

Индивидуальный предприниматель
Янковский Владимир Александрович

Утверждена приказом
№ 1 от «30» января 2024 года
Индивидуальный предприниматель
Янковский Владимир Александрович

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
«Программирование на JavaScript»

Возраст обучающихся: 13-20 лет.
Срок реализации программы: 79 часов
Форма обучения: очно-заочная, с применением
исключительно электронного обучения,
дистанционных образовательных технологий
Автор-составитель: Янковский Владимир Александрович

г. Омск, 2024 год

Содержание

№	Наименование раздела	Страница
1.	Комплекс основных характеристик программы	
1.1.	Пояснительная записка	3
1.2.	Цели и задачи дополнительной общеобразовательной общеразвивающей программы	3
1.3.	Планируемые результаты освоения программы	4
1.4.	Содержание дополнительной общеобразовательной общеразвивающей программы	5
2.	Комплекс организационно-педагогических условий	
2.1.	Учебный план	6
2.2.	Календарный учебный график	8
2.3.	Формы, порядок и периодичность проведения промежуточной аттестации обучающихся	8
2.4.	Оценочные и методические материалы	8
2.5.	Материально-техническое обеспечение образовательного процесса	26
2.6.	Кадровые условия обеспечения образовательного процесса	27
2.7.	Список литературы	28

Раздел 1. Комплекс основных характеристик программы

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование на JavaScript» направлена на формирование у обучающихся основных навыков программирования и проектирования, формирование профессиональной направленности.

Образовательная программа разработана на основании следующих нормативно-правовых документов:

Федеральный закон от 29.12.2012 года № 273-ФЗ «Об образовании в Российской Федерации»;

Приказ Министерства просвещения Российской Федерации от 27.07.2022 года № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

Актуальность программы: В современном мире умение программировать и работать с технологиями для веб-сайтов является ценным и востребованным навыком, который все больше и больше ценится на рынке, особенно сегодня, в условиях ограниченных ресурсов и повышенных расходов. В результате прохождения обучения слушатель сформирует знания в области основ программирования, разовьет логическое и алгоритмическое мышление, активизирует навыки использования полученных знаний и умений в практической деятельности.

Адресат общеразвивающей программы: данная программа разработана для лиц от 13 до 20 лет. К освоению программы допускаются лица, имеющие базовые знания html и css.

Режим занятий: Продолжительность одного академического часа – 30 минут. Общее количество часов в неделю указано в календарном учебном графике. Занятия проводятся 5 раз в неделю.

Расписание занятий составляется для создания наиболее благоприятного режима труда и отдыха обучающихся и возрастных особенностей обучающихся.

Срок освоения общеразвивающей программы: программа рассчитана на 6 недель обучения.

Особенности организации образовательного процесса: Форма обучения – очно-заочная, с применением исключительно электронного обучения, дистанционных образовательных технологий.

Формы занятий: Лекции, практическая работа, самостоятельная работа.

1.2. Цели и задачи дополнительной общеобразовательной общеразвивающей программы

Цель программы: обучение обучающихся программированию посредством языка JavaScript, развитие инженерного мышления, формирование профессиональной направленности.

Задачи программы:

1. Сформировать у обучающихся базовые представления о языках программирования, алгоритме (программе), исполнителе, способах записи алгоритма;
2. овладеть понятиями «объект», «событие», «управление», «обработка событий»;

3. изучить базовые алгоритмические структуры;
4. содействовать развитию общей информационной культуры как одного из аспектов будущей профессиональной деятельности;
5. подготовить необходимую теоретическую и практическую базу знаний и умений для дальнейшей работы слушателя с серверными языками программирования и базами данных

1.3. Планируемые результаты освоения программы

Личностные результаты:

- понимание человеческих, культурных и социальных аспектов, связанных с информационными технологиями;
- владение первичными навыками анализа и критичной оценки получаемой информации;
- ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- развитие чувства личной ответственности за качество окружающей информационной Среды.

Метапредметные результаты:

- владение умением организации собственной учебной деятельности, включающим постановку цели (учебной задачи), планирование последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработку структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;
- владение основными универсальными умениями информационного характера: постановка и формулирование проблемы; поиск и выделение необходимой информации; структурирование и визуализация информации; самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера;
- владение умениями самостоятельно планировать пути достижения целей, соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, оценивать правильность выполнения учебной задачи.

Предметные результаты:

В результате изучения программы слушатель должен знать и уметь:

- базовые понятия и терминологию языка JavaScript;
- алгоритмические конструкции;
- общую структуру языка JavaScript, его синтаксис;
- правила записи операторов языка JavaScript, порядок выполнения операций, стандартные и пользовательские функции, записи и выполнения арифметических и логических функций;
- основы объектной модели документа (DOM).
- изменять содержимое HTML-страниц средствами JavaScript, в том числе добавлять, удалять, отображать и скрывать элементы объектной модели документа, вносить изменения в их текстовое наполнение;
- модифицировать оформление HTML-страниц средствами JavaScript, в том числе форматировать текст, списки, таблицы, управлять положением блоков на странице, добавлять элементам классы CSS, вносить изменения в свойства CSS.

1.4. Содержание дополнительной общеобразовательной общеразвивающей программы

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование на JavaScript» включает в себя двадцать четыре темы:

1. «Введение в язык, подключение javascript. Основы работы с Git». Знакомство с языком javascript, особенности, применение языка. Работа с javascript в браузере. Понятие переменной, объявление переменных, типы данных. GIT – система контроля версий. GIT HUB- сервис для хостинга IT-проектов.
2. «Типы данных, операторы, методы и свойства». Работа с типами данных и переменными. Операторы в javascript. Логические операторы. Операторы сравнения. Методы и свойства объекта.
3. «Динамическая типизация данных в javascript. Условия, ветвления». Динамическая типизация. Преобразование типов данных. Ветвления.
4. «Функции, анонимные функции, callback - функции, чистые функции». Понятие функции. Написание функций в javascript. Callback-функция в javascript. Правила написания кода.
5. «Циклы. Рекурсия». Реализация повторяющихся задач с использованием циклов. Рекурсия и циклические операторы.
6. «Функции. Область видимости и замыкание». Типы областей видимости. Глобальная область видимости. Локальная область видимости. Замыкания.
7. «Знакомимся с объектами и массивами, методы переборов и псевдомассивы». Объекты, массивы, псевдомассивы. Методы перебора массивов.
8. «Подготовка для работы со страницей». Рефакторинг. Подготовка проекта к подключению верстки.
9. «DOM». Знакомство с моделью Document Object Model (DOM). Метод getElementBy. Работа с атрибутами. Работа со стилями.
10. «Работа с DOM». Создание, удаление, перемещение объектов на странице. Устаревшие методы.
11. «Обработчики событий. Особенности их работы». События в javascript. Обработка с помощью функции. Особенности их работы. Всплытие событий. Предотвращение всплытия. Погружение.
12. «Storage and Cookie». Объекты веб-хранилища. Строки данных.
13. «Call Stack, контекст вызова This, Стрелочные функции». Стек вызовов. Call Stack. Контекста вызова This. Объект инициатора вызова. Явная и неявная привязка. Стрелочные функции.
14. «ООП - Наследование, прототипы, конструкторы и классы». Объектно-ориентированное программирование. Прототип. Доступ к прототипу. Пишем прототип. Функции конструкторы. Наследование.
15. «Особенности современного стандарта ES6». Стандарт ES6. Стрелочные функции. Интерполяция. Classes. Статические переменные. Статические методы. Вес объекта. Наследование. Геттеры и сеттеры. Деструктуризация. Операторы Rest и Spread.
16. «Модульная структура». Функциональные блоки. Инкапсуляция. Модульный подход. Сборка модулей. Webpack dev server. ESLint.

