

# JavaScript базовый (финал)

reduce еще раз и все пройденные темы

# Array reduce - метод (экземпляра) массива

Метод `reduce()` применяет функцию `reducer` к каждому элементу массива (слева-направо), возвращая одно результирующее значение.

Метод `reduce()` выполняет функцию `callback` один раз для каждого элемента, присутствующего в массиве, за исключением пустот, принимая четыре аргумента: начальное значение (или значение от предыдущего вызова `callback`), значение текущего элемента, текущий индекс и массив, по которому происходит итерация.

При первом вызове функции, параметры `accumulator` и `currentValue` могут принимать одно из двух значений. Если при вызове `reduce()` передан аргумент `initialValue`, то значение `accumulator` будет равным значению `initialValue`, а значение `currentValue` будет равным первому значению в массиве. Если аргумент `initialValue` не задан, то значение `accumulator` будет равным первому значению в массиве, а значение `currentValue` будет равным второму значению в массиве.

# Синтаксис

```
array.reduce(callback(accumulator, currentValue, index, array), initialValue);
```

callback - Функция, выполняющаяся для каждого элемента массива, принимает четыре аргумента:

- accumulator- аккумулирующий значение, которое возвращает функция callback после посещения очередного элемента, либо значение initialValue, если оно предоставлено
- currentValue- Текущий обрабатываемый элемент массива.
- index (Необязательный) - Индекс текущего обрабатываемого элемента массива.
- array (Необязательный) - Массив, для которого была вызвана функция reduce.

initialValue (Необязательный) - Объект, используемый в качестве первого аргумента при первом вызове функции callback.

# Наглядный пример

Значение, возвращенное методом `reduce()` будет равным последнему результату выполнения колбэк-функции — 10.

Колбэк-функция будет вызвана четыре раза, аргументы и возвращаемое значение при каждом вызове будут следующими:

```
[0, 1, 2, 3, 4].reduce(function (previousValue, currentValue, index, array) {  
  return previousValue + currentValue;  
});
```

	previousValue	currentValue	index	array	возвращаемое значение
первый вызов	0	1	1	[0, 1, 2, 3, 4]	1
второй вызов	1	2	2	[0, 1, 2, 3, 4]	3
третий вызов	3	3	3	[0, 1, 2, 3, 4]	6
четвёртый вызов	6	4	4	[0, 1, 2, 3, 4]	10

# initialValue в reduce (пример)

Значение, возвращённое методом `reduce()` на этот раз, конечно же, будет равным 20.

Если же вы зададите начальное значение `initialValue`, результат будет выглядеть так:

```
[0, 1, 2, 3, 4].reduce(function (accumulator, currentValue, index, array) {  
  return accumulator + currentValue;  
}, 10);
```

	accumulator	currentValue	index	array	возвращаемое значение
первый вызов	10	0	0	[0, 1, 2, 3, 4]	10
второй вызов	10	1	1	[0, 1, 2, 3, 4]	11
третий вызов	11	2	2	[0, 1, 2, 3, 4]	13
четвёртый вызов	13	3	3	[0, 1, 2, 3, 4]	16
пятый вызов	16	4	4	[0, 1, 2, 3, 4]	20

# Вывод в чем фишка Reduce метода у массива

- Универсальный, разработчик сам контролирует что возвращается
- Итерируется по всему массиву
- Функция которая исполняется при каждой итерации (callback) имеет доступ к промежуточным данным (accumulator)
- accumulator может изначально иметь какое либо значение через использование initialValue

# Базовые знания

- Числа (Number)
- Строка (String)
- Логические значения (Boolean)
- Null
- Undefined
- Числа (BigInt)
- Символы (Symbols)
- Операторы и базовый встроенный синтаксис JavaScript-a
- Функции
- Объекты
- Массивы
- Date (дата)
- Map / Set
- WeakMap / WeakSet

# Что используем на практике

Числа (Number) - (Базовое понятие для человека)

Строки (String) - (Базовое понятие для человека)

Логические значение (Boolean) - (базовое понятие в программировании)

Объекты (всегда используем - основа основ)

Массивы (самая удобная форма итерируемого и индексируемой коллекции)

Функции (метод у объектов также функция)

Дата (Date), часто используется библиотека DayJs

Null/Undefined - пустые значения - от них никуда не уйти (одна из основ JS)

