# JavaScript

**## Регулярные выражения**

**\* Зачем нужны регулярные выражения?**

Регулярные выражения – мощное средство поиска и замены в строке.

\* Какие методы есть у инстансов регулярных выражений в JS?

\* Какие методы есть у строк, которые позволяют работать с регулярными выражениями?

\* Что такое флаги в регулярных выражениях? Рассказать про флаги глобальности, регистронезависимости, многострочности.

\* Как работают методы указания количества (кванторы) `\*`, `+`, `{3}`?

\* Что в регулярных выражениях означают символы `[]` и `[^]`, `.`, `?`, `^`, `$`, `|`?

\* Что такое жадные и ленивые квантификаторы?

\* Как использовать базовые плейсхолдеры `\W`, `\w`,`\D`, `\d`, `\S`, `\s`, `\B`, `\b`?

\* Как работает `\b` для начала строки и для конца строки? Почему регулярное выражение `\babc\b` сработает для строки `abc`, но регулярное выражение `\b\.\.\b` не сработает для строки `..`?

\* Когда использовать экранирующий символ `\`?

\* Зачем нужны группы? Как с ними работать?

\* Можно ли создавать вложенные группы?

\* Что означают `$1`, `$2` и т.д. в текстовых редакторах в поле замены, если поиск поддерживает регулярные выражения?

\* Как сделать незапоминающуюся группу (группу, вхождение которой не будет возвращено при матчинге строки на регулярное выражение)?

\* Как использовать значение группы в самом регулярном выражении?

\* Решить задачи:

\* Написать регулярное выражение по переводу Function Expression в Function Declaration.

\* Написать регулярное выражение по переводу Function Declaration в Function Expression.

\* Написать регулярное выражение для поиска всех команд Телеграма в сообщении (примеры: `/help`, `/start`, `/active\_orders`, `/order123`). Учесть, что команд в подстроке может быть несколько.

## Promises

\* Что такое Promise? Для чего он нужен?

\* Как создать Promise?

\* В чем разница при создании промисов с помощью конструктора и с помощью фабрики?

\* Методы `then` и `catch`

\* Что делают?

\* Что такое promise chaining?

\* Что могут возвращать их коллбэки и как это повлияет на состояние промиса, возвращаемого данным `then` или `catch`?

\* В чем отличие `then(resolveHandler).catch(rejectHandler)` и `then(resolveHandler, rejectHandler)`?

\* Можно ли отменить выполнение промиса?

\* Рассказать про методы:

\* `Promise.resolve`

\* `Promise.reject`

\* `Promise.all`

\* `Promise.race`

\* В чем преимущества и недостатки коллбэков и промисов?

\* Какие значение и состояние будут у следующего промиса и почему:

```javascript

const promise = new Promise((resolve, reject) => {

resolve(0);

reject(1);

resolve(2);

});

```

#### Ресурсы

\* [JavaScript — from callbacks to async/await](https://medium.freecodecamp.org/javascript-from-callbacks-to-async-await-1cc090ddad99)

\* [Promise (learn.javascript.ru)](https://learn.javascript.ru/promise)

\* [We have a problem with promises](https://pouchdb.com/2015/05/18/we-have-a-problem-with-promises.html)

\* [You're Missing the Point of Promises](https://blog.domenic.me/youre-missing-the-point-of-promises/#toc\_1)

\* [You Don't Know JS: Async & Performance](https://github.com/leonardomso/You-Dont-Know-JS/tree/master/async%20%26%20performance) [(рус.)](https://github.com/devSchacht/You-Dont-Know-JS/tree/master/async%20%26%20performance) - можно прочесть первые три главы