17. «Скрипты и время выполнения». Скрипты и время выполнения `setTimeout`, `setInterval` и `requestAnimationFrame`. Объект `Date`.
18. «Параметры документа, окна и работа с ними». Объект `window.screen`. Событие `resize`. Событие `scroll`. Работа с элементами на странице.
19. «Регулярные выражения». Создание регулярного выражения. Написание шаблона регулярного выражения. Использование простых шаблонов. Использование специальных символов. Работа с регулярными выражениями.
20. «Делегирование». Делегирование событий. Применение делегирования: действия в разметке.
21. «Функции помощники». Функции помощники внутри модульного javascript. Паттерн анимации.
22. «Promise. Fetch». Асинхронный javascript. Создание Promise. Пример с `setTimeout`. Промисификация. Метод `Fetch`.
23. «Async/Await, Try/Catch». Конструкция `Async/Await`. Конструкция `Try/Catch`. Обработка ошибок.
24. «Админ-панель, json-server, серверные запросы». Работа с серверными запросами. Инструмент `json-server`.

С учетом проведения занятий с применением дистанционных образовательных технологий и электронного обучения, в водной части (подготовительный урок) обозначаются правила работы и взаимодействия (технические особенности работы и правила обмена информацией).

Раздел 2. Комплекс организационно-педагогических условий

2.1. Учебный план

№ п/п	Название темы	Количество часов				Формы аттестации
		Всего	Теоретическая работа	Практическая работа	Самостоятельная работа	
1.	Введение в язык, подключение javascript. Основы работы с Git	6	3	0	3	Задание
2.	Типы данных, операторы, методы и свойства	2	1	0	1	Задание
3.	Динамическая типизация данных в javascript. Условия, ветвления	2	1	0	1	Задание
4.	Функции, анонимные функции, callback - функции, чистые функции	3	1	1	1	Задание
5.	Циклы. Рекурсия	3	1	1	1	Задание

6.	Функции. Область видимости и замыкание	2	1	0	1	Задание
7.	Знакомимся с объектами и массивами, методы переборов и псевдомассивы	3	1	1	1	Задание
8.	Подготовка для работы со страницей	2	1	0	1	Задание
9.	DOM	2	1	0	1	Задание
10.	Работа с DOM	2	1	0	1	Задание
11.	Обработчики событий. Особенности их работы	3	1	1	1	Задание
12.	Storage and Cookie	3	1	1	1	Задание
13.	Call Stack, контекст вызова This, Стрелочные функции	2	1	0	1	Задание
14.	ООП - Наследование, прототипы, конструкторы и классы	4	1	2	1	Задание
15.	Особенности современного стандарта ES6	2	1	0	1	Задание
16.	Модульная структура	4	2	0	2	Задание
17.	Скрипты и время выполнения	3	1	1	1	Задание
18.	Параметры документа, окна и работа с ними	3	1	1	1	Задание
19.	Регулярные выражения	4	2	1	1	Задание
20.	Делегирование	6	1	4	1	Задание
21.	Функции помощники	2	1	0	1	Задание
22.	Promise. Fetch	3	2	0	1	Задание
23.	Async/Await, Try/Catch	3	1	1	1	Задание
24.	Админ-панель, json-server, серверные запросы	6	3	0	3	Задание
25.	Итоговый контроль	4	0	0	4	Проект

2.2. Календарный учебный график

Период обучения (нед.)/кол-во ак. часов	1	2	3	4	5	6
Наименование программы						
Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование на JavaScript»	16	11	14	20	10	8

Общее количество часов освоения дополнительной общеобразовательной общеразвивающей программы социально-гуманитарной направленности – 79 часов. Календарные сроки реализации программы устанавливаются в соответствии с потребностями обучающегося на основании договора об оказании платных образовательных услуг.

2.3. Формы, порядок и периодичность проведения промежуточной аттестации обучающихся

Формы промежуточной аттестации определены учебным планом дополнительной общеобразовательной общеразвивающей программы технической направленности «Программирование на JavaScript». Промежуточная аттестация проводится после освоения обучающимся каждой темы. По окончании освоения программы проводится итоговый контроль – выполняется необходимый функционал согласно техническому заданию. Обучающийся, выполнивший все задания, считается успешно прошедшим промежуточную аттестацию. Оценочные материалы для проведения промежуточной аттестации приведены в разделе 2.4 данной программы.

2.4. Оценочные и методические материалы

Задание к уроку «Введение в язык, подключение javascript. Основы работы с Git»

- 1) Создать пустой репозиторий на GitHub без файла README.md
- 2) Создать HTML страницу и подключить к ней файл скрипта
- 3) В репозитории создать отдельную ветку dev на GitHub и поместить туда index.html и script.js
- 4) Выполнить push
- 5) В файле скрипта создать переменные:
 - title,
 - screens,
 - screenPrice,
 - rollback,
 - fullPrice,
 - adaptive
- 6) Вывести на экран в модальном окне (alert) сообщение с любым текстом
- 7) Вывести в консоль сообщение с любым текстом
- 8) Проверить, чтобы все работало и не было ошибок в консоли
- 9) В репозитории создать новую ветку lesson01, выполнить commit в новую ветку и push

Задание к уроку «Типы данных, операторы, методы и свойства»

- 1) Следующим переменным присвоить значения
 - title- строка с названием проекта,
 - screens - строка с названиями типов экранов через запятую («Простые, Сложные, Интерактивные»),

- screenPrice- любое число,
- rollback - любое число от 1 до 100,
- fullPrice- любое число (сколько хотите заработать),
- adaptive- булево значение

2) Используя методы и свойства:

- Вывести в консоль тип данных значений переменных title, fullPrice, adaptive;
- Вывести в консоль длину строки из переменной screens
- Вывести в консоль «Стоимость верстки экранов (screenPrice) рублей/долларов/гривен/юани» и «Стоимость разработки сайта (fullPrice) рублей/долларов/гривен/юани»
- Привести строку screens к нижнему регистру и разбить строку на массив, вывести массив в консоль
- Вывести в консоль Процент отката посреднику за работу (fullPrice * (rollback/100))

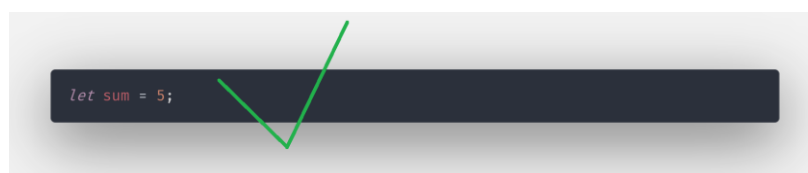
3) Проверить, чтобы все работало и не было ошибок в консоли 4) Добавить папку или ветку со вторым уроком в свой репозиторий на GitHub

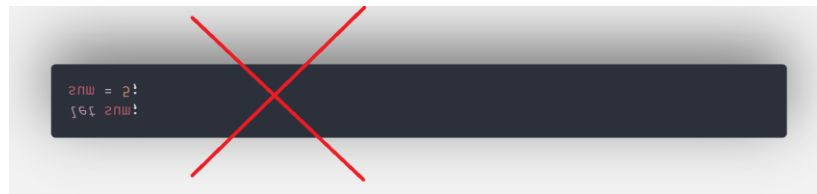
Необходимо выполнить в отдельном JS файле, подключенному к отдельной HTML странице

- 1) Создать переменную num со значением 266219 (тип данных число)
- 2) Вывести в консоль произведение (умножение) цифр этого числа
Например: число 123, при помощи javascript получить каждую цифру (1, 2, 3) и перемножить их. Правильно использовать цикл или методы перебора.
- 3) Полученный результат возвести в степень 3, используя только 1 оператор (Math.pow не подходит)
- 4) Вывести в консоль первые 2 цифры полученного числа
- 5) В отдельном репозитории для усложненных уроков, добавить папку или ветку со вторым уроком в свой репозиторий на GitHub

Задание к уроку «Динамическая типизация данных в Javascript. Условия, ветвления»

