



вокруг самых популярных вопросов-ответов (по опыту собственных собеседований, и собеседований, в которых я выступал собеседующей стороной), который случались за время моей профессиональной практики.

## Вопросы, относящиеся к общим знаниям в области JavaScript

**ООП в JavaScript.** Один из самых популярных вопросов, на мой взгляд. Время от времени можно услышать вариации а-ля «Что такое прототипное наследование», «Инкапсуляция в JavaScript». Информацию можно найти [тут](#) или [тут](#).

**Утечки памяти в JS.** Огромная тема, можно минут 10 рассказывать, мне кажется. Самым лаконичным примером я считаю `var a = {}; a.a = a;`. Такой код создаст утечку в IE <= 7 версии, т.к. осёл до 8 версии не умеет чистить циклические зависимости. Должно хватить для простой демонстрации понимания механизма работы с утечками. Далее будут вопросы а-ля «Как создать утечку, используя фреймворк N». В деталях можно прочитать [тут](#).

**Замыкания в JS.** Нет, ну правда, какое собеседование без вопроса о [замыканиях](#)? Вопрос отчасти относится к предыдущему («Инкапсуляция в JavaScript»), но чаще всего рассматривается отдельно.

**Самовывзывающиеся функции.** Тоже весьма частый вопрос на собеседованиях. Ещё иногда подходят к этому со стороны jQuery:

«Почему все плагины обособлены в конструкцию `(function() {} ) () ; ?`».

**Область видимости переменных. Ключевое слово «this».** Да и, честно говоря, всё, что связано с [баблингом переменных и функций](#). Мол, чем отличается запись `var a = function() {};` от записи `function a() {};`. Рассказать про [this](#).

**Как поменять контекст функции. Частичное применение функции.** Ну, тут можно оставить без комментариев, всё упирается в знание разницы между `.call`, `.apply` и `.bind` и некоторых стандартных решений на уровне языка. Обо всём этом можно прочитать в [одной замечательной статье](#).

**Разница между операторами `==` и `===`.** За ответом [сюда](#)

**Напишите регулярное выражение для проверки строк соответствующих формату даты `dd.mm.yyyy`.** Ну, тут и без комментариев всё понятно.

**Каков будет результат выполнения следующего кода:**

```
for (var i = 0; i < 10; i++) {  
  setTimeout(function () {  
    console.log(i);  
  }, 0);  
}
```

В дополнение к вопросу хочу сказать, что обычно вторым вопросом идет «Как сделать так, чтобы выводило цифры по порядку?»

**Что вернет выражение `+new Date()`? Чем отличается от `Date.now()`.** Отличный вопрос. Ответ в том, что `+new Date()`; создаст экземпляр объекта `Date` и благодаря `+` переведет его в числовой формат. Во втором случае вызовется статический метод конструктора, который является более приоритетным, т.к. во-первых он не требует создания экземпляра, а во-вторых является более понятным.

**Напишите функцию принимающую строку с именем файла и возвращающую расширение (фрагмент после последней точки).** Я просто оставлю [ссылку на ответ](#).

**Производительность циклов.** Иногда (но далеко не всегда) такой вопрос имеет место. Лично мне он нравится. По крайней мере, человек должен знать, почему `for` быстрее, чем `Array.prototype.forEach`. Но желательно ознакомиться с информацией из [этой статьи](#).

На этом, как правило, стандартные вопросы заканчиваются. Дальше идут вопросы, касательно специфики вакансии. Я имею ввиду, более «профильные» для компании. Например, если требуется JS разработчик со знанием Backbone/Marionette(Chaplin), то следующий блок вопросов будет про jQuery, underscore/lodash и прочие связанные технологии, как бы очевидно это не звучало. Вопросов по данным тематикам море, но парочку любимых я всё-таки приведу:

**Вопросы, относящиеся к jQuery**

## Почему перед исходным кодом jQuery стоит восклицательный знак?

Расскажите про **\$.Deferred**. Могу посоветовать одно: кроме знания [\\$.Deferred](#), было бы неплохо упомянуть о [нативных Promise'ax](#)

## Как создать утечку памяти с помощью jQuery.

Я люблю простые примеры, поэтому обычно привожу следующее:

```
var jqSelector = $("#selector"),
    nativeSelector = document.getElementById("selector");

// Удаляем элемент "нативным" способом
nativeSelector.parentNode.removeChild(nativeSelector);

// Выводим закешированное значение селектора jQuery
console.log(jqSelector); // Привет, jQuery.cache!
```

А вообще, в части вопросов о нативном JS уже было похожее, и там же приведена ссылка на примеры с использованием jQuery.

