**Введение в ООП**

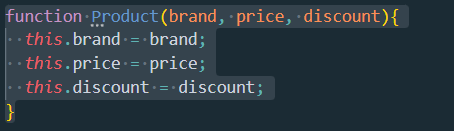
**ООП** - это не конструкция языка и не технология. Это способ организации кода в единую логически осмысленную структуру с целью оптимизации кода и обеспечения эффективного взаимодействия всех компонентов ООП-системы.

**ООП** также позволяет исключить повторяемость кода, сделать его более универсальным (например, за счет **наследования**, когда один класс наследует поведение и свойства другого).

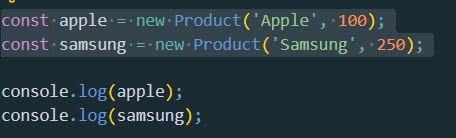
### Способы создания объекта

###### Функция-конструктор

Конструкторами являются функции, в которых используется ключевое слово «this».

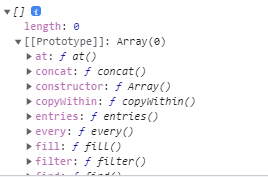


this позволяет сохранять и получать доступ к уникальным значениям создаваемого экземпляра. Экземпляры создаются с помощью ключевого слова «new».

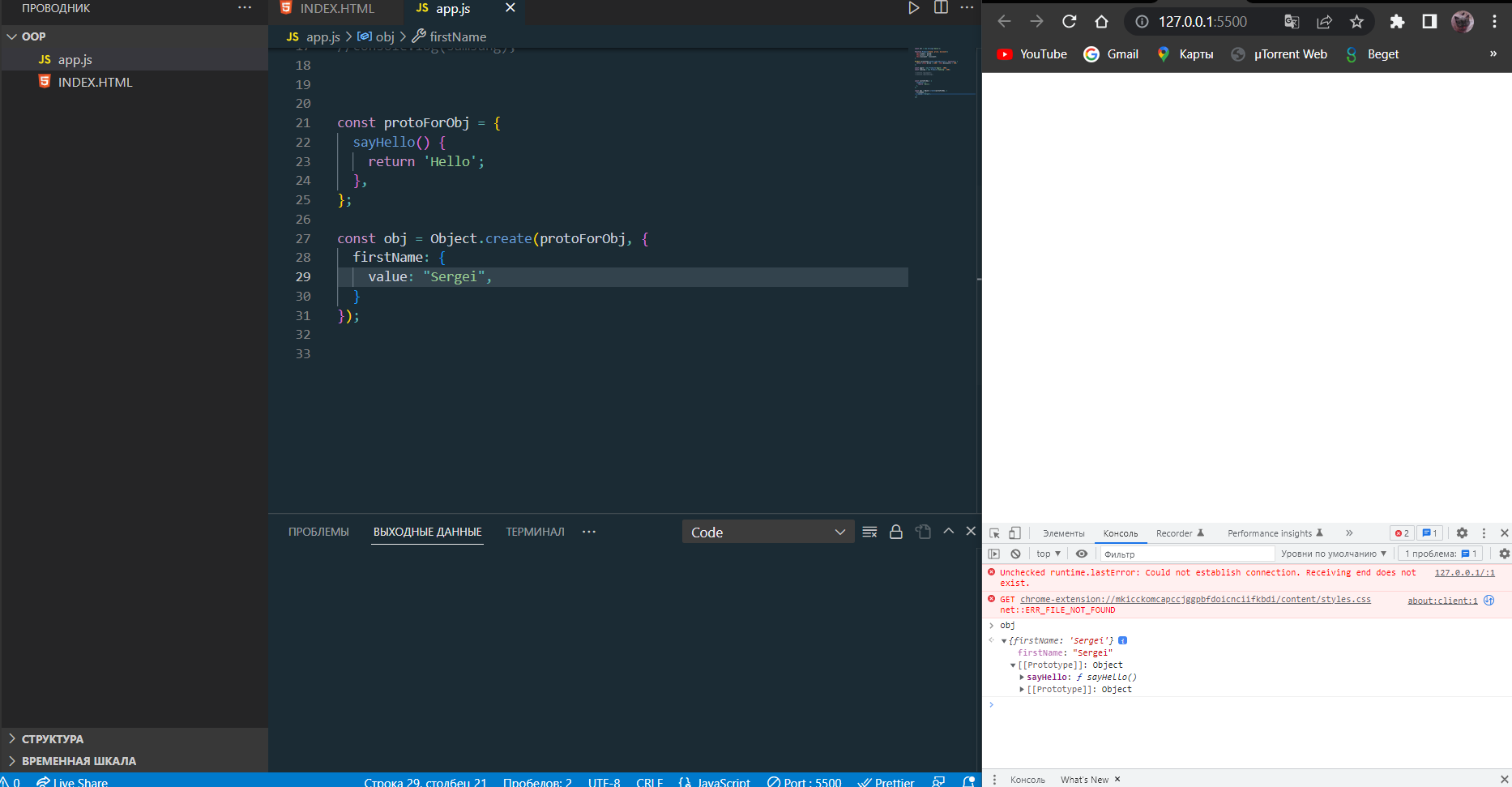


Прототип - свойство обьектов , которое содержит свойства и методов своих родителей.

Есть у любого обьекта

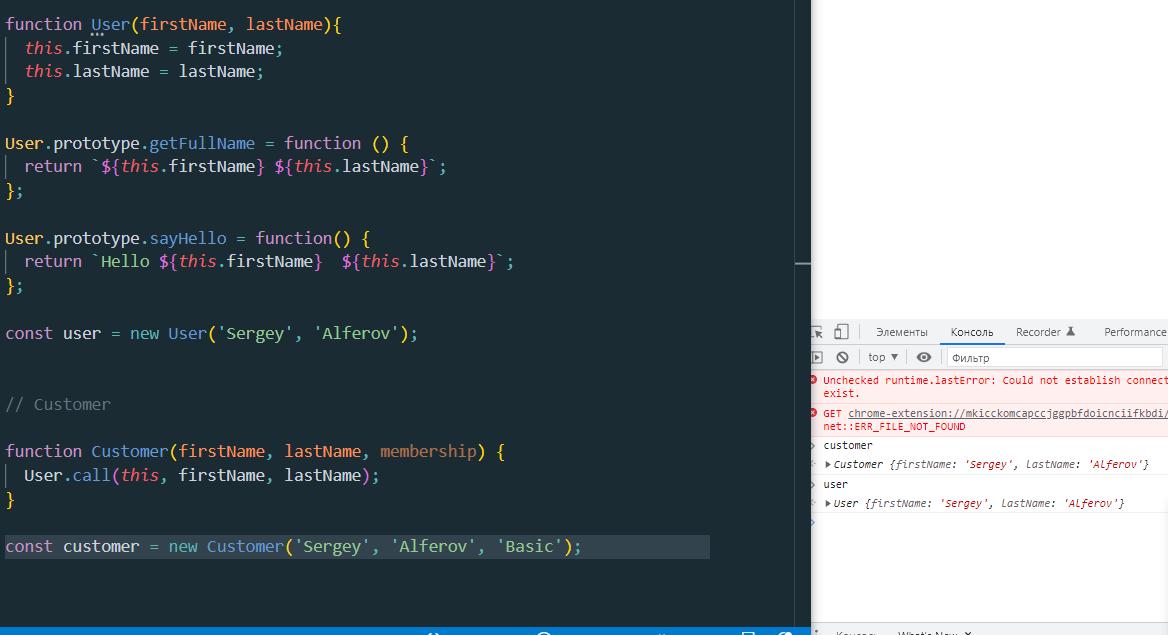


Object create - специальный метод, который позволяет создать новый объект с указанным объектом прототипа .

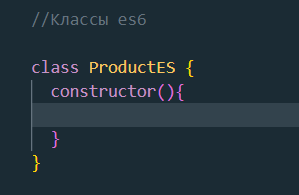


Наследование - это способ расширения одного класса другим классом.