- 1) Весь функционал что был ранее оставляем, если что-то необходимо удалить, об этом будет написано в задании
- 2) При присваивании уже объявленным пустым переменным какого-либо значения, значение переменной присваиваем сразу при декларации (не нужно создавать пустую переменную, а потом ей же присваивать значение)





- 3) Спрашиваем у пользователя «Как называется ваш проект?» и результат сохраняем в переменную title
- 4) Спросить у пользователя «Какие типы экранов нужно разработать?» сохранить в переменную screens (пример: «Простые, Сложные, Интерактивные»)
- 5) Спросить у пользователя «Сколько будет стоить данная работа?» и сохранить в переменную screenPrice (пример: 12000)
- 6) Спросить у пользователя «Нужен ли адаптив на сайте?» и сохранить данные в переменной adaptive (булево значение true/false)
- 7) Спросить у пользователя по 2 раза каждый вопрос и записать ответы в разные переменные 1. «Какой дополнительный тип услуги нужен?» (например service1, service2) 2. «Сколько это будет стоить?» (например servicePrice1, servicePrice2) в итоге 4 вопроса и 4 разные переменных, вопросы задаются в такой последовательности Название - Стоимость - Название - Стоимость
- 8) Вычислить итоговую стоимость работы учитывая стоимость верстки экранов и дополнительных услуг (screenPrice + servicePrice1 + servicePrice2) и результат занести в переменную fullPrice
- 9) Объявить переменную servicePercentPrice и занести в нее итоговую стоимость за вычетом отката посреднику (servicePercentPrice = fullPrice - Откат посреднику), округлив результат в большую сторону (методы объекта Math в помощь). Вывести servicePercentPrice в консоль.
- 10) Написать конструкцию условий (расчеты приведены в рублях) (**вывести в консоль**)
Если fullPrice больше 30000, то «Даем скидку в 10%»
Если fullPrice больше 15000 и меньше 30000, то сообщение «Даем скидку в 5%»
Если fullPrice меньше 15000 и больше 0 то в консоль вывести сообщение «Скидка не предусмотрена»
Если отрицательное значение, то вывести «Что то пошло не так»
Учесть варианты 0, 15000 и 30000(к какому уровню не важно)
Проверить, чтобы все работало и не было ошибок в консоли. Добавить папку с третьим уроком в свой репозиторий на GitHub.

Необходимо выполнить в отдельном JS файле, подключенному к отдельной HTML странице

- 1) Переменная lang может принимать 2 значения: 'ru' 'en'.
Написать условия при котором в зависимости от значения lang будут выводиться дни недели на русском или английском языке. Решите задачу
 - a) через if,
 - b) через switch-case
 - c) через многомерный массив без ифов и switch.

2) У нас есть переменная `namePerson`. Если значение этой переменной «Артём» то вывести в консоль «директор», если значение «Александр» то вывести в консоль «преподаватель», с любым другим значением вывести в консоль «студент».

Решить задачу с помощью нескольких тернарных операторов, без использования `if` или `switch`

3) Запустить проект в репозиторий для усложненных заданий на GitHub.

Задание к уроку «Функции, анонимные функции, `callback` - функции, чистые функции»

1) Объявить функцию `getAllServicePrices`. Функция возвращает сумму всех дополнительных услуг. Результат сохраняем в переменную `allServicePrices`. Тип - `function expression`

2) Объявить функцию `getFullPrice`. Функция возвращает сумму стоимости верстки и стоимости дополнительных услуг (`screenPrice + allServicePrices`). Результат сохраняем в переменную `fullPrice`. Тип - `function declaration`

3) Объявить функцию `getTitle`. Функция возвращает `title` меняя его таким образом: первый символ с большой буквы, остальные с маленькой". Учесть вариант что строка может начинаться с пустых символов. « КальКулятор Верстки»

4) Объявить функцию `getServicePercentPrices`. Функция возвращает итоговую стоимость за вычетом процента отката. Результат сохраняем в переменную `servicePercentPrice` (итоговая стоимость минус сумма отката)

5) Почистить консоль логи и добавить недостающие, должны остаться:

- вызовы функции `showTypeOf`

- вывод строки с типами экранов для разработки `screens`

- сообщение о скидке пользователю (вызовы функции `getRollbackMessage`)

- стоимость за вычетом процента отката посреднику (вызовы функции `getServicePercentPrices`)

6) Проверить, чтобы все работало и не было ошибок в консоли

7) Добавить папку с четвертым уроком в свой репозиторий на GitHub

Необходимо выполнить в отдельном JS файле, подключенному к отдельной HTML странице

1) Создайте функцию, которая принимает 1 аргумент (название произвольное)

— Если в качестве аргумента передана не строка - функция оповещает об этом пользователя

— В полученной (как аргумент) строке функция должна убрать все пробелы в начале и в конце

— Если строка более 30 знаков - то после 30го символа часть текста скрывается и вместо них появляются три точки (...)

2) Запустить проект в репозиторий для усложненных заданий на GitHub

Задание к уроку «Циклы. Рекурсия»

1) Переписать получение значения переменной `screenPrice` циклом `do while`. Вопрос должен задаваться один раз обязательно, далее уже по условию

- 2) Добавить проверку что введённые данные являются числом, которые мы получаем на вопрос «Сколько это будет стоить» в функции getAllServicePrices
- 3) Поправить проект так, чтоб расчеты велись верно. Проверить типы получаемых переменных и привести их к нужным.
- 4) Проверить, чтобы все работало и не было ошибок в консоли.
- 5) Добавить папку с уроком в свой репозиторий на GitHub

Придумать способ сохранять в переменную ответ пользователя после проверки на число именно как число при любом вводе. (с пробелами и без в переменную заносилось именно число) На данный момент проверка isNumber пропустит такой вариант « 123 » и именно это значение попадет в переменную. Необходимо это исправить. Так же учитывайте, что человек может нажать отмену и в проверку уйдет значение NULL.

Необходимо выполнить в отдельном js файле, подключенному к отдельной HTML странице

- 1) Создать массив arr = []
 - Записать в него 7 любых многозначных чисел в виде строк
 - Вывести в консоль только те, что начинаются с цифры 2 или 4 (Должны присутствовать в массиве)
- 2) Вывести в столбик все простые числа от 1 до 100 (сделать при помощи цикла)
 - Статья про простые числа - КЛИК
 - Рядом с каждым числом написать оба делителя данного числаНапример: “Делители этого числа: 1 и n”
- 3) Запустить проект в репозиторий для усложненных заданий на GitHub

Задание к уроку «Функции. Область видимости и замыкание»

Для этого задания создайте отдельный репозиторий.

Используйте функции alert, confirm, prompt для общения с пользователем.

Написать игровой бот.

«Загадывание случайного числа от 1 до 100»

Что должна делать программа:

- спрашивает пользователя: «Угадай число от 1 до 100».
- если пользовательское число больше, то бот выводит «Загаданное число меньше» и предлагает ввести новый вариант;
- если пользовательское число меньше, то бот выводит «Загаданное число больше» и предлагает ввести новый вариант;
- если пользователь ввел не число, то выводит сообщение «Введи число!» и предлагает ввести новый вариант;
- если пользователь нажимает «Отмена», то игра заканчивается и выводится сообщение «Игра окончена».
- если пользовательское число равно загаданному, то игра заканчивается и выводит сообщение «Поздравляю, Вы угадали!!!».

Программа должна быть выполнена с помощью рекурсии, без единого цикла.

Загаданное число должно храниться «в замыкании»

Дописать функционал игрового бота.

Кол-во попыток пользователя должно быть ограничено: 10

— если пользовательское число больше, то бот выводит «Загаданное число меньше, осталось попыток ...» и предлагает ввести новый вариант;

— если пользовательское число меньше, то бот выводит «Загаданное число больше, осталось попыток ...» и предлагает ввести новый вариант;

— если пользователь вводит правильное число, то бот выводит «Поздравляю, Вы угадали!!! Хотели бы сыграть еще?», при нажатии ОК игра перезапускается (снова 10 попыток и новое загаданное число);

— если пользователь ввел не число, то выводит сообщение «Введи число!» и предлагает ввести новый вариант;

— если пользователь нажимает «Отмена», то игра выводит прощальное сообщение и завершается;

— если закончились попытки, то программа сообщает: «Попытки закончились, хотите сыграть еще?»

