую ветку
- реализовать store api, create Store (без поддержки middlewares), combineReducers
- покрыть код тестами
- сделать пуллреквест

- сбросить ссылку на пуллреквест в чат по дз

3 **Работа с асинхронными actions в redux**

Цели занятия:

объяснить назначение middlewares;
реализовать redux-thunk.

Краткое содержание:

middleware;
redux-thunk;
side effect.

Домашние задания

1 Разработать приложение "чат" на основе redux

Цель: В процессе выполнения задания вы попрактикуетесь в работе с сайд-эффектами в redux-системах, потренируете работу с redux-thunk

Необходимо:
- создать и настроить проект*

- реализовать приложение чат - которое позволяет отправлять сообщения и отображает входящие сообщения из канала в firebase (api и формат сообщения будет предоставлено)
-- создать Redux структуру для приложения час
-- создать UI для приложения
-- добавить функционал по отображению входящих сообщений
-- добавить функционал по отправке сообщений
-- добавить функционал по обработке смайликов (в виде картинок)

- подготовить работу к сдаче*
- сделать ревью 2 других работ
- сбросить ссылку на PR, опубликованный проект и рассмотренные пуллреквесты в чат по дз

4 **REST, RPC и сетевые запросы**

Цели занятия:

разобраться с вариантами реализации сетевых подключений (XHR, fetch, WS);
разобраться с разницей REST, RPC;
посмотреть на возможные варианты работы с запросами с среде redux (thunks, action driven requests).

Краткое содержание:

xhr, fetch, ws;
rest, rpc;
thunks;
запросы управляемы экшенами.

5 Разработка одностраничных приложений

1 **Клиентский роутинг, как строится одностраничное приложение**

Цели занятия:

объяснить, какие API можно использовать для организации SPA, знать алгоритм создания роутера.

Краткое содержание:

history api;
hash api;
роутинг.

Домашние задания

1 Разработка библиотеки клиентского роутинга

Цель: В этом задании вы освоите работу с Hash/History API и узнаете как устроены важные части любого клиентского фреймворка - роутеры
При выполнении задания вы потренируете
- умение струкрутировать код
- работу с браузерными API
- тестирование кода, связанного с сайд-эффектами в бразере

Необходимо:
- создать и настроить проект*

- Разработать библиотеку клиенсткого роутинга
-- конфигурация роутов должна поддерживаться через функции/строки/регулярки
-- должна поддерживаться передача параметров в хуки роутера
-- реализовать поддержку асинхронных onBeforeEnter, onEnter, onLeave
-- добавить настройку для работы с hash/history api
-- опубликовать пакет

- подготовить работу с сдаче*
- сделать ревью 2 других работ
- сбросить ссылку на PR, опубликованный проект и рассмотренные пуллреквесты в част с преподавателем

2 **«Особенности деплоя и сборки одностраничных приложений»**

Цели занятия:

разобрать какие настройки нужно сделать, чтобы успешно деплоить приложения с history api на публичные сервисы

Краткое содержание:

fallback;
настройка webpack.

---

3 **Использование redux для хранения состояния приложения, использование селекторов для отвязки роутов от redux**

Цели занятия:

применить redux для организации логики приложения.

разобрать вопросы:
- ак использование redux может привести к высокой связанности приложения,
- какие паттерны используются для того, чтобы этого избежать (использование функций высшего порядка для связи с редакс)
- использование селекторов для обработки данных, мемоизация в селекторах.

Краткое содержание:

чем удобно использование redux в сравнении с хуками и свойствами роутов;
подписка на обновления redux;
подключение к работе шаблонизатора.

---

4 **Консультация**

Цели занятия:

получить ответы на вопросы.

Краткое содержание:

решение задач.

Домашние задания

1 Разработка сайта-календаря и использованием

клиенского роутинга

Цель: Теперь у вас есть возможность опробовать вашу библиотеку роутинга на практике.

Реализовать одностраничное приложение - календарь задач.

