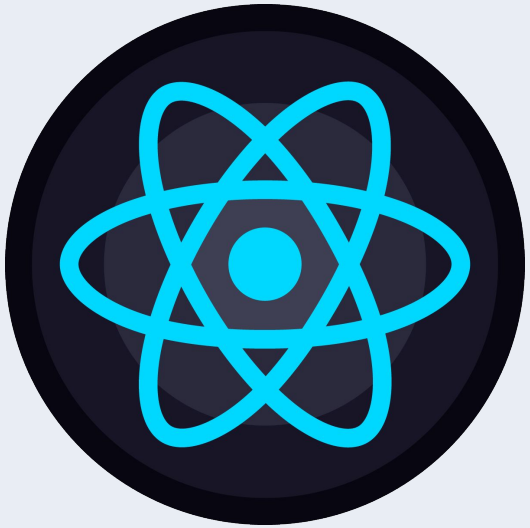


ReactJS

Урок 1

Введение в ReactJS

Современный Javascript

План урока

- Что такое ReactJS, в чем его преимущества?
- Сравнение с другими технологиями
- Современный JS. Что нового в ES6 и как это использовать
- Let, const, области видимости блока

План урока

- Деструктуризация, значения по умолчанию при деструктуризации
- Arrow functions, отличие от обычных функций
- Классы и наследование
- Промисы

Что такое ReactJS, в чем его преимущества? Сравнение с другими технологиями

React.js

React.js — это библиотека для создания интерфейсов. Была разработана Facebook, как расширение к JS. Отвечает за представление данных, получение и обработку ввода пользователя.

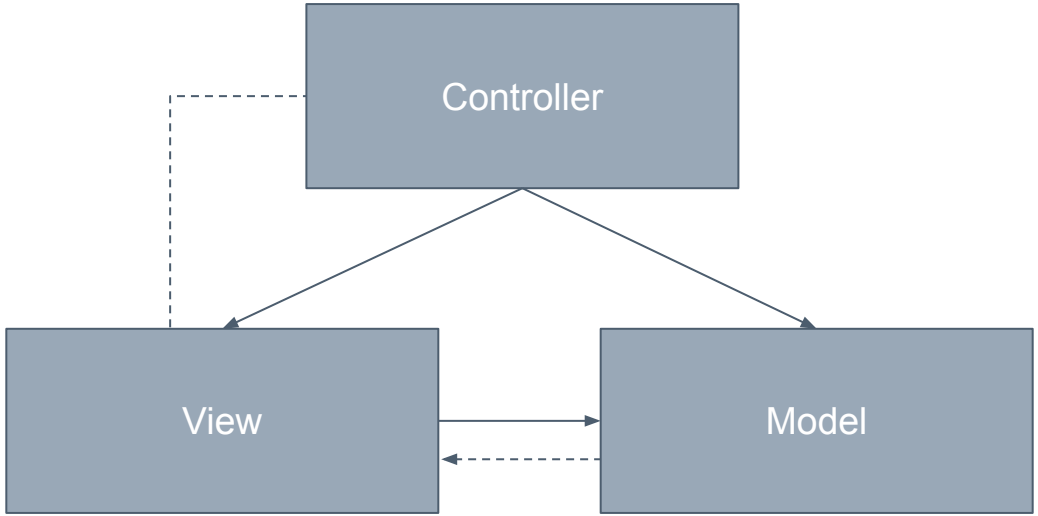
React.js

React может совместно работать с концепцией MVC (Model-View-Controller: модель-вид-контроллер). Суть концепции в четком разделении разрабатываемых приложений на три части со строго соблюдаемыми правилами и установленными задачами:

React.js

- **Model** (Обработка данных и логика приложения)
- **View** (Представление данных пользователю в любом поддерживаемом формате)
- **Controller** (Обработка запросов пользователя и вызов соответствующих ресурсов)

React.js



Плюсы:

- Простой в работе. Отлично подходит для командной разработки, строгое соблюдение UI, и шаблона рабочего процесса. Быстрый (не замедляет работу приложения)

Плюсы:

- Вы всегда можете сказать, как ваш компонент будет отрисован, глядя на исходный код (те же возможности есть в Angulsr). Если известно состояние — то известен и результат отрисовки. Не нужно прослеживать ход выполнения программы. Удобно, когда разрабатывается сложное приложение