Программа должна быть выполнена с помощью рекурсии, без единого цикла. Загаданное число и оставшиеся кол-во попыток должно храниться «в замыкании».

Задание к уроку «Знакомимся с объектами и массивами, методы переборov и псевдомассивы»

1) Перенести все функции в объект (сделать их методами объекта)

2) Создать в объекте метод **start** и перенести в него вызов метода **asking** и переопределение свойств. Вне самого объекта запускаем только метод **start** который в нужном порядке выполнит все действия.

3) Создать в объекте метод **logger** который будет выводить в консоль необходимую информацию. Данный метод запускаем в самом конце метода **start** (после того как все расчеты уже были произведены).

4) Вывести в консоль из метода **logger** все свойства и методы объекта `appData` с помощью цикла **for in**

Таким образом вне объекта теперь должен быть только вызов метода **start()**

Поправить весь проект, ошибок в консоли быть не должно, а в консоль должна выводиться необходимая информация!

Необходимо выполнить в отдельном js файле, подключенному к отдельной HTML странице

1) Создать массив `week` и записать в него дни недели в виде строк

Вывести на экран все дни недели

Каждый из них с новой строчки

Выходные дни - курсивом

Текущий день - жирным шрифтом(использовать объект даты)

2) Запустить проект в репозиторий для усложненных заданий на Github

Задание к уроку «Подготовка для работы со страницей»

1) Сделать проверку при получении данных:

- ответ на вопрос «Как называется ваш проект?» - строка

- ответ на вопрос «Какие типы экранов нужно разработать?» - строка
- ответ на вопрос «Сколько будет стоить данная работа?» - число
- ответ на вопрос «Какой дополнительный тип услуги нужен?» - строка
- ответ на вопрос «Сколько это будет стоить?» - число

Что значит проверка данных: где должен быть текст там только текст (голые цифры не должно пропускать, а текст с цифрами - должно. Пример: «Купил ВАЗ 2108» - ок, «4567989» - нет), где цифры только цифры!

Если проверку не прошло, то переспрашивать

- 2) Проверить, чтобы все работало и не было ошибок в консоли
- 3) Добавить папку с уроком на свой GitHub

- 1) Если пользователь введет одинаковый ответ на вопрос «Какой дополнительный тип услуги нужен?», то необходимо не перезаписывать, а сохранять оба ответа в `appData.services`, добавить каждому названию ключа уникальность!
- 2) Посчитать свойство `appData.screenPrice` методом `reduce`.

Задание к уроку «DOM»

Необходимо выполнить данное задание в скрипт-файле из прошлых уроков (перенести его в проект из файлов для урока, заменить уже имеющийся пустой `script.js`)

Не забываете, все элементы со страницы мы получаем в самом верху кода. (в самую первую очередь)

- 1) Задание по проекту, получить каждый элемент в отдельную переменную:

Получить заголовок «Калькулятор верстки» через метод `getElementsByTagName`. (тэг `h1`, получить именно элемент, а не коллекцию)

Получить кнопки «Рассчитать» и «Сброс» через метод `getElementsByClassName`. (класс `handler_btn`)

Получить кнопку «+» под выпадающим списком через метод `querySelector`. (класс `screen-btn`)

Получить все элементы с классом `other-items` в две разные переменные. В первую элементы у которых так же присутствует класс `percent`, во вторую элементы у которых так же присутствует класс `number` через метод `querySelectorAll`.

Получить `input type=range` через его родителя с классом `rollback` одним запросом через метод `querySelector`.

Получить `span` с классом `range-value` через его родителя с классом `rollback` одним запросом через метод `querySelector`.

Получить все инпуты с классом `total-input` справа через метод `getElementsByClassName`. (класс `total-input`, получить именно элементы, а не коллекции)

Получить все блоки с классом `screen` в изменяемую переменную (`let`) через метод `querySelectorAll` (далее мы будем переопределять ее значение)

- 2) Проверить, чтобы все работало и не было ошибок в консоли
- 3) Добавить папку с уроком на свой GitHub

Необходимо выполнить в отдельном `js` файле, подключенному к отдельной `html` странице

- 1) Выведите на страницу текущую дату и время в 2-х форматах:

- а) 'Сегодня Вторник, 4 февраля 2020 года, 21 час 5 минут 33 секунды'
- б) '04.02.2020 - 21:05:33'
- 2) Для вывода в формате (а) напишите функцию, которая будет менять склонение слов в зависимости от числа, «час, часов, часа»
- 3) Для вывода в формате (б) напишите функцию, которая будет добавлять 0 перед значениями которые состоят из одной цифры (из 9:5:3 1.6.2019 сделает 09:05:03 01.06.2019)
- 4) С помощью функции `setInterval`, реализуйте обновление даты и времени каждую секунду
- 5) Добавить папку с уроком на свой GitHub

Задание к уроку «Работа с DOM»

Скачать архив, прикрепленный к уроку (`you-dont-know-js`). В `index.html` код менять нельзя, только подключить скрипт. Используя только файл скрипта выполнить такие действия:

Восстановить порядок книг.

Заменить картинку заднего фона на другую из папки `image`

Исправить заголовок в книге 3 (Получится – «Книга 3. `this` и Прототипы Объектов»)

Удалить рекламу со страницы

Восстановить порядок глав во второй и пятой книге (внимательно инспектируйте индексы элементов, поможет `dev tools`)

в шестой книге добавить главу «Глава 8: За пределами ES6» и поставить её в правильное место.

Проверить, чтобы все работало и не было ошибок в консоли. Добавить папку с уроком на свой GitHub.

- 1) Скопировать к себе доску `coderep additional task`
- 2) Добавлять новые элементы с текстом из инпута
- 3) Прикрепить ссылку на `Coderep`

Задание к уроку «Обработчики событий. Особенности их работы»

Для этого задания создайте отдельный репозиторий. Скачать архив, прикрепленный к уроку.

1) Повесить на кнопку обработчик события `click` и реализовать такой функционал:

В `input[type=text]` можно ввести цвет: `red`, `green`, `blue` и так далее.

По нажатию на кнопку необходимо брать этот цвет и добавлять его свойству `style=>background-color:>` квадрата

Работать должно так: ввели в `input[type=text]` `yellow`, по нажатию на кнопку значение `input[type=text]` попадает в свойство `style=>background-color: yellow` и фон квадрата должен поменяться

2) В кружке (который внутри квадрата) есть кнопка. Дать ей свойство `style= «display: none;»`

3) Повесить на `input[type=range]` обработчик события `input` и реализовать такой функционал:

при каждом изменении положения ползунка значение `input[type=range]` необходимо заносить в свойства ширины и высоты кружка (который внутри квадрата) (`height` и `width`) (в процентах!!)

- 1) Запретить нажатие кнопки рассчитать если не выбран ни один тип экрана в выпадающем списке и не введено их количество. Учесть, что блоков с типом экранов может быть несколько, но пустых (незаполненных) элементов быть не должно
- 2) Повесить на `input[type=range]` (в блоке с классом `.rollback`) обработчик события. При перемещении ползунка значение под ним (в элементе `span`) должно меняться. А также это значение должно заноситься в свойство `rollback` нашего объекта для последующих расчетов!
- 3) В нашем объекте присутствует метод `getServicePercentPrice`. Данный метод рассчитывает доход с учетом отката посреднику. Перенести его логику в метод `addPrices` и выводить в поле с подписью «Стоимость с учетом отката»
- 4) В методе `addScreens` мы добавляем в свойство `appData.screens` новые объекты. Добавить свойство `count` в которое занести количество экранов из `input`. В методе `addPrices` посчитать общее количество экранов и вывести на страницу итоговое значение в поле с подписью «Количество экранов»
- 5) Удалить из проекта метод `getRollbackMessage`

1) Сделать так, чтоб после нажатия на кнопку. Рассчитать изменение значения `input[type=range]` меняло и сумму в поле с подписью «Стоимость с учетом отката». Сумма должна пересчитываться с учетом реального значения процента отката. Проверить чтоб значение не менялось до расчета, только после расчёта.

