1. Через script написать «Hello world!»
2. Дан массив, в виде класса Student. Имеет свойства: ФИО, Год рождения(число), Учебная группа, Средняя оценка(натуральное). Причем массив Mas содержит три поля с такими же массивами Student, но в каждом из них разные данные (aka Двойной массив в итоге). Пробежитесь по ученикам и найдите:   
   - Где данные не заполнены (?? - оператор)  
   - Найдите лучшего по оценке (условия)  
   - Преобразуйте год рождения лучшего в строку и выведите «Ученик \*ФИО\* родился в \*год\* и получил \*Оценка\*»
3. Работа с функциями:  
   - Создайте fun1, которая складывает a и b , а затем выводит результат в return.  
   - Создайте fun2, которая работает также как fun1, но в return подается функция hz. (функция внутри функции)  
   - Создайте fun3, которая выводит массив из функций DA и NET. Каждая из них выводит соответствующее сообщение на экран. Примените обе эти функции, обращаясь к fun3.  
   - Создайте fun4, которая проверяет число a на вхождение в интервал [c,d]. А вот c и d являются сокращенными функциями (стрелочная и анонимная). Заполнить функции c и d можно чем угодно.
4. Берем данные задания 2. Добавляем новое свойство Оценка поведения.  
   Создайте именно уже объект. Создайте в нем конструктор, который дает изначальные данные (какие угодно, в пределах разумного). Оценка поведения должна быть закрыта локально в переделах класса и должен быть метод получения значения этого свойства (getSocialMark). Создайте новый класс BestStudent, который наследует класс Student, но содержит метод Run (Прибавляет среднюю оценку и социальную, а затем выводит на экран). Выведите для каждого ученика «Ученик \*ФИО\* имеет оценку \*Run()\* и социальную \*Соц. оценка\*».
5. Изучите различные области видимости при помощи ключевого слова this.  
   (Общее, класс, функции и т.д.)
6. Вы знаете уже, что такое массивы и что такое объекты. Теперь поймите, что класс это тоже самое, но имеющее свои примочки. Переделайте задачу 4 уже именно под класс.
7. Поиграйте немного со строками и массивами. Переход от одного состояния в другое.  
   Желательно делать через функции.
8. Выведите на экран дату и попробуйте разные функции из объекта Math.
9. BOM. Это работа с браузером. Иногда она может быть полезна, если некоторое действие не было завершено или требуется подтверждение. Пусть через 5 секунд после захода на сайт мне вылезло сообщение «Покиньте страницу».
10. DOM. Это уже работа с документом, т.е со страницей. Изменение страницы с помощью кода. Попробуйте как-то изменить страницу с помощью кода (хотя бы 3 элемента и разными способами). Включите в список также элементы с разными событиями (хотя бы 2).
11. Формы. Это элементы windows: кнопки, текст, поля. Полезно если пользователю нужно ввести какие-то данные. Создайте эти примитивные элементы и проверьте операции: сложения, вычитания и умножения (в полях данные, кнопка активирует, а текст выводит).

===

1. На моем «сервере» есть функция fun, которая делит один аргумент на другой нацело (отбросить дробную часть). Подключение к серверу настолько плохое, что среднее время подключения к серверу сопоставимо с вероятностью быстрого подключения в 0.2. Остальные 0.8 сервер отсылает ошибку «Превышено время подключения» и повторяет запрос через 3 сек. До тех пор, пока не подключиться. После подключения вы должны активировать функцию fun и вывести ответ. (Внимание: В задаче не надо использовать сервер. В данном контексте работу сервера будет обеспечивать клиент,т.е. функция fun у вас будет в коде. Ваша задача – правильно использовать async и await + отловите ошибку через try-catch-throw)
2. Итоговая задача. Реализация минимальной ИС с малой симуляцией работы сервера.  
   Пусть есть сервер с вероятностью подключения в 80% с повтором через 1 сек.  
   Во время каждого успешного подключения отсылается 1 команда с моментальным ответом клиенту.  
   На сервере хранятся данные об учениках в виде объекта (можно и класс). Данные такие:  
   ФИО(строка), Группа (строка), Стипендия (число), Оценки(Массив чисел из 4 оценок: Мат.Анализ,Философия,Программирование и Англ.язык) и Средняя оценка(число типа float). Реализуйте базовые действия с данными со стороны клиента: Добавить запись, Изменить запись и Удалить запись. А так же напишите функцию, которая при нажатии на кнопку возьмет данные с сервера, подсчитает среднюю оценку (изначально пусто), определит по этой оценке стипендию, запишет эту стипендию и отправит эти данные обратно на сервер. Можно писать доп. функции. Нельзя использовать массив вместо объекта или класса.   
   Доп. инфа:  
   4,5>= -5k;

3,5>= -4k;

2,5 > -2k;

2,5<-0 **(Напишите функцию показа данных для проверки. Можно моментально. При повторном подключении высылать сообщение об ошибке.  
Данная задача имитирует работу сервера и написание API для работы.  
Вы овладеете базовым знанием по работе с алгоритмами, группировкой данных, написанием интерфейса, работе с задачами со временем. Все остальное будет лишь увеличиваться в объемах работы и некоторых новых технологиях.)**