Плюсы:

- Возможно рендерить (визуализировать) React на сервере (Angulsr и пр. шаблоны предлагают использовать доп. программы или платные сервисы)

Минусы:

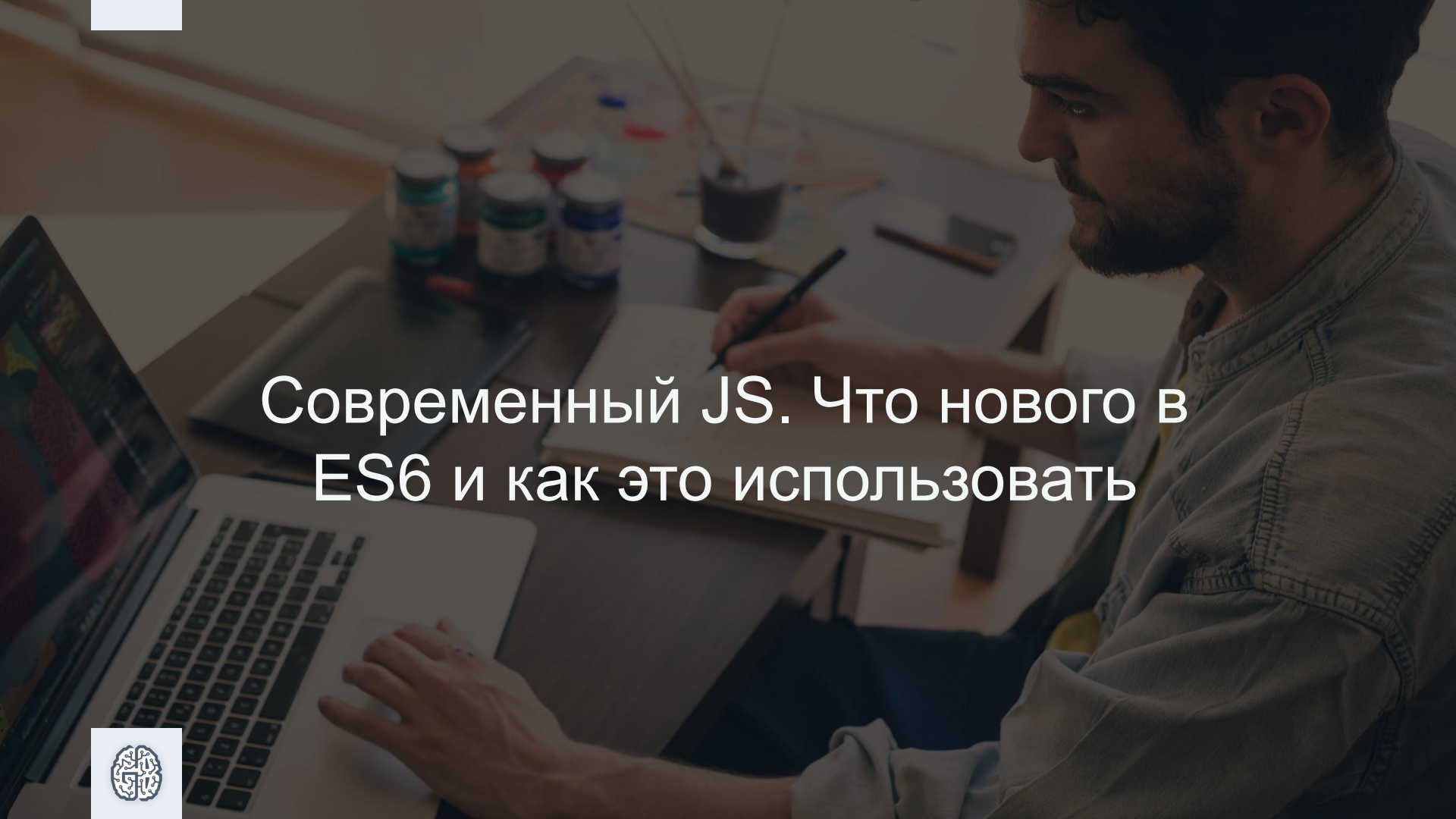
- Все-таки его работа заключается просто в переводе js-кода в html, что не позволяет создавать полноценных динамических web-приложений

Минусы:

- Не всегда понятная логика работы.Понять, как работают некоторые компоненты, и как они взаимодействуют бывает непросто.