Задание к уроку «Storage and Cookie»

Это задание выполняется отдельно от нашего проекта с бюджетом!

Для этого задания создайте отдельный репозиторий. Для выполнения задания можно использовать готовую верстку или сверстать самостоятельно.

Напишем свой `ToDo List`

Посмотрите дополнительное видео в уроке

Что необходимо реализовать (первые 2 пункта делаем по видео):

1. Отмечать выполненные дела, выполненные дела должны перемещаться в блок с выполненными делами
 2. Поле ввода после добавления дела должно очищаться
 3. Пустые дела добавляться не должны
 4. Удаление дел на кнопку КОРЗИНА
 5. Сохранять данные о делах в `localStorage` (советую в виде массива)
 6. Дела из `localStorage` подгружаться должны автоматически при загрузке странице
- внимание** чтобы сохранить массив в `localStorage` необходимо его конвертировать в `json` формат (`JSON.stringify`)
- внимание** из `localStorage` мы всегда получаем `json` строку и её необходимо конвертировать обратно в формат `javascript` (`JSON.parse`)
- Проверить, чтобы все работало и не было ошибок в консоли (Учесть вариант отсутствия объекта в `localStorage` пользователя при первой загрузке страницы)
- Сохранить проект в отдельном репозитории на `GitHub`

Задание к уроку «Call Stack, контекст вызова This, Стрелочные функции»

- 1) Перевести весь проект на стрелочные функции (кроме методов объекта).
- 2) В нашем объекте везде использовать `this` как ссылку на объект `appData`. Если какой то обработчик ломает контекст - привязать нужный контекст в виде объекта `appData`.
- 3) Блокировать (свойство `disabled`) все `input[type=text]` и `select` с левой стороны после нажатия кнопки. Рассчитать, после этого кнопка Рассчитать пропадает и появляется кнопка Сброс (`id=reset`).
- 4) В объекте реализовать метод `reset()`, срабатывающий по нажатию на кнопку Сброс. Метод `reset()` должен привести объект к исходному состоянию:

Кнопка Сброс должна замениться на кнопку Рассчитать

Должны быть убраны все дополнительные элементы (которые добавлялись динамически) и значения полей ввода

Все `input[type=text]` и `select` должны быть разблокированы

Метод `reset` должен всю программу возвращать в исходное состояние

Метод `reset()` пишем самостоятельно, никаких перезагрузок страницы. Метод должен быть расписан наподобие `start()`.

Проверить, чтобы все работало и не было ошибок в консоли. Добавить папку с уроком на свой GitHub.

- 1) В нашем проекте (в верстке) есть `input[type=checkbox]` с `id=cms-open`. При его выборе должен открываться блок с классом `hidden-cms-variants`.
Внимание, блоку с классом `hidden-cms-variants` необходимо добавлять свойство `display: flex`, а не `display: block`.
- 2) При выборе `option` с значением «Другое» (`value=other`) должен открываться блок с классом `main-controls__input`, но только тот, что внутри блока с классом `hidden-cms-variants` (ВНИМАНИЕ, блоков с классом `main-controls__input` в проекте много, искать стоит внутри определенного элемента).
- 3) Если в `input[type=checkbox]` выбран вариант с числовым `value` (`value=50`) то высчитываем общую стоимость работы с учетом данного `value`. Значение - процент от общей стоимости работы.
Пример: общая стоимость работы равна 30.000. При выборе варианта WordPress с `value=50` стоимость работы рассчитывается так: $30.000 + 15.000 = 45.000$ (15.000 это 50% от 30.000).
- 4) При нажатии на кнопку Сброс метод `reset()` должен возвращать в исходное состояние и блок с классом `hidden-cms-variants`.

Задание к уроку «ООП - Наследование, прототипы, конструкторы и классы»

Данное задание является отдельной программой, поэтому его необходимо выполнять в отдельной папке, в которой нужно создать базовую html разметку и файл JavaScript.

- 1) Создать функцию-конструктор `DomElement`, который содержит

свойства: `selector`, `height`, `width`, `bg`, `fontSize`

содержит метод, который создает элемент на странице в зависимости от условия:

если строка `selector` начинается с точки, создаем `div` с классом

если строка `selector` начинается с решетки `#` то создаем параграф с `id`

пример:

если передана строка `'.block'`, то функция конструктор создает элемент с `class= «block»`

если передана строка `'#best'`, то функция конструктор создает элемент с `id =best»`

с помощью `cssText` задавать стили:

высотой - `height`,

шириной - `width`,

`background` – `bg`

размер текста - `fontSize`

Внутри созданного блока записывать любой текст. Метод записи может быть любым.

2) Создать новый объект на основе класса `DomElement`.

3) Вызвать его метод чтобы создать элемент на странице.

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в отдельном репозитории на `GitHub`.

1) Используя функцию-конструктор `DomElement` из основного задания №1, создать квадрат 100 на 100 пикселей. Ему необходимо задать фон(`background`) любого цвета и свойство `position: absolute`.

2) Поместить его на страницу только после выполнения события `DOMContentLoaded`. Внутри тега `body` должно быть только подключение скрипта. (В случае подключения файла скрипта к странице перед закрывающим тэгом `body`).

3) Написать обработчик события для `keydown`, который будет принимать `callback`-функцию. Данная функция будет отлавливать нажатие на стрелки клавиатуры. В зависимости от нажатой кнопки (Вверх - стрелка вверх, Влево - стрелка влево, Вправо - стрелка вправо, Вниз - стрелка вниз) наш квадрат будет перемещаться на 10 пикселей при каждом нажатии. Добавить папку с уроком на свой `GitHub`.

Задание к уроку «Особенности современного стандарта ES6»

Напишите расширения метода прототипа:

1) Два класса, `First` и `Second`, `Second` наследует от `First`.

2) В `First` есть метод `hello` - он печатает в консоль «Привет я метод родителя!».

3) Нужно написать в `Second` метод `hello`, чтоб он сначала вызывал метод `hello` из `First`, а потом сразу печатал в консоль «А я наследуемый метод!»

Проверить, чтобы все работало без ошибок в консоли. Запустить задание в репозиторий на `github` или реализовать на доске `CodePen` и прикрепить ссылку.

Напишите простым языком, чему вы научились!

Представьте, что ваше сообщение прочитает 8-ми летний ребенок и он должен понять, чему вы научились. Это называется метод Фейнмана.

Решить задачи в `CodePen`

Ссылки прикрепите в соответствующие поля ниже

1) Калькулятор

The screenshot shows a Codepen editor with tabs for HTML, CSS, and JS. The JS tab is active, containing the following code:

```
<input type="text" id='a'>
<input type="text" id='b'>
<button id="sum">сумма</button>
<button id="mult">произведение</button>
<input type="text" id='res'>
<div class="task">
  Решить задачу где возможно используйте <b>this</b>
<br>
  В объект можно добавлять свои свойства<br>
  Есть объект calculator с тремя методами:
  <ul>
```

The result area shows the rendered HTML with two input fields, two buttons labeled 'сумма' and 'произведение', and a third input field. Below the inputs, the text reads: 'Решить задачу где возможно используйте this. В объект можно добавлять свои свойства. Есть объект calculator с тремя методами:'. A list item is visible: '• sum() возвращает сумму этих двух значений'. The interface includes a 'Resources' tab, zoom controls (1x, 0.5x, 0.25x), and a 'Rerun' button.