Приложение должно предоставлять минимум следующие страницы:

Календарь
Список
О проекте
Calendar

Все представления позволяют

создавать, редактировать, удалять задачи (заголовок, описание, статус выполнения, дата)
фильтровать по статусу выполнения
делать fuzzy search (можно взять Fuzzy search)
состояние должно отображаться на url (чтобы его можно было сохранять в закладки)

Для выполнения нужно взять:

апи для хранения данных из прошлого домашнего задания
роутер из прошлого домашнего задания
для работы с состоянием использовать redux-toolkit

Задание проверяется после:

открыт пуллреквест
написаны тесты (покрытие 60%)
сделан деплой на github pages

6 Разработка с использованием React.js

1 Что такое React, JSX, настройка окружения

Цели занятия:

объяснить, какую задачу решает React, базовый API React;
объяснить, что такое JSX;
настроить окружение для разработки на React.

Краткое содержание:

react;
jsx;
babel + react;
storybook;
cra.

2 **Умные и глупые компоненты в разрезе React**

Цели занятия:

различать умные и глупые компоненты;
объяснить, как организовать иерархию компонентов.

Краткое содержание:

smart component;
dumb component;
state.

Домашние задания

1 Разработка библиотеки компонентов

Цель: В этом задании мы начинаем работать с библиотекой React. Вы настроите проект, узнаете как работать с Storybook и потренируетесь создавать компоненты

Необходимо:
- создать и настроить проект*

- настроить сторибук
- настроить тесты с локи

- создать компоненты:
-- заголовок (с поддержкой уровней)
-- параграф текста
-- пробельный блок (с горизонтальной линией)
-- схлопывающийся блок (может сворачиваться в заголовк или показывать контент)
-- картинка

-- опубликовать storybook на github pages

- подготовить работу с сдаче*
- сделать ревью 2 других работ
- сбросить ссылку на PR, опубликованный проект и рассмотренные пуллреквесты в част с преподавателем

3 **Жизненный цикл компонентов**

Цели занятия:

рассмотреть жизненный цикл компонента;
использовать его для загрузки данных;
использовать функциональные компоненты с хуками.

Краткое содержание:

lifecycle hooks;
react hooks;
где грузить данные.

---

4 **Совместное использование React и redux**

Цели занятия:

настроить связку React-redux.

Краткое содержание:

react-redux;
connect.

---

5 **Роутинг и ленивая загрузка страниц при работе с React**

Цели занятия:

настроить роутинг в React, использовать React.lazy для ленивой загрузки компонентов.

Краткое содержание:

react-router;
react-lazy.

Домашние задания

1 Приложение для учета расходов

Цель: В результате работы вы создадите прототип приложения для учета расходов

Здесь вы на практике используйте свои наработки из прошлых заданий, плюс поработаете с подключением и использвоанием новых библиотек

В процессе работы вы потренируете

- использование сторонних (и своих) библиотек
- работу с клиентским роутингом в реакте

Нужно сделать приложение для учета расходов

На странице настроек можно заводить категории, и у каждой могут быть подкатегории (вложенность категорий можно ограничить 2, но, если очень хочется, то можно и не ограничивать)
В каждую категорию или подкатегорию на определенную дату можно вносить сумму которую пользователь потратил, если в подкатегории есть расходы, то они должны учитываться и в категории.

Например: Категория "Машина" имеет 2 подкатегории
"Сервис"
"Заправка"

Если на одно и тоже число на "сервис" потрачено 20 единиц, а на "заправку" 10 единиц, то суммарные затраты на это число для категории "машина" должны быть 20 + 10 =30 единиц. Если на это число пользователь не выбрал подкатегорию и занес затраты 5 единиц просто в категорию “машина”, то эти затраты должны тоже учитываться и тогда суммарные затраты - 20 + 10 + 5 = 35 единиц

Нужно иметь возможность выводить расходы за день за неделю за месяц и за произвольный промежуток времени вывод таблицей и диаграммой (pie chart)
Для IOS есть похожее приложение
https://www.dropbox.com/sh/ayn0gfelhele0nc/AABnZcOcGcIowrGzqZehUSESa/Screenshots/Russian?dl=0#/

