Неделя 17 Домашняя работа

**Ответы на вопросы:**

1. Что такое объекты в JavaScript?

Объекты в JavaScript – это ассоциативные массивы для хранения коллекций различных значений, которые состоят из пар «ключ (уникальное имя значения, играющее ту же роль, что и имя переменной, по которому происходит обращение к значению) — значение (данные, которые необходимо записать: строка, число, булево значение, массив, другой объект или функция)».

1. Как создать объект в JavaScript?

Объект может быть создан с помощью фигурных скобок **{…}** с необязательным списком свойств либо с помощью конструктора **new Object()**

1. Как создать класс в JavaScript?

Чтобы объявить класс, используем ключевое слово class. После него указывается имя класса с заглавной буквы — переменная, к которой можно будет обратиться при необходимости.

1. Как добавить методы в класс?

class MyClass {

// методы класса

constructor() { ... }

method1() { ... }

method2() { ... }

method3() { ... }

...

}

1. Как создать экземпляр класса?

С ключевым словом extends можно создать новый класс, который наследует от исходного все его свойства и методы:

class MyNewClass extends MyClass {

}

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; – происходит обращение к несуществующему свойству объекта

city = Moscow;

console.log(person); // **undefined**

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!"); – полиморфизм с переопределением метода

}

}

const dog = new Dog("Buddy");

dog.sound(); // **"Woof!"**

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; – обращение к методу родительского класса

introduceFunc(); // **undefined**  – вызов необъявленной функции

1. Что выведет код? Почему именно так?

function sayHello() {

console.log(`Hello, ${this.name}!`); – обращение к имени объекта

}

let name = "Nina"; – создание переменной name

const person1 = {

name: "Alice",

greet: sayHello – функция sayHello() записана в значение ключа greet объекта person1

};

const person2 = {

name: "Bob",

greet: sayHello – функция sayHello() записана в значение ключа greet объекта person2

};

sayHello(); // **"Hello, ! "**  – вызов функции

person1.greet(); // **"Hello, Alice! "** – обращение к методу greet объекта person1

person2.greet(); // **"Hello, Bob! "** – обращение к методу greet объекта person2