Таким образом, мы можем добавить новый функционал к уже существующему.



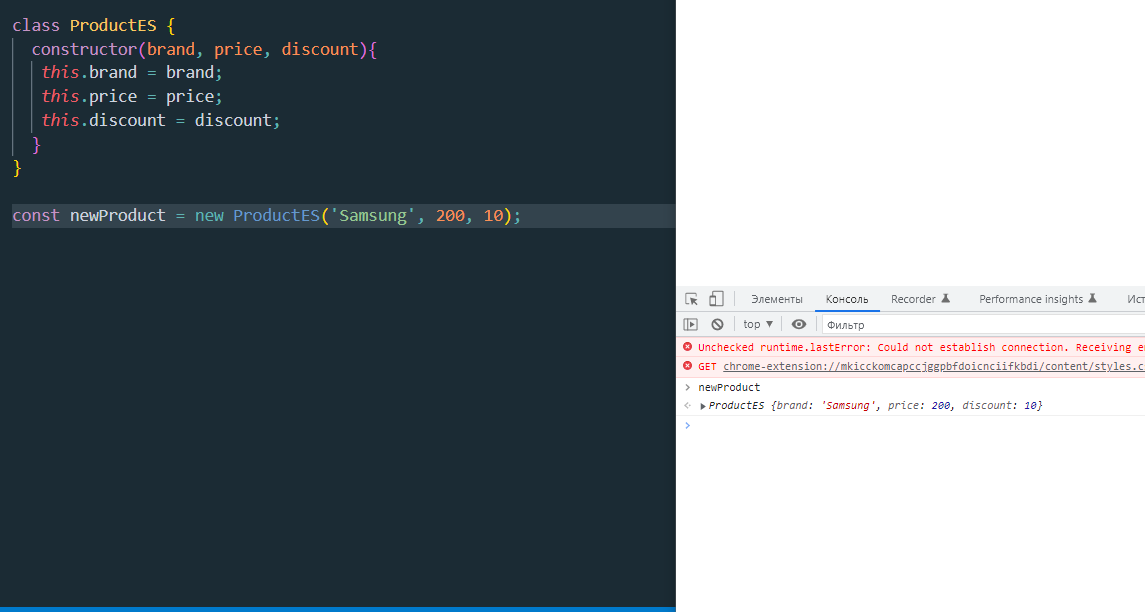
**Классы в стандарте ES6**

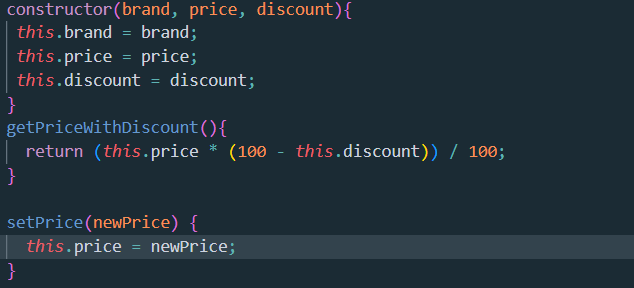


Отличие от es5 в ключевом слове class.

Constructor применяется если мы хоти добавлять методы в класс, объявление переменных

Мы не можем вызвать класс без оператора new

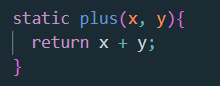




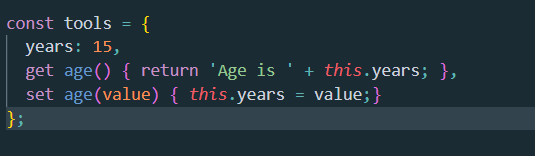
Создание методов в es6

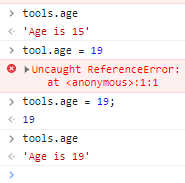
Статические методы класса - это такие методы, которые могут быть вызваны без создания экземпляра класса.

Чтобы создать статический метод в ES6 классах, нужно добавить перед методом слово static:



Сеттер/геттер - функция, которая определяет поведение свойства при его записи/чтении.







Наследование в ES6