Минусы:

- React не поддерживает браузеры от IE8 и младше. Необходимо использовать дополнительные приложения типа webpack, babel.


Современный JS. Что нового в
ES6 и как это использовать

Современный JS. Что нового в ES6 и как это использовать

Язык Java Script стандартизируется организацией ECMA (European Computer Manufacturers Association).

Стандарт носит название **ECMAScript.**

Современный JS. Что нового в ES6 и как это использовать

Стандарт определяет следующее:

- Синтаксис языка —ключевые слова, операторы, выражения и прочее;
- Типы — числа, строки, объекты и прочее;
- Правила наследования;
- Стандартные библиотеки, объекты и функций — JSON, Math, методы массивов, методы объектов.

Современный JS. Что нового в ES6 и как это использовать

Стандарт не определяет всё, что связано с HTML и CSS, DOM.

ECMAScript 6 (ES6) — новый стандарт JavaScript, принятый в 2015г.

Для конвертирования коа рекомендуется использовать: **Babel** (http://babeljs.io/repl/)

Let, const, области видимости блока

Let, const, области видимости блока

Оператор let — альтернатива var в ES6. Этот оператор очень чётко даёт возможность разобраться с областью видимости.

Let, const, области видимости блока

Константа const – эта такая переменная, которая после объявления недоступна для редактирования, при этом не имея ограничения на значение переменной.

Деструктуризация, значения по умолчанию при деструктуризации

Деструктуризация, значения по умолчанию при деструктуризации

Деструктуризация – это особый синтаксис присваивания, при котором можно присвоить массив или объект сразу нескольким переменным, разбив его на части.

Синтаксис: let [a,b,c]=[1,2,3]

Деструктуризация, значения по умолчанию при деструктуризации

Spread – выполняет разделение объекта на составляющие (присваивает каждой компоненте массива значение оператора а)

Деструктуризация, значения по умолчанию при деструктуризации

Rest – объединяет объекты массива в один.

Деструктуризация, значения по умолчанию при деструктуризации

Значение «по умолчанию» вводится в том случае, если значений в массиве меньше, чем переменных.