2) Функция расчета

The screenshot shows a Codepen editor with tabs for HTML and JS. The JS tab is active, containing the following code:

```
Написать функцию которая принимает 2 целых числа
<b>x</b> и <b>y</b>
<br>
Функция должна вычислять сумму цифр результата
<b>x</b> в степени <b>y</b>.
<br>
Например вызов<br>
getResult(4, 8) --> 25 <br>
4<sup><small>8</small></sup> = 65536, а сумма его цифр составляет 6 + 5 + 5 + 3 + 6 = 25.
<br>
```

The result area shows the rendered text: 'Написать функцию которая принимает 2 целых числа x и y. Функция должна вычислять сумму цифр результата x в степени y. Например вызов getResult(4, 8) --> 25. 4⁸ = 65536, а сумма его цифр составляет 6 + 5 + 5 + 3 + 6 = 25.' Below this, it says 'Форкните доску чтобы её сохранить у себя на доске'. The interface includes a 'Resources' tab, zoom controls (1x, 0.5x, 0.25x), and a 'Rerun' button.

3) Выбор стран и городов

The screenshot shows a Codepen editor with tabs for HTML, CSS, and JS. The JS tab is active, containing the following code:

```
<select name="country" id="country">
  <option value="rus">Россия</option>
  <option value="uk">Украина</option>
  <option value="bel">Беларусь</option>
  <option value="jap">Япония</option>
</select>
<select name="city" id="city"></select>
<div class="result"></div>
<iframe width="560" height="315">
```

The result area shows a dropdown menu with 'Россия' selected, a second empty dropdown menu, and a large image of a cartoon character with a red bow tie. The text 'Codepen countries' is visible above the image. The interface includes a 'Resources' tab, zoom controls (1x, 0.5x, 0.25x), and a 'Rerun' button.

Задание к уроку «Модульная структура»

Начиная с этого занятия мы начинаем разработку нового проекта. Данное задание требует отдельного репозитория с названием, например, 3DGIo - название самой страницы, но вы

можете выбрать любое удобное для вас название. Дополнительно я рекомендую так же сохранить начальную сборку webpack в еще один дополнительный репозиторий с названием, к примеру, webpack. Далее мы будем его настраивать и улучшать, а если вы будете держать его в отдельном репозитории вам удобнее будет использовать стартовую сборку для других проектов!

1. Собрать начальную сборку проекта с помощью webpack
2. Создать два модуля с названиями one.js и two.js
3. Подключить оба модуля к входному файлу index.js
4. Каждый модуль должен выводить в консоль свое название
5. Перед push на GitHub собрать проект (должна появиться папка dist и в ней файл main.js) и подключить минифицированный скрипт к index.html.

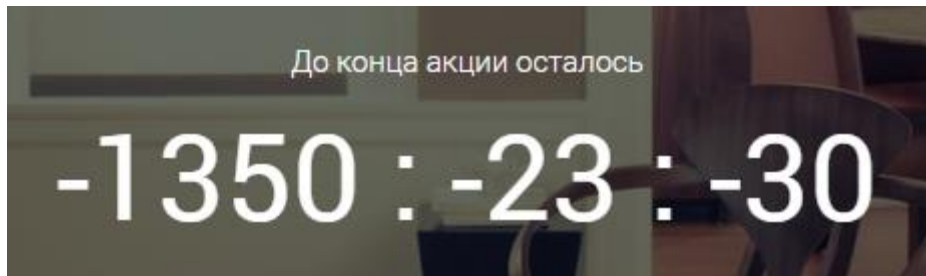
Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в отдельном репозитории на GitHub.

Задание к уроку «Скрипты и время выполнения»

- 1) Написать таймер обратного отсчета по видео.
- 2) Переписать таймер с помощью setInterval. Будьте внимательны, чтобы setInterval не вызывал сам себя, не превращался в рекурсивный вызов!

Проверить себя легко. Выведите что-нибудь в консоль внутри функции updateClock. Вывод в консоль должен происходить 1 раз за 1000ms.

- 3) Проверить, чтобы все работало и не было ошибок в консоли
- 4) У таймера есть проблема (нужно исправить):
 - Если дата уже прошла, а мы заходим на страницу, то получим



- 5) Изменить скрипт так, чтобы в таком случае выводилось: 00:00:00
 - Необходимо подставлять 0 перед значениями, которые состоят из одной цифры (из 4:6:50 делает 04:06:50)

- 6) В отдельной папке создать html и js

Вывести текущий день и время на страницу в таком формате

Добрый день (утро, вечер, ночь в зависимости от времени суток)

Сегодня: Понедельник

Текущее время: 12:05:15 PM

До нового года осталось 175 дней

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

Создайте отдельную HTML страницу с полем ввода (тэг input) и пустым параграфом (тэг p) Стилизация на ваше усмотрение.

Введённый в поле текст должен отображаться внутри параграфа, но с задержкой в 300мс. При этом каждый введённый пользователем в поле символ сбрасывает предыдущий отложенный вызов и запускает новый.

Таким образом программа должна ожидать завершения ввода пользователя и только после этого изменять текст в параграфе (тэг p)

Ссылки в помощь

Микропаттерны оптимизации в Javascript

Что такое Throttling и Debouncing?

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в отдельном репозитории на GitHub.

1) Задание в Codepen

2) На отдельной HTML странице!!!

Написать любую анимацию, используя requestAnimationFrame и кнопку, активирующую её. Кнопка должна ставить анимацию на паузу и продолжать анимацию после повторного нажатия на нее. Добавить кнопку reset, которая будет возвращать анимацию в первоначальное состояние.

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в отдельном репозитории на GitHub.

Задание к уроку «Параметры документа, окна и работа с ними»

1) Написать скрипт Меню и модального окна по видео

2) Написать анимацию появления модального окна (не меню!!!)

Использовать JS анимацию. Использовать нативный JavaScript. Использование сторонних библиотек запрещено!

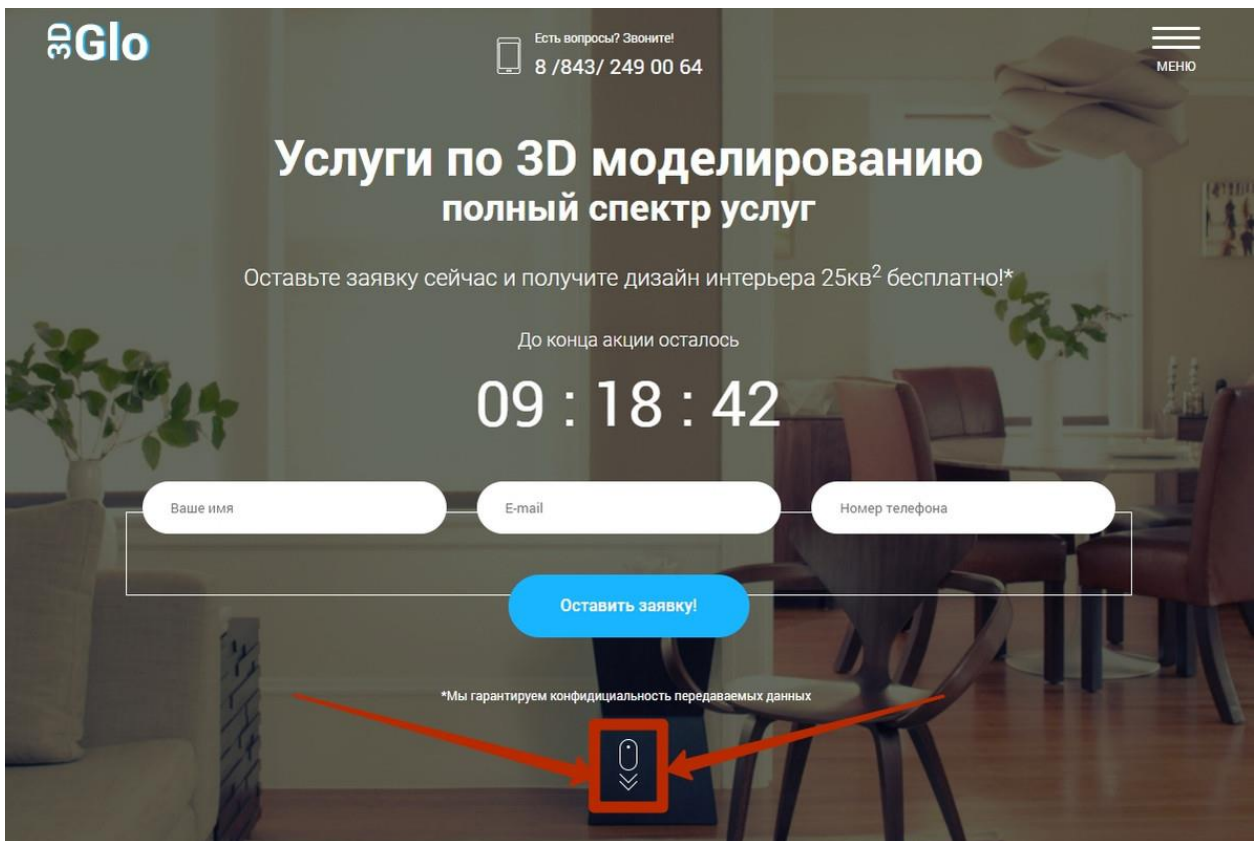
Необходимо манипулировать элементами посредством JS. CSS анимация не подходит для анимирования модального окна.

