1. Data Types?    
   Answer: string, number, Boolean, undefined, null, BigInt, Symbol, Object.
2. What are primitive data types?   
   Answer: типы данные которые копируются по значению.
3. What are the possible ways to create objects in JavaScript?   
   Answer: Скобочная нотация, через ключевое слово 'new', Object.create()
4. What is undefined property?   
   Answer: свойство отсутствия значения
5. What is null value?   
   Answer: присваевоемое пустое значения;
6. What is the difference between null and undefined?    
   Answer: null - присваевоемое значени, undefined - присваивается в случае если переменной не задано значение
7. What is the difference between window and document?   
   Answer:    
   window - глобальный объект браузера и корень DOM так же можно назвать окном браузера   
   document - другой документ загруженный в браузер(htmp, php);
8. What is isNaN?    
   Answer: метод JS возвращающий булевское значение при проверке “на число”. Пареобразует значение в число и проверяет
9. What is the difference between let, const and var?   
   Answer:    
    var - имеет глобальную область видимости, хостится, можно перезаписать.   
      let - имеет блочную область видимости, не хостится, можно перезаписать   
    const - имеет блочную область видимости, не хостится, нельзя перезаписать, нужно определять сразу.
10. What are the differences between undeclared and undefined variables?   
    Answer:    
     undefined - переменная объявлена, но ей не присвоено значение   
       undeclared - происходит когда переменная не объявлена и мы пытаемся получить к ней доступ
11. What are global variables?   
    Answer: переменные объявленные в глобальной области видимости
12. What are the problems with global variables?   
    Answer: переменные можно перезаписать из любого участка кода
13. What is NaN property?   
    Answer: Свойство глобального объекта “не число”
14. What are classes in ES6?   
    Answer: Синтаксический сахар для создания прототипов
15. How do you check whether a string contains a substring?   
    Answer: использовать метод includes
16. How do you check if a key exists in an object?   
    Answer: *obj*.hasOwnProperty( *prop* ) вернет булевое значение
17. How do you copy properties from one object to other?   
    Answer:    
    - мы можем сделать неглубокое копирование через 'for in'   
    - можно использовать метод Object.assign() - принимает целевой объект в качестве первого аргумента (куда будут копировать свойства) и исходные объекты(откуда будут скопированны свойства)
18. What is a proxy object?    
    Answer:
19. What is the main difference between Object.values and Object.entries method?   
    Answer:    
    - Object.values - вернет массив значений   
    - Object.entries - возвращает массив, элементы которого будут массивами пары ключ-значение из объекта к которому применяется метод
20. How can you get the list of keys of any object?   
    Answer: Object.key()
21. How do you create an object with prototype?   
    Answer: Мы можем использовать метод Object.create(prototype)    
     prototype - объект-прототип, свойства которого мы наследуем
22. What is a WeakSet?   
    Answer:
23. What are the differences between WeakSet and Set?   
    Answer:
24. What is a WeakMap?   
    Answer:
25. What are the differences between WeakMap and Map?   
    Answer:
26. What is Hoisting?    
    Answer: Возможность доступа к переменной или функции объявленной с помощью Function Declaration до ее объявление
27. How do you assign default values to variables?   
    Answer: Через присваивание. Знак “=”

**Context, Scope proto and Prototype**

1. What is the difference between Call, Apply and Bind?   
   Answer:    
   call - передаем аргументы строкой   
   apply - передаем аргументы массивом   
   bind - создает функцию-обертку которую можно вызвать в другом месте кода
2. What is scope in javascript?   
   Answer: это доступность переменных, функций, объектов и т.д. в текущей области видимости. Условно - переменные, методы и т.д. к которым мы можем получить доступ в теле функции
3. What is the difference between proto and prototype?   
   Answer:    
   - proto - внутреннее свойство объекта, указывающее на его прототип. Сыылка на объект-прототип   
   - prototype - prototype является свойством объекта Function
4. What is prototype chain?   
   Answer: Каждый объект имеет ссылку на свой объект-прототип. При попытке получить доступ к какому-то свойству, движок сначала ищет в целевом объекте, потом в объекте прототипе и так далее. Это и будет цепочка прототипов. Наверное можно определить как - возможность получать доступ к свойствам объекта-прототипа

**JSON**

1. What is JSON and its common operations?   
   Answer: JSON - объект JS/Текстовый формат обмена данными. Используется в REST-запросах.
2. How do you parse JSON string?    
   Answer: JSON.parse()
3. Why do you need JSON?   
   Answer: Использовать в запросах. Передавать.получать данные

1. How do you define JSON arrays?   
   Answer:    
   {   
   "jsonArray": ["1", "some string"]   
   }
2. What is the purpose JSON stringify?   
   Answer: Для преобразования объектов в JSON

**Array methods**

1. What is the purpose of the array slice method?   
   arr.slice([start], [end])   
   Answer: Вернет новый массив в который скопирует элементы начиная от  аргумента 'start' заканчивая "end". Без аргументов создаст копию массива.
2. What is the purpose of the array splice method?   
   Answer: С его помощью можно: заменить, удалить, добавить элементы в массив.
3. What array methods do you know?   
   Answer: push, pop, shift, unshift, reverse, every, some, filter, find, etc.   
    forEach, map, reduce
4. What is the difference between Array.forEach() and Array.map()?   
   Answer: Array.map() - возвращает новый массив   
    Array.forEach() - мутирует массив