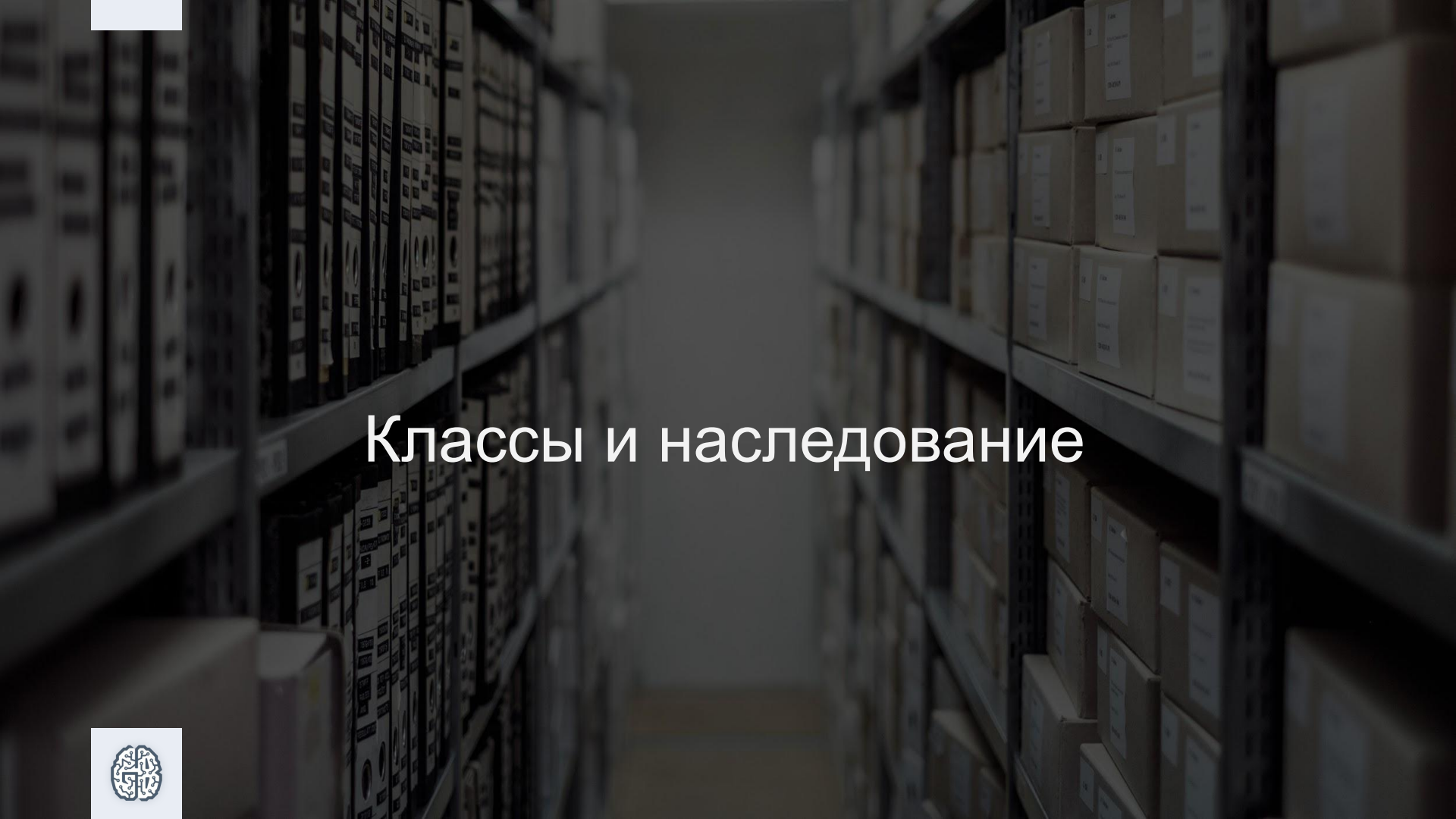
Синтаксис: let [a,b=”f”,c=”d”] = [].

Arrow functions, отличие от обычных функций

Arrow functions, отличие от обычных функций

Стрелочные функции представляют собой сокращённую запись функций в ES6.

Синтаксис:let NameFunction= (Параметр 1, Параметр 2) => a + b;