3) Если пользователь заходит на сайт с устройства, у которого ширина экрана меньше 768px (мобильного устройства) - анимация отключается

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

1) Написать скрипт плавной прокрутки страницы при клике на элементы меню, используя чистый JS. Данную задачу можно решить несколькими способами. Как пример - requestAnimationFrame или методом scrollToView (второй вариант красивее, но работает не во всех браузерах)

2) На первом слайде так же есть КНОПКА, нажав на которую мы должны плавно переместиться на следующий слайд



Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

Задание к уроку «Регулярные выражения»

1) В калькуляторе разрешить ввод только цифр:

После выполнения необходимо проверить работоспособность SELECT - при выборе любого значения текст option должен отображаться.

2) У нас на странице есть 3 формы (первый экран, последний экран и модальное окно). Необходимо валидировать (проверять введенное значение на допустимые символы) поля ввода всех форм:

В поля ввода `type=text` и `placeholder= «Ваше сообщение»` позволить ввод только кириллицы в любом регистре, дефиса и пробела.

В поля ввода `type=email` позволить ввод только латиницы в любом регистре, цифры и спецсимволы: @ - _ . ! ~ * ' "

(Собака , Дефис , Подчеркивание , Точка , Восклицательный знак , Тильда , Звездочка , Одинарная кавычка)

В поля ввода `type=tel` позволить ввод только цифр, круглых скобок и дефис.

Решить задачи в CodePen

Реализовать проверку введенных данных в поля ввода с помощью события blur (при потере полем ввода фокуса) и заменять их на корректные при необходимости по правилам:

Должны удаляться все символы, кроме допустимых

Несколько идущих подряд пробелов или дефисов должны заменяться на один.

Пробелы и дефисы в начале и конце значения должны удаляться.

Для поля type=text Первая буква каждого слова должна приводиться к верхнему регистру, а все остальные — к нижнему.

Задание к уроку «Делегирование»

- 1) Реализовать табы по видео.
- 2) В функции toggleMenu() много обработчиков событий. Используя делегирование событий, сделать обработчики для:
Крестика закрытия меню и пунктов меню.
На кнопку меню.
- 3) У вас должно быть максимум 2 обработчика события в функции toggleMenu()
Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

- 1) Написать 1 обработчик для всех событий внутри функции toggleMenu()
- 2) Реализовать следующий функционал: если клик произошел мимо меню, оно закрывается
Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

- 1) Скачать дополнительный файл с версткой Rabbits.zip

- 2) Выполнить работу по следующему ТЗ:

На каждой клетке вы видите кнопки с направлением, напишите скрипт, чтобы при клике на кнопку, клетка менялась с соответствующей клеткой своим положением, соответственно направлению стрелки.

Если вверх, вниз вправо или влево идти некуда, то происходить ничего не должно (со следующей строкой поменяться можно (например, 20 с 21).

При клике на кнопку RESET все клетки должны вставать на свои места

Лезть в вёрстку запрещено, использовать только те классы, что предоставлены

Вызывать перезагрузку страницы запрещено, программа должна работать и сбрасываться без перезагрузок.

Выполняем задание только для десктопной версии, в мобильной не нужно.

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в новый репозиторий на GitHub.

- 1) Реализовать слайдер на сайте по видеоуроку
- 2) Удалить все элементы со страницы с классом dot (из верстки Index.html)

3) Написать скрипт, который будет на страницу добавлять точки с классом `dot` равному количеству слайдов

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

1) Передавать классы для работы слайдера из точки входа `index.js`

2) Если класс слайдера передан неверно и такого элемента на странице нет - завершать работу программы (`return`) (что бы не было ошибок с подключенным модулем слайдера)

3) Если класс каждого слайда передан неверно и таких элементов на странице нет - завершать работу программы (`return`) (что бы не было ошибок с подключенным модулем слайдера)

4) Для классов активности предусмотреть значения по умолчанию (`slide-active` и `dot-active` к примеру)

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

1) Реализовать калькулятор на сайте по видео.

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

1) Написать эффект изменения общей стоимости на чистом JS - перебор цифр

Пример как должно выглядеть – КЛИК.

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

Задание к уроку «Функции помощники»

1) Создать в проекте `helpers.js`

2) В `helpers.js` добавить функцию `animate`

3) Перевести анимацию модального окна на использование функции-помощника `animate`

1) Анимировать накрутку стоимости в калькуляторе через функцию-помощник `animate`

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

Задание к уроку «Promise. Fetch»

1) Написать две функции для реализации AJAX запросов: `getData` и `sendData`

3) Реализовать получение данных из файла `.json` (прикреплен под видео) через функцию `getData`

4) После получения объекта из файла `.json` должна произойти отправка данных (которые мы получили из файла `.json`) на URL через функцию `sendData`

`https://jsonplaceholder.typicode.com/posts`

5) Ошибки должны быть обработаны

При загрузке страницы сперва должно произойти получение данных из файла и после этого сразу отправка.

Проверить, чтобы все работало и не было ошибок в консоли. Для этого задания создайте отдельный репозиторий.

Реализовать отправки данных на <https://jsonplaceholder.typicode.com/posts> методом XMLHttpRequest.

Проверить, чтобы все работало и не было ошибок в консоли. Для этого задания создайте отдельный репозиторий.

Задание к уроку «Async/Await, Try/Catch»

1) Написать модуль отправки данных формы по видео

2) Подключить скрипт отправки данных к:

Модальному окну

Контактной форме в самом низу страницы

3) Сделать валидацию данных при отправке:

В поля name= «user_phone» разрешить ввод только цифр, знака «+», круглых скобок и дефис

В поля name= «user_name» разрешить ввод только кириллицы и пробелов

В поля name= «user_message» разрешить только кириллицу, пробелы, цифры и знаки препинания.

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

1) Вместо текстового оповещения statusBlock при отправке использовать прелоадер картинку или анимацию «пример»

Проверить, чтобы все работало и не было ошибок в консоли. Сохранить проект в репозитории на GitHub с нашим проектом.

1) Реализовать функционал выбора автомобилей и показа информации для каждого. Пример в видео Cars

2) Реализовать получение данных с помощью Promise/Fetch

3) Обязательно обработать ошибки

Проверить, чтобы все работало и не было ошибок в консоли. Для этого задания создайте отдельный репозиторий.

1) По ссылке, копировать себе проект!

