**Домашнее задание на эту неделю:**

Вопросы 💎

1. **Чем отличается класс от объекта? Приведите примеры и того, и другого, отличные от указанных в уроке.**

Класс это описание объекта, его состояния и поведения, а объект - конкретное воплощение или экземпляр класса.

Приведу пример из ДЗ

const comment = { // это класс

        name : commentName.value, //это объект

        body : commentBody.value, //и это объект

        avatar : commentAvatar.value, //тоже объект

        time : Math.floor(*Date*.now() / 1000) //еще объект

    }

**2. Приведите примеры объектно-ориентированного подхода в реальной жизни.**

\*Возможный вариант ответа: например, библиотека могла бы содержать классы Книга (название, автор, год выпуска, количество страниц), Сотрудники (ФИО, должность, зарплата, дата начала работы) и Читатели (ФИО, контакты, номер читательского билета).\*

Класс – домашние животные

Объект – мой котик

Свойства - порода Шотландский вислоухий, пушистый,серый,толстый

Методы - умеет ходить в лоточек, громко мяукает, когда хочет есть, может кушать/пить

-------

Класс – бытовая техника

Объект – телевизор

Свойства – большой экран, черный

Методы – работает, если включить

**Приведите примеры объектно-ориентированного подхода в программировании.**

\*Возможный вариант ответа: например, товары в интернет магазине (название товара, тип, цена), комментарии в соцсети (автор комментария, дата, текст комментария), видео в Youtube (название видео, его адрес, автор и дата загрузки).\*

Класс – UI интерфейс

Объект - Форма авторизации

Свойства – Имя пользователя, пароль

Методы – например, может сохраняться по кнопке в локальное хранилище или в куки

1. **Что такое конструктор? Самостоятельно изучите и напишите, какие бывают виды конструкторов.**

Конструктор – это функция, которая создает объект и задает ему какое-то значение.

Какие бывают виды? В джаваскрипт есть встроенные конструкторы объектов, такие как:

new String() // A new String object

new Number() // A new Number object

new Boolean() // A new Boolean object

new Object() // A new Object object

new Array() // A new Array object

new RegExp() // A new RegExp object

new Function() // A new Function object

new Date() // A new Date object

Кстати, тут нет Math. Его нельзя использовать с new. Math является глобальным объектом.

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

function bike() {

    console.log(*this*.name)

}

let name = "ninja"

let obj1 = { name: "pomidor", bike: bike }

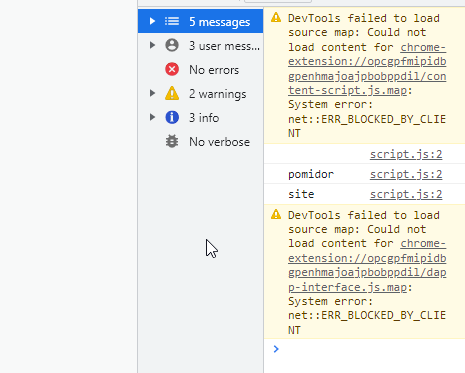
let obj2 = { name: "site", bike: bike }

bike()

obj1.bike()

obj2.bike()

Выведетpomidor и потом site



Функция bike вызывается только для двух свойств объектов, а переменная ниндзя просто создана и скучает в сторонке, как Пабло Эскобар на мемасике)

1. **Чем статические свойства и методы отличаются от нестатических? В каких ситуациях они применяются?**

Статические методы (static) применяются для реализации функций(методов), которые будут принадлежать классу целиком, но при этом не относящимся к каким-то отдельным объектам.

То есть синтаксис static необходим для одного единственного события: присвоить методу класса принадлежность самому классу и способность вызываться только из класса, и игнорировать любые вызовы из объектов.

Например:

class User {

static getMethod() { }

}

User.getMethod(); // вызов метода из класса

Это дает возможность использовать метод класса ко множеству объектов. В то время если бы мы вызывали метод из объекта например:

let user1 = new User();

user1.getMethod(); //непосредственный вызов метода из объекта

Например, без статика было бы трудно сравнить объекты, применяя классы

User.getMethod(user1, user2, user3); //

А так мы берем данные используя только один класс для этого всего, а не вызываем для каждого юзера отдельно его метод.

**6. Самостоятельно изучите, что такое геттеры и сеттеры, приведите пример класса с их использованием.**

Объекты в JavaScript обладают двумя типами свойств.

А) Свойства-данные — это обычные свойства.

Б) Свойства-аксессоры — это функции для присвоения и получения значений объекта. Вне объекта они выглядят как обычные свойства.

(accessor properties) — это 2 метода, которые получают или устанавливают значение объекта. Для этого используются два ключевых слова:

get — для геттеров (getters). Геттеры — для чтения, они позволяют получить значение объекта.

set — для сеттеров (setters). Сеттеры — для записи, они позволяют присваивать значения объекта.

Геттеры позволяют получить значение объекта. (гет – get – забрать)

const programmer = {

    // свойство-данные

    firstName: 'Natalik',

    // свойство-аксессор (геттер)

    get getName() {

        return *this*.firstName;

    }

};

// доступ к свойству-данным

console.log(student.firstName); // Вывод: Natalik

// доступ к свойству-аксессору

console.log(student.getName); // Вывод: Natalik

Сеттеры позволяют изменять значение объекта.

const programmer = {

    firstName: 'Natalik',

    // свойство-аксессор (сеттер)

    set changeName(*newName*) {

*this*.firstName = *newName*;

    }

};

console.log(programmer.firstName); // Вывод: Natalik

// изменяем свойство объекта с помощью сеттера

programmer.changeName = 'Marina';

console.log(programmer.firstName); // Вывод: Marina

7. Что выведет код? Почему именно так?

class Person {

    constructor(*name*) {

*this*.name = *name*

    }

}

const member = **new** Person("John")

console.log(typeof member)

Выведет object так как сначала создали новый объект, и мы с помощью typeof как бы спрашиваем что это за «зверь» эта переменная .

8. Что выведет код? Почему именно так?

const person = {

    name: "Valera",

    age: 23

}

console.log(person)

выведет коллекцию свойств, так как мы обратились к объекту person