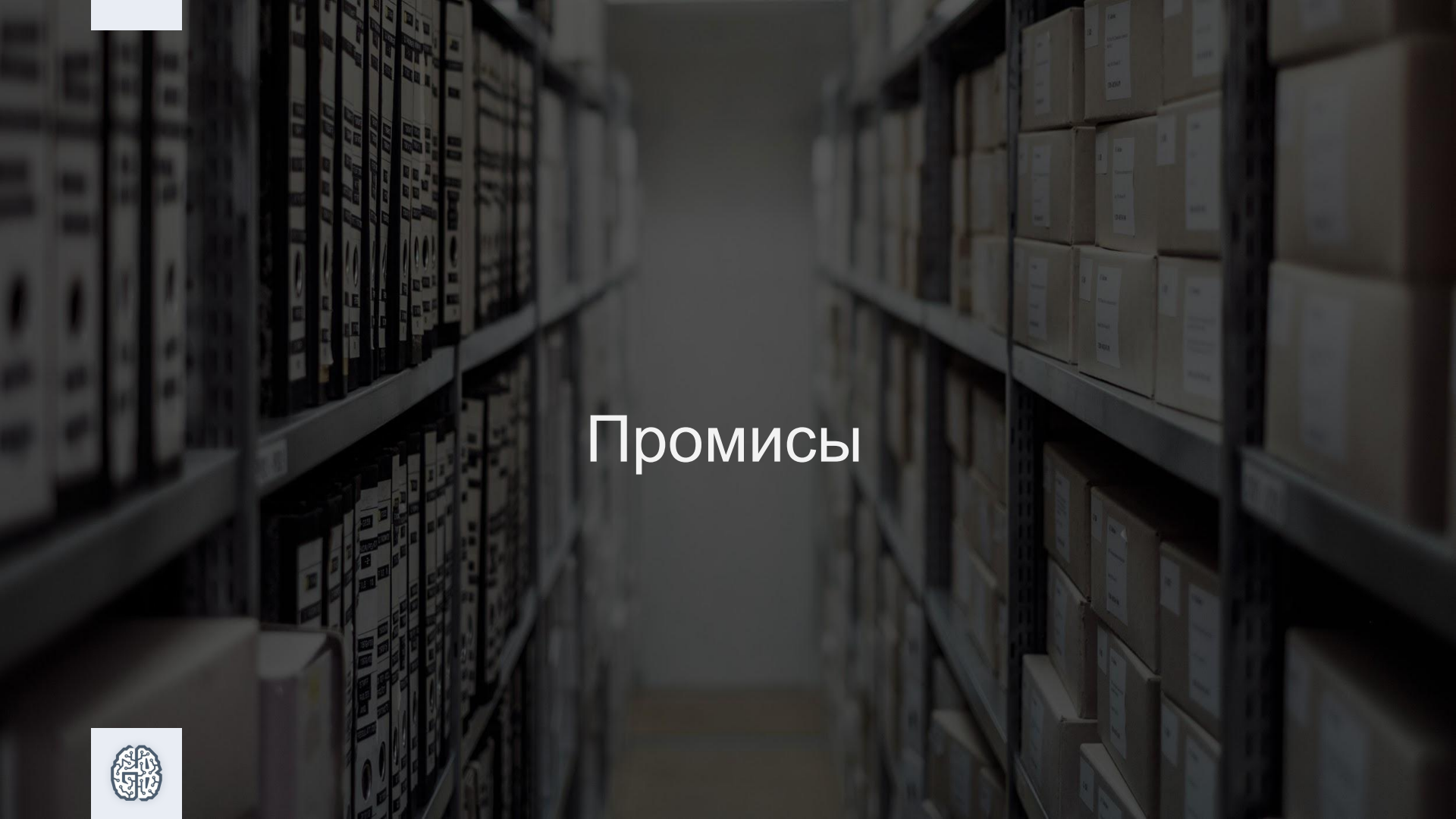

Классы и наследование

Для использования классов необходимо:

- Классы создаются в блочном коде ({и }),
- Обозначается записью classN, означает, что будет создана функция-конструктор functionN (в ES5)

Для использования классов необходимо:

- Свойство constructor используется для обозначение того, что будет происходить непосредственно в самом конструкторе.
- При перечисления методов не надо использовать запятые (вне класса они запрещены)
- Для вызова функции класса необходимо использовать оператор new.

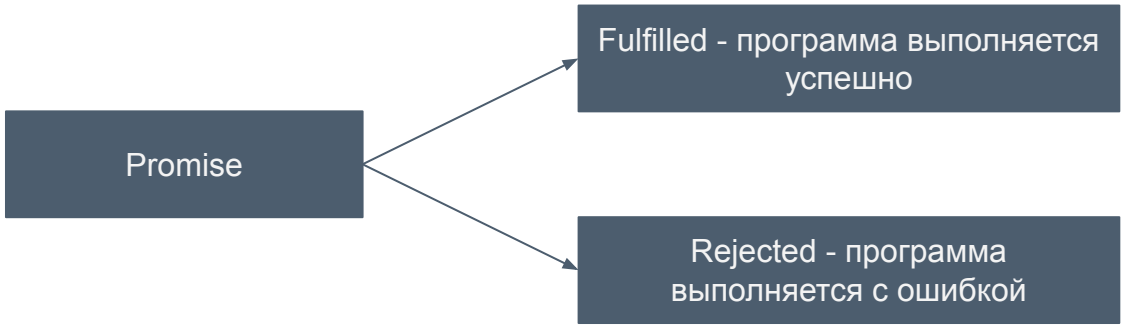

Промисы

Промисы

Промисы(Promise) - это объекты, которые ждут выполнения асинхронной операции.

После выполнения операции промис принимает одно из двух состояний: ***fulfilled*** (resolved, успешное выполнение) или ***rejected*** (выполнено с ошибкой), позволяя оценить работу кода.