2) Прокомментировать каждую строчку кода (что именно происходит в строке)

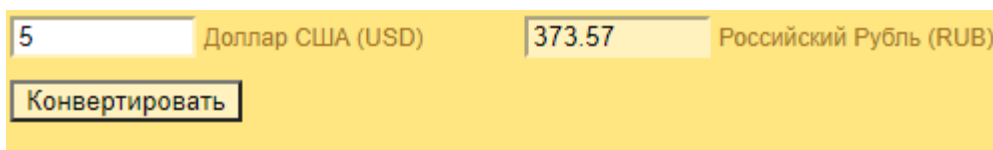
Пример работы приложения в видео EvalTryCatch

Проверить, чтобы все работало и не было ошибок в консоли. Для этого задания создайте отдельный репозиторий в ветку comments.

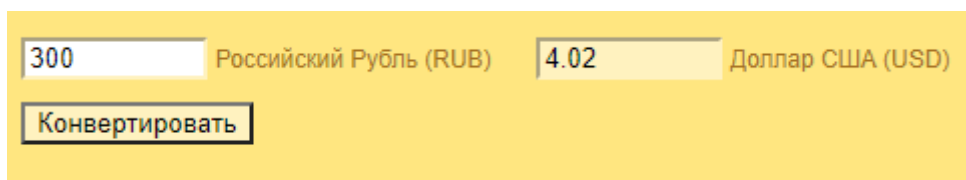
1) В отдельном репозитории реализовать приложение конвертер валют

2) С помощью API - <https://exchangeratesapi.io/> получить стоимость доллара и евро (не обязательно использовать этот api, есть и другие)

- 3) На странице должен быть select или radio кнопки с выбором валюты USD или EUR
- 4) Добавьте на страницу input, вводим количество валюты и получаем количество рублей
пример



- 5) Так же возможность конвертировать обратно из рублей в валюту



- 6) Дизайн и исполнение на ваше усмотрение, главное, чтобы пользователь мог в поле ввода ввести текст, нажать кнопку и получить перевод своего текста.

Проверить, чтобы все работало и не было ошибок в консоли. Для этого задания создайте отдельный репозиторий.

Задание к уроку «Админ-панель, json-server, серверные запросы»

- 1) Повторить код за уроком
- 2) Оптимизировать работу с методом fetch(), а именно:
создать два метода для получения и отправки данных в нашем сервисе
все методы должны передавать в эти методы URL и все остальные необходимые данные
обязательно обработать ошибки
если у нас произошла ошибка (спровоцировать можно путем редактирования самого запроса, добавить банальную ошибку в сам URL) выводить под таблицей надпись «Произошла ошибка, данных нет!»
- 3) По желанию эти методы реализовать через async/await и try/catch
Проверить, чтобы все работало и не было ошибок в консоли. Для этого задания создайте отдельный репозиторий.

Итоговый проект: выполнить необходимый функционал согласно ТЗ в течение 7 календарных дней.

2.5. Материально-техническое обеспечение образовательного процесса

Реализация образовательной программы осуществляется с применением исключительно электронного обучения, дистанционных образовательных технологий, организуя учебные занятия в виде онлайн-курсов, обеспечивающих для обучающихся достижение и оценку результатов обучения путем организации образовательной деятельности в электронной информационно-образовательной среде, к которой предоставляется открытый доступ через информационно-телекоммуникационную сеть «Интернет». Взаимодействие с обучающимися осуществляется с использованием платформы для организации аудио и видеоконференций Яндекс.Телемост.

Обучающимся предоставляется доступ к ресурсу дистанционного обучения через интернет-платформу Геткурс. На интернет-платформе размещены видео-лекции, а также необходимый справочный материал.

В процессе обучения можно использовать электронно-библиотечные ресурсы Электронной библиотеки «Научная электронная библиотека».

Учет и хранение результатов образовательного процесса осуществляется в электронно-цифровой форме в соответствии с требованиями Закона Российской Федерации от 21 июля 1993 года № 5485-1 «О государственной тайне», Федерального закона от 27 июля 2006 года № 152-ФЗ «О персональных данных», Федерального закона от 22 октября 2004 года № 25-ФЗ «Об архивном деле в Российской Федерации».

Обучающийся осваивает образовательную программу удаленно, взаимодействуя с педагогическим работником исключительно посредством цифровых образовательных сервисов и ресурсов электронной информационно-образовательной среды. Непосредственное взаимодействие обучающегося с педагогическим работником в аудитории не производится. В течение всего периода обучения для обучающихся созданы условия получения доступа к электронной информационно-образовательной среде.

Рабочее место педагогического работника оборудовано персональным компьютером и компьютерной периферией (веб-камерой, микрофоном, аудиокolonками и(или) наушниками).

В состав программно-аппаратных комплексов включено (установлено) программное обеспечение, необходимое для осуществления учебного процесса (операционная система (операционные системы), офисные приложения, средства обеспечения информационной безопасности, архиваторы, графический, видео- и аудио-редакторы).

При реализации программы созданы условия для функционирования электронной информационно-образовательной среды. Данная среда способствует освоению обучающимися программы в полном объеме независимо от места нахождения обучающихся.

2.6. Кадровые условия обеспечения образовательного процесса

В соответствии с профессиональным стандартом «Педагог дополнительного образования детей и взрослых», утвержденным приказом Министерства труда Российской Федерации от 22 сентября 2021 года № 652н, право на занятие педагогической деятельностью имеют лица, имеющие среднее профессиональное или высшее образование, а также лица, обучающиеся по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности дополнительных общеобразовательных программ, и успешно прошедшие промежуточную аттестацию не менее чем за два года обучения.

Порядок допуска лиц, обучающихся по образовательным программам высшего образования, к занятию педагогической деятельностью по общеобразовательным программам утверждён приказом Министерства просвещения Российской Федерации от 18 сентября 2020 года № 508.

2.7. Список литературы

1. Водопьянов С.Э., Симонов И.Н., Гаев Л.В. Области применения javascript: расширение возможностей веб-разработки/ Академическая публицистика. 2023. № 5-2. С. 40-43. https://elibrary.ru/download/elibrary_53823159_26838967.pdf
2. Гаврильев Э.И., Авдеев Т.В. Модель оценки квалификации it-специалистов в области разработки программного обеспечения/ Безопасность цифровых технологий. 2021. № 2 (101). С. 67-84. https://elibrary.ru/download/elibrary_46295367_23801247.pdf
3. Государев И.Б. Электронное обучение веб-программированию на основе технологий тестирования javascript-сценариев/ Педагогический опыт: теория, методика, практика. 2015. № 2. С. 71-73. https://elibrary.ru/download/elibrary_24039213_90789010.pdf
4. Иванов А.Н., Шегай Л.Ю. Метод разработки веб-приложения с использованием современных технологий на основе языка программирования javascript/ Ученые заметки ТОГУ. 2020. Т. 11. № 4. С. 267-274. https://elibrary.ru/download/elibrary_45716243_63428933.pdf
5. Слепцова Е.А., Соколова А.П. Основы программирования на javascript/В сборнике: EUROPEAN RESEARCH. сборник статей XIII Международной научно-практической конференции: в 2 частях. 2017. С. 135-139. https://elibrary.ru/download/elibrary_45716243_47712991.pdf
6. Сугаипов С.А.А., Гериханов З.А. Анализ производительности алгоритмов сортировки в javascript/ Тенденции развития науки и образования. 2023. № 98-10. С. 130-132. https://elibrary.ru/download/elibrary_54207994_31135995.pdf
7. Трубицына В.С., Снегин О.В. Javascript: современные возможности интерактивности веб-сайтов организаций/ В сборнике: Донецкие чтения 2023: образование, наука, инновации, культура и вызовы современности. Материалы VIII Международной научной конференции. Донецк, 2023. С. 359-361. https://elibrary.ru/download/elibrary_54979447_15185220.pdf
8. Халимов Н.Н. Обзор графических javascript-библиотек для создания интерактивных программ/В сборнике: Научные исследования: теория, методика и практика. сборник материалов V Международной научно-практической конференции. 2019. С. 170-171. https://elibrary.ru/download/elibrary_36978355_15079476.pdf