**Functions**

1. What are lambda or arrow functions?   
   Answer: Функции-стрелки, появились в ES6, имеют сокращенный синтаксис и свои отличия. Основные: не имеют свойства arguments, игнорируют call, apply, bind, не имеют своего this - берут его из окружения, не могут быть конструкторами или генераторами и т.д
2. What is a first class function?   
   Answer: Насколько я понимаю, в JS выражения написанные как FE можно называть функциями первого класса. ТО есть - это функция, котрую мы можем присвоить переменной или переиспользовать в другом месте кода, но не имеющая никаких побочных эффектов
3. What is a first order function?   
   ???
4. What is a higher order function?   
   Answer: Функция принимающая в качестве аргументов функцию и/или возвращающая функцию
5. What is a unary function?   
   Answer: функция принимающая один аргумент.    
   Если unary подразумевает мономорфная функция, то это функция которая всегда принимает одни и те же данные
6. What is the currying function?   
   Answer: трансформация функции из функции с множеством аргументов, в множество функций с одним аргументом
7. What is a pure function?   
   Answer: функция принимающая и возвращающая одно и то же значение без каких-либо сайд эффектов
8. What are closures?   
   Answer: Комбинация функции и лексического окружения. Дает доступ к области видимости внешней функции из фнутренней. Под капотом имеет объект с двумя свойствами: Environment record - ссылка на внутреннее окружение   
   Environment outer - ссылка на внешнее лексическое окруужениею   
   Таким образом и имеет возможность обращаться к внешним переменным, функциям и т.д
9. What is IIFE(Immediately Invoked Function Expression)?    
   Answer: функция которая вызывается в момент определения
10. What is a callback function?    
    Answer: Любая функция которая передается как аргумент.
11. What is an anonymous function?   
    Answer: Любая функция, если ее имя не определено после ключевого слова function.

**Async JavaScript**

1. What is a promise?   
   Answer: Специальный объект, который может хранить свое состояние - pending, и два формальных колбека - fulfilled и reject.
2. Why do you need a promise?   
   Answer: для работы с асинхронными дейсвиями. Промис гарантирует что данные из АПИ не будут обрабатываться пока запрос не завершен
3. What are the three states of promise?   
   - pending   
   -fulfilled   
   -reject
4. Why do we need callbacks?   
   Answer: Получить результат или обработать ошибку
5. What is a callback hell?   
   Answer: приходилось писать кучу колбеков для обработки запросов с бека пока не придумали как работать с асинхронными операциями через промисы, в будущем через сахар async/await
6. What is promise chaining?   
   Answer: Возможность строить асинхронные цепочки из промисов
7. What is promise.all?   
   Answer: Метод, принимающий в качестве аргументов массив промисов и возвращающий промис, который выполнится когда выполнятся все промисы в массиве
8. What is the purpose of race method in promise?   
   Answer: получить результатом первый выполнившийся промис
9. What is the use of setTimeout?   
   Answer. Я сомневаюсь в вопросе, поэтому два ответа.    
   Если вопрос в том какая польза от setTimeout, то можно сказать что в некоторых случаях, нам нужно поработать с очередями(например при подгрузке скриптов или сторонних библиотек типа кликров или аналитики) и setTimeout можно использовать чтоб выбросить подгрузку скрипта из потока.    
   Если вопос о том, как использовать, то это просто функция, которая может отложить выполнение какого-либо действия на какое-то время. Удобно использовать при получении нотификаций и отмечать их как прочитанные через определенное время
10. What is the use of setInterval?   
    Answer: Можем использовать при работе с каким-нибудь необычными интеграциями. Если не реализована проверка на коннект с интеграцией на беке, то мы можем стучаться на бек каждый раз через какой-то промежуток времени
11. What is an event loop?   
    Answer: Событийный цикл. Регулирует последовательное исполнение контекстов.   
    Если говорить о полной работе лупа, то мы имеем примерно следующее. Все наши экшены хранятся в хипе. Когда происходит какое-то синхронное действие, событие попадает в Стек и выполняется. Если же событие асинхронное, оно попадает в webAPi, оттуда, в зависимости от приоритетов(микротаска или макротаска), событие попадает в очередь т дальше уже выполняется в стеке.
12. What is call stack?   
    Answer:    
     Стек выполнения. Конткест в котором выполняется код. Структура данных, которая работает по принципу “первый вошел - последний вышел”

**Common**

1. What is a strict mode in javascript?   
   Answer: Режим разработки, использует более строгий вариант JavaScript
2. How do you validate an email in javascript?   
   Answer: С помощью регулярки
3. How do you get the current url with javascript?   
   Answer: window.location.href
4. What is tree shaking?   
   Answer: метод оптимизации библиотек путем удаления любого кода из окончательного файла, который фактически не используется.
5. What is memoization?   
   Answer: Мемоизация — сохранение результатов выполнения функций для предотвращения повторных вычислений
6. What are modules?   
   Answer: Простой файл содержащий код. Можно сказать, что это реализация принципа инкапсуляции
7. Why do you need modules?   
   Answer: Удобство в разработке. Чистота кода. Реализация принципов программирования.
8. How do you detect a mobile browser?   
   Answer: window.navigator.userAgent
9. How do you detect javascript disabled in the page?   
   Answer: Использовать тег <noscript> для отображения альтернативного сообщения.
10. What is a rest parameter?   
    Answer: Остаточные параметры. Можно деструктуризировать массив и поместить оставшиеся елементы в одну переменную с помощью рест параметра.
11. What is a spread operator?   
    Answer: для развертывания объектов
12. How do you encode an URL?   
    Answer: encodeURI ()
13. How do you decode an URL?   
    Answer: decodeURI()
14. What is nodejs?   
    Answer: среда выполнения JS.    
    Позволяет писать бек на JS.   
    Запускать приложения локально   
      
    80) How to abort async operation?  
    использовать  AbortController