**jQuery.extend**. Рассказать всё про функцию, возможно попросят написать аналог. Кстати, возможно будет полезно прочитать статью про [примеси в JS](#).

**Составьте AJAX-запрос на сервер, используя jQuery**. Не особо популярный вопрос, но если спрашивать нечего, спросят что-то вроде него. Разумеется, вы должны знать, как использовать функции `$.ajax`, `$.get`, `$.post` и устанавливать настройки в

`$.ajaxSettings`. Всё это уже [давно есть на хабре](#).

Остальные вопросы скорее «для массовки», я считаю, что о jQuery больше нечего спрашивать. Конечно, будут ещё вопросы типа «Как добавить класс к объекту, используя jQuery», но я думаю, это не будет неожиданностью и не вызовет проблем.

## Вопросы, относящиеся к Underscore/Lodash

Такие встречаются, честно говоря, очень редко. Но всё же, приведу краткий список того, что я могу вспомнить:

**Чем `_throttle` отличается от `_debounce`.** Опять же, ответ есть [на хабре](#).

**Как работает функция `_extend`.** Аналог вопроса про `jQuery.extend`, ссылка на доку выше.

**Расскажите о шаблонизаторе Underscore.** Кстати, не исключено, что это только начало, и вас будут спрашивать про шаблонизатор jQuery и все остальные. Я бы советовал хоть одним глазом прочитать [эту статью](#).

**Lodash vs Underscore.** Вопрос для разведения холивара на собеседованиях

## Что дальше?

А дальше специальные вопросы, касательно фреймворков и

библиотек, которые использует компания в своих проектах.

Обычно, больше всего ценятся ответы на вопросы из категории «нативного» JS, да ведь оно и не мудрено: технологии появляются и исчезают, и совсем не удивительно, если программист о чём-то не слышал, но ответы на эту категорию вопросов показывают, как быстро человек может изучить и понять, как работает та или иная библиотека/фреймворк.

Разумеется, если вам есть, что добавить к этой статье — пишите в лс, я обязательно дополню её вашей информацией.

Спасибо  f0rk за интересные вопросы.

Проголосовать:



+104



Поделиться:



Сохранить:



Комментарии (313)

## Похожие публикации

Пишем одностраничный клиент на javascript

51



## JavaScript Performance Best Practices

32

TheShock • 6 декабря 2010 в 21:20

## 10 интересных jquery плагинов

5

Loner • 6 октября 2010 в 13:36

## Популярное за сутки

### Наташа — библиотека для извлечения структурированной информации из текстов на русском языке

14

alexkuku • вчера в 16:12

### Unit-тестирование скриншотами: преодолеваем звуковой барьер. Расшифровка доклада

4

lahmatiy • вчера в 13:05

### Люди не хотят чего-то действительно нового — они хотят привычное, но сделанное иначе

25

ПЕРЕВОД

Smileek • вчера в 10:32

### Руководство по SEO JavaScript-сайтов. Часть 2. Проблемы, эксперименты и рекомендации

2

ПЕРЕВОД



## Как адаптировать игру на Unity под iPhone X к апрелю

0

P1CACHU • вчера в 16:13

## Лучшее на Geektimes

### Стивен Хокинг, автор «Краткой истории времени», умер на 77 году жизни

33

HostingManager • вчера в 13:49

### Обзор рынка моноколес 2018

70

lozga • вчера в 06:58

### «Битва за Telegram»: 35 пользователей подали в суд на ФСБ

40

alizar • вчера в 15:14

### Стивен Хокинг и его работа — что дал ученый человечеству?

8

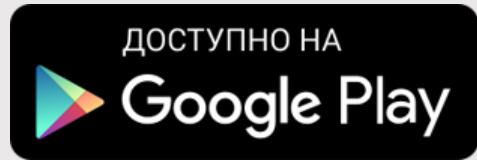
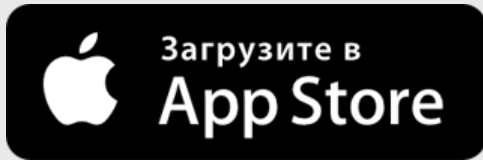
marks • вчера в 14:46

### Sunlike — светодиодный свет нового поколения

17

AlexeyNadezhin • вчера в 20:32

Мобильное приложение



Полная версия

2006 – 2018 © TM