

# Программирование на JavaScript

---

**JS**  
**COURSE**  
**ORT DNIPRO**

---

**ORT****DNIPRO**.ORG/**JS**

**Первым делом**

**Наша группа: JS17**

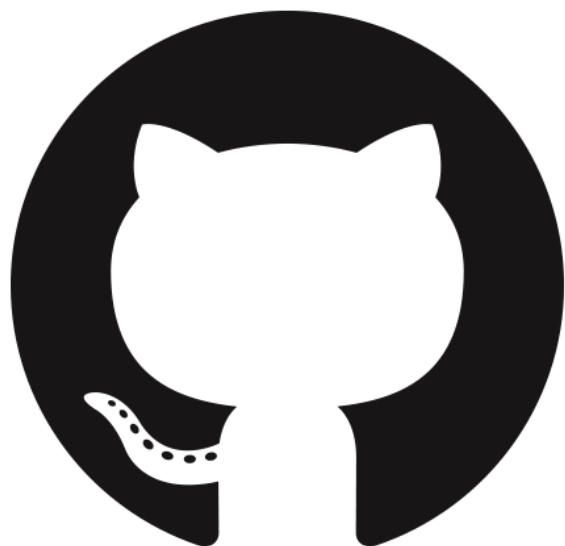
**<https://js17.online>**



Общение при помощи  
мессенджера

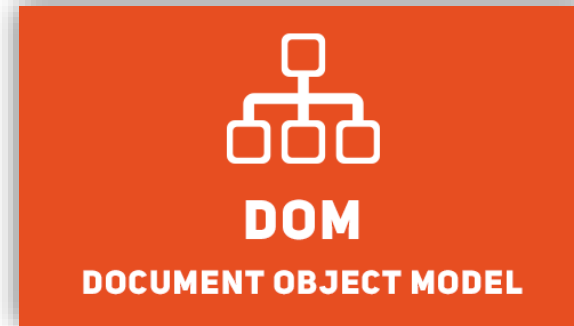