# Работа с функциями

Функция как описание процесса

Если что то нужно сделать то получается это функция

Функция как конструктор

Адаптация классов в JS без использования синтаксического сахара

Функция как обертка

Более сложные конструкции основанные на свойствах функции

Методы тоже функции под капотом (если уходить в ООП)

# Работа с объектами

В JS большая часть окружения является объектом

Объект удобная форма описания чего либо

Все сложные конструкции в JS под капотом являются частными случаями объекта, например:

- Массив
- Коллекция Map / WeakMap
- Коллекция Set / WeakSet
- Дата Date
- Объекты обертки над примитивами

# Зачем это нужно:

- Работа с JS на Front-end-е (первостепенная цель)
- Работа с данными в библиотеках и фреймворках (React/VUE/Angular)
- Работа с JS на Back-end-е (дополнительно)
- Для прохождения собеседования (получить хороший оффер)

Как еще нужно использовать знания языка программирования:

- Читать плохой код
- Писать чистый код
- Мыслить как программист

Так же:

- На хорошей базе знаний JS легче строить знания более продвинутых тем
- На основе хорошие знания JS в целом - легко учить современные фреймворки

# Что нас ждет

Изучение JavaScript-а более углубленно с двух сторон:

- Работа с браузером и с HTML / CSS
- Изучения более продвинутых тем таких как ООП и Асинхронности

Изучение фреймворка React

- React упрощает работу фронта с точки зрения связи с браузером
- React дает больше возможностей для упора в реализацию бизнес логики веб - приложения
- React имеет свой синтаксис но под капотом работает на JavaScript-е
- Без знаний JS React будет магией (магия для программиста это плохо!)

**ЧТО НЕОБХОДИМО: ЗНАТЬ БАЗУ JAVASCRIPT-A**

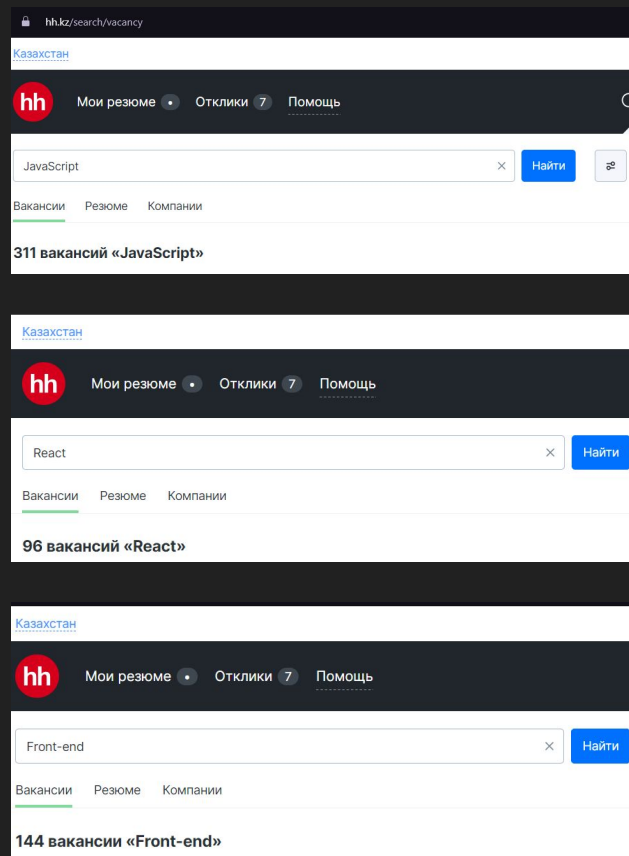


# Немного статистики

- Хорошие знания JS => чистый и понятный код
- Хорошие знания JS => вкусный оффер
- Хорошие знания JS => высокая конкурентоспособность
- Хорошие знания JS => легко учить фреймворки (адаптивность)

На уровне Junior большая конкуренция!!

(скриншоты за 01.06.2024, фильтр по Казахстану)



# Ресурсы

метод массива Reduce MDN документация - [ТЫК](#)

“Основы” онлайн учебник learn JavaScript - [ТЫК](#)

“Качество кода” онлайн учебник learn JavaScript - [ТЫК](#)

“Объекты: основы” онлайн учебник learn JavaScript - [ТЫК](#)

“Типы данных” онлайн учебник learn JavaScript - [ТЫК](#)