1.Что такое Promise и зачем он нужен?

Promise может находится в 3 состояниях: ожидание, успешное выполнение и ошибка. В зависимости от состояния выполняется определенное действие: ожидание – ждет выполнения операции, успешное выполнение – возвращает результат выполнения операции, ошибка – выводит причину ошибки выполнения операции. Это нужно для того, чтобы не создавать callback внутри calback’а.

Пример:

const getData = () => {  
 fetch(`https://jsonplaceholder.typicode.com/todos/1`).then((data)=>{  
 data.json().then(response => {  
 console.log(response)  
 const li = document.createElement(`li`)  
 li.className = `list-group-item`  
 li.innerText = response.title  
 const ul = document.querySelector(`.list-group`)  
 ul.append(li)  
 })  
 })  
}  
getData()

2.Конструкция async\await

Конструкция async\await позволяет писать код для выполнения асинхронных задач, который выглядит как обычный синхронный код.

Пример:

const searchCoctail = async () => {  
 const coctailDb = await fetch(`https://www.thecocktaildb.com/api/json/v1/1/search.php?s=${inputField.value}`)  
 const data = await coctailDb.json();  
 console.log(data)

}

3.Контекст JS

Контекст – это определенное окружение, в котором выполняется код в JavaScript

Контекст классов this

Обращение к кому-то элементу в зависимости от окружения, может быть глобальным, находится внутри функции. Может либо ссылаться на определенный объект, либо иметь любое значение.

4.Замыкание JS

Замыкание - это создание и вызов функции внутри другой(родительской) функции. Причем вложенная функция может перенимать атрибуты родительской функции.

Пример:

function fontSizeChanger(size) {

return function() {

document.body.style.fontSize = size + 'px';

};

}

const size12 = fontSizeChanger (12);

const size14 = fontSizeChanger (14);

const size16 = fontSizeChanger (16);

5.Зачем нужны стрелочные функции, разница в функциях

Стрелочные функции не имеют собственного контекста и являются анонимными, отличаются от обычных функций более кратким синтаксисом и привязаны к значению this.

Пример обычной функции:

function f(name) {

console.log(name)

}

f(‘Kirill’)

Пример стрелочной функции:

let a = (name) =>{

console.log(name)

}

a(‘Kirill’)