не стоит возиться с дизайном, можно просто взять 1 из тем с https://bootswatch.com/

Хранение данных должно быть сделано в базе Firebase (https://react-firebase-js.com/). База должна быть независимой для каждого пользователя (поддержка авторизации через react-firebase)

Ожидаемые страницы:
- главная (с заглушкой для не зарегистрированных

пользователей)
- о проекте
- страница логина / регистрации

Остальные страницы должны быть доступны после авторизации:
- настройки
- главная (где можно добавлять доходы и расходы, просматривать данные за последнюю неделю)
- просмотр данных за неделю/месяц/произвольный период. Просмотр в виде таблицы или диаграммы (https://react-google-charts.com/) - период и тип просмотра должен отображаться на урл, чтобы состояние можно было сохранить

При попытке входа на страницу без регистрации должен быть редирект на страницу логина

Задание выполняется в новом репозитории

7 Основы разработки на Node.js

1 Сферы применения Node.js, отличия от разработки в браузере

Цели занятия:

объяснить, что такое Nodejs;
рассмотреть аналоги знакомых инструментов для работы с nodejs (nodemon, debugger и тп).

Краткое содержание:

nodejs framework;
common js;
debugger;
nodemon.

Домашние задания

1 Написать консольное приложение для публикации страниц на github

Цель: В результате работы вы
- познакомитесь поближе с консольными командами и тем, как они могут делать полезную работу
- создадите консольное приложение, которое принимает параметры через аргументы командной строки или через ввод от пользователя
- создадите инструмент, который сможете сами использовать и дорабатывать в дальнейшем

Необходимо:
- создать и настроить проект*

- реализовать скрипт командной строки, который поддерживает ввод параметров
-- через аргументы
-- через интерфейс с запросами в консоли

- позволяет опубликовать проект (собрать его при необходимости) на github pages с заданными параметрами
- команда для сборки (если нужно)
- директория для публикации
- репозиторий для публикации
- настройки доступа к репозиторию

- опубликовать, чтобы можно было использовать через npx
- оформить README с подробным описанием и gif с демонстрацией работы

- подготовить работу с сдаче*
- сделать ревью 2 других работ
- сбросить ссылку на PR, опубликованный проект и рассмотренные пуллреквесты в чат по ДЗ

---

2 **Создание сервера приложений с использованием Node.js**

Цели занятия:

установить и настроить EXPRESS, подключать к нему мидлвары для сессий и шаблонизации.

Краткое содержание:

parse args;
commander;;
inquirer
основные модули для работы с FS/OS.

8 Сопровождение кода

1 **Презентация работы - что сделать, чтобы было хорошо**

Цели занятия:

создать привлекательный README, оформлять репозиторий и пакеты при публикации

Краткое содержание:

readme badges;
readme rules;
настройка readme для гитхаб профиля (расширенная информация на странице).

Домашние задания

1 Создать приложение-игру крестики-нолики с поддержкой многопользовательской игры

Цель: В результате выполнения ДЗ вы создадите базовый WS-сервер, с поддержкой подключения клиентов, узнаете как общаться чреез WS и как настраивать выполнение серверного javascript. Потренируете навыки разработки в контексте многопользовательской работы и используете redux не только на клиенте

Необходимо:
- создать и настроить проект*

- создать приложение сервер, которое поддерживает подключение 2 пользователей на игру, и любого числа пользователей на просмотр
- создать клиентское приложение, которое может работать с сервером, отображать ход игры и позволяет активным игрокам делать ходы
- обработать ситуации, когда зрители пытаются сделать ход, или актинвый игрок ходит вне своей очереди
- опубликовать работу с использованием сервиса heroku

- подготовить работу с сдаче*
- сделать ревью 2 других работ
- сбросить ссылку на PR, опубликованный проект и

рассмотренные пуллреквесты в част с преподавателем

9 Проектная работа

1	**Защита проектных работ**	**Цели занятия:** защитить проект и получить рекомендации экспертов. **Краткое содержание:** презентация проектов перед комиссией; вопросы и комментарии по проектам.