**1. Что такое замыкание (Closures)?**

#### Замыкание – функция, которая запоминает свои внешние переменные и может получить к ним доступ. В JS все функции – замыкания. Замыкание – функция внутри функции. Замыкание – возможность функции, которая находится внутри другой функции, получать доступ к переменным которые находятся в области видимости родительской функции, даже когда родительская функция прекратила своё выполнение. Лексическое окружение – невидимый (скрытый) объект, который есть у любого блока, скрипта, функции в JS. Состоит из двух частей: 1) Переменные 2) Ссылка на внешнее лексическое окружение Лексическое окружение создаётся каждый раз при вызове функции. 2. Что такое \_\_proto\_\_\_ и что такое prototype? И то и другое свойство объекта. Два объекта не равны друг другу, только если это один и тот же объект. \_\_\_proto\_\_\_ есть у всех объектов. \_\_\_proto\_\_\_ это объект. Разные \_\_\_proto\_\_\_ по типу объектов это разные объекты, а у одинаковых по типу объектов – одинаковые proto. При создании любого объекта за кадром скрытно JS создает через new Name. Теперь понятно почему \_\_\_proto\_\_\_ у одинаковых типов объектов(например функция и функция или массив и массив) равны: они создаются через один и тот же конструктор new. Любой класс и любая функция конструктор имеет prototype. Итоги. Зачем классу нужен объект prototype? И зачем экземплярам этого класса свойство \_\_\_proto\_\_\_, которое ссылается на prototype класса? Примеры: 1) false 2) false(два прототипа всегда не равны) 3) true, 4) False( у каждой функции свой персональный прототип) 5) false 6) false 7) true 8) true 9)Function.prototype 10) Number.prototype 3. Что такое IIFE? IIFE или Immediately Invoked Function Expression — это функция, которая вызывается или выполняется сразу же после создания или объявления. Для создания IIFE необходимо обернуть функцию в круглые скобки (оператор группировки), превратив ее в выражение, и затем вызвать ее с помощью еще одних круглых скобок. Это выглядит так: (function(){})(). 4. Что такое классы (Classes)?

#### Классы — это относительно новый способ написания функций-конструкторов в JS. Это синтаксический сахар для функций-конструкторов. В основе классов лежат те же прототипы и прототипное наследование.

#### 5. Что такое функция обратного вызова (Callback Function)? Функция обратного вызова — это функция, вызов которой отложен на будущее (происходит при некоторых условиях, например, при наступлении события).