АНО «Развитие человеческого капитала»
Обособленное подразделение «Техноград»

УТВЕРЖДЕНО
Приказом Исполняющего обязанности
генерального директора
АНО «Развитие человеческого капитала»
№ ОД-99/24 от «20» сентября 2024 г

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

Наименование программы

«JavaScript. Продвинутый уровень»

Категория слушателей: студенты образовательных учреждений СПО, обучающиеся по специальности «Информационные системы и программирование», программисты приложений, разработчики сайтов, Web, мобильных и других мультимедийных приложений, лица старше 18 лет, заинтересованные в получении новых компетенций

Объем: 24 часа.

Форма обучения: очная с применением дистанционных технологий.

Форма документа, выдаваемого по результатам освоения программы: сертификат о прохождении курса.

УЧЕБНЫЙ ПЛАН.

Наименование компонентов программы	Обязательные аудиторные учебные занятия (ч.)		Внеаудиторная (самостоятельная) учебная работа (ч.)	Всего учебной нагрузки (ч.)
	всего	в т.ч. практических и семинарских занятий		
Тема 1. Концепция асинхронного программирования.	**4**	3	-	**4**
Тема 2. Функциональное программирование.	**4**	3	-	**4**
Тема 3. Основные принципы работы в объектно-ориентированном программировании.	**4**	3	-	**4**
Тема 4. Работа с Node.js.	**4**	3	-	**4**
Тема 5. Разбор JavaScript.Dom. Навигация по HTML-элементам.	**4**	3	-	**4**
Тема 6. Работа с базами данных на JavaScript.	**4**	3	-	**4**
ИТОГО:	**24**	**18**	**0**	**24**

СОДЕРЖАНИЕ ПРОГРАММЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, РАЗДЕЛАМ (МОДУЛЯМ).

Наименование тем, разделов (разделов)	Содержание тем, разделов (модулей)
Тема 1. Концепция асинхронного программирования.	Синхронное программирование (очередное выполнение блоков кода, взаимозависимых друг от друга), асинхронное: отложенное выполнение блока (с задержкой) без влияния на остальные блоки кода, конструкция Promise (свойства state и result, аргументы resolve и reject), обработчики ошибок и правильного выполнения (методы then и catch), очистка (finally, остановка индикаторов загрузки, закрытие ненужных соединений). Конструкция Async / await (возвращение Promise), «синтаксический сахар» (упрощение кода), функция fetch, асинхронные функции классов. **Практическое задание № 1. Создать Promise, который при успешном выполнении будет с задержкой 3 секунды выводить «OK», при ошибке выводить «Error».**
Тема 2. Функциональное программирование.	Парадигмы языков программирования, набор правил при написании кода. Понятие функционального программирования. Парадигма декларативного программирования, императивного программирования. Описание характеристик функционального программирования (функции, функции высшего порядка, чистые функции, идемпотентная функция, побочные эффекты). Первый класс функций, функции более высокого порядка, отрицательное значение изменчивости и побочных эффектов, открытие своего кода для ссылок. Дополнительные возможности функционального программирования: рекурсия, каррирование, частичное применение, композиция функций. Вызов функции из самой функции, базовый случай. Повторное использование функции с изменением параметров, арность функции. Применение нескольких аргументов к одной функции одновременно. Превращение двух и более функций в одну. **Практическое задание № 2. Написать рекурсивную функцию, которая вычисляет n-ое число фиббоначи.**
Тема 3. Основные принципы работы в объектно-ориентированном программировании.	Наследование классов (extends, «родительский» и «дочерний» классы). Solid (принципы: единственной ответственности, открытости-закрытости, подстановки Барбары Лисков, разделения интерфейса, инверсии зависимостей). Абстракция как объект зависимости модулей верхнего и нижнего уровней, деталей. Прототипное наследование (перенос части свойств, методов), свойство Prototype (равняется null или ссылке на другой объект), использование прототипа. Свойства-аксессоры. Цикл for…in для вывода собственных и унаследованных ключей. Фильтрация свойств на основании метода obj.hasOwnProperty. **Практическое задание № 3. Создать класс Figure со свойством figurename, от которого будут наследоваться два класса: Square и Rectangle.**
Тема 4. Работа с Node.js.	Серверная часть: Ajax (связь между клиентом и сервером, взаимодействие с сервером без обновления страницы), http запрос-ответ. Метод Fetch, использование response. Заголовки в браузере (получение одного заголовка, перебор всех заголовков, headers, установка заголовка).

	Виды запросов (get – чтение данных с сайта, post – отправка данных на сайт, delete – удаление данных), fetch-параметры. Отправка изображений. Полноценное написание сервера (многостраничная функция, некорректный запрос create server). Написание клиента (HTML, главная страница). **Практическое задание № 4. Создать асинхронную форму GetUsers (names), которая получает на массив логины пользователей GitHub.**
Тема 5. Разбор JavaScript.Dom. Навигация по HTML-элементам.	Навигация по DOM-элементам (объект document). Работа скрипта HTML через DOM. Структура DOM (HTML). Правила ChildNodes: FirstChild, LastChild. Родительский и дочерний тэги, Sibling. GetElement, QuerySelector. Классы узлов: EventTarget (корневой класс), Node (класс для DOM-узлов), Document (документ в целом), CharecterData (абстрактный класс), Text (текст внутри элемента), Comment (класс для комментариев), Element (базовый класс для DOM-элементов), HTMLElement (базовый класс для HTML-элементов). NodeName / TagName. TextContent, outerHTML. DOM-свойства: myData. HTML-атрибуты, создание элементов, их внедрение. CSS стили в DOM, className, GetComputedStyle. **Практическое задание № 5. Сформировать HTML-страницу с помощью js, сделать в ней тэг со списком li и еще три тэга.**
Тема 6. Работа с базами данных на JavaScript.	Описание инструментов работы с PostgreSQL. Создание базы данных (Query Tool), подготовка и создание таблицы, работа с данными внутри таблицы (insert into, select), удаление таблицы (drop table). Связи между таблицами (один к одному, один к многим, многие ко многим). Удаление и изменение данных в таблице. Работа в postgresql.js, ввод информации для подключения к базе данных, создание скрипта кода, таблицы. Работа в MongoDB, создание коллекции, её функции (получение всех записей коллекции, получение записи по параметру, получение первых трех записей, сортировка значений по возрастанию или убыванию). **Практическое задание № 6. Создать таблицу «Animal» с полями type и color.**