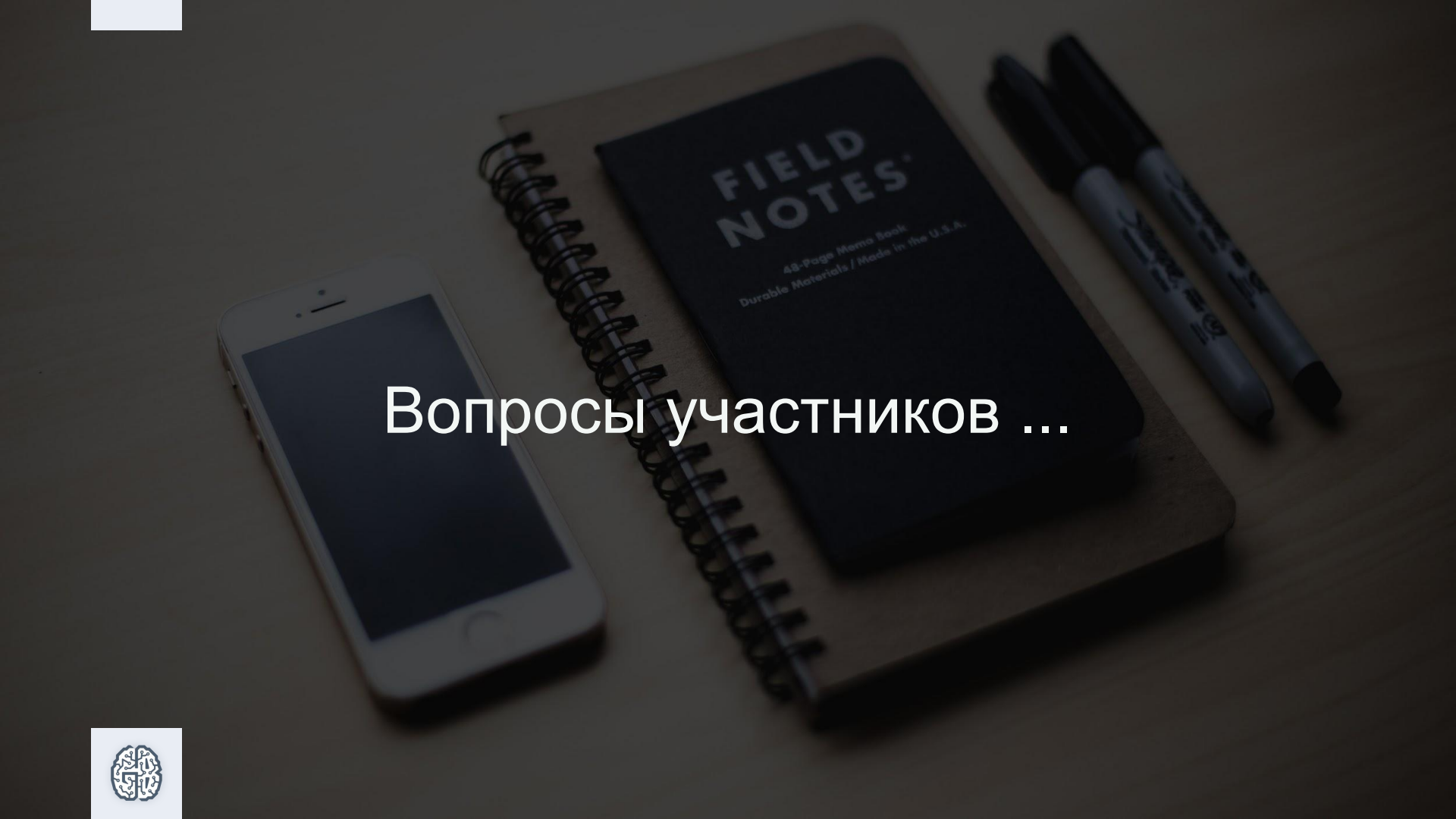
Промисы


Promise
Fulfilled - программа выполняется успешно
Rejected - программа выполняется с ошибкой

Промисы

На promise можно навешивать коллбэки двух типов:

- ***onFulfilled*** – срабатывают, когда promise в состоянии «выполнен успешно».
- ***onRejected*** – срабатывают, когда promise в состоянии «выполнен с ошибкой».



Вопросы участников ...