1. *Что такое объекты в JavaScript?*

Объект – это один из типов данных в JS, предназначенных для хранения набора свойств. Каждое свойство состоит из названия(ключа) и значения.

1. *Как создать объект в JavaScript?*

Объект можно создать с помощью литеральной записи. Пустой объект создают с помощью фигурных скобок {}.

const fruit = {};

внутри фигурных скобок можно сразу прописать свойства объекта

const fruit ={

name: ‘apple’,

color: ‘green’,

};

Также можно создать объект с помощью конструктора Object

const fruit = new Object({name: ‘apple’, color: ‘green’})

1. *Как создать класс в JavaScript?*

Чтобы объявить класс, используем ключевое слово class и название с большой буквы. Между фигурными скобками находится всё что касается этого класса, то есть свойства и методы

class MyClass {

method1 () {…}

method2 () {…}

}

1. *Как добавить методы в класс?*

Методы добавляем через прототип

MyClass.prototype.method = function() {…}

1. *Как создать экземпляр класса?*

Экземпляр класса создается с помощью ключевого слова new

const fruit = new Fruit (‘apple’, ‘green’);

1. *Что выведет код?*

const person = {

name: "Алиса",

age: 25,

};

console.log(person.name); //?

“Алиса”

1. *Что выведет код? Почему именно так?*

const person = {

name: 'John',

age: 25,

};

let city = person.city;

city = Moscow;

console.log(person); //?

Выведет ошибку, Moscow не определен. Перепутана логика добавления нового свойства к объекту. Сначала определяем переменную, потом указываем, что новое свойство person.city должно принять значение этой переменной

let city = 'Moscow';

person.city = city;

1. *Что выведет код? Почему именно так?*

class Animal {

constructor(name) {

this.name = name;

}

sound() {

console.log("Animal sound");

}

}

class Dog extends Animal {

sound() {

console.log("Woof!");

}

}

const dog = new Dog("Buddy");

dog.sound(); //?

"Woof!"

С помощью ключевого слова extends дочерний класс Dog наследует свойства родительского класса Animal, но при это одно свойство переписываем на новое.

1. *Что выведет код? Почему именно так?*

class Person {

constructor(name, age) {

this.name = name;

this.age = age;

}

introduce() {

console.log(`Hi, my name is ${this.name} and I'm ${this.age} years old.`);

}

}

class Student extends Person {

constructor(name, age, major) {

super(name, age);

this.major = major;

}

study() {

console.log(`I'm studying ${this.major}.`);

}

}

const person = new Person("John", 25);

const student = new Student("Alice", 20, "Computer Science");

const introduceFunc = person.introduce;

introduceFunc(); //?

Ошибка в последних двух строчках, нужно просто написать person.introduce()

1. *Что выведет код? Почему именно так?*

function sayHello() {

console.log(`Hello, ${this.name}!`);

}

let name = "Nina";

const person1 = {

name: "Alice",

greet: sayHello

};

const person2 = {

name: "Bob",

greet: sayHello

};

sayHello(); //?

person1.greet(); //?

person2.greet(); //?

Hello, CodePen! (так как проверяла в CodePen)

Hello, Alice!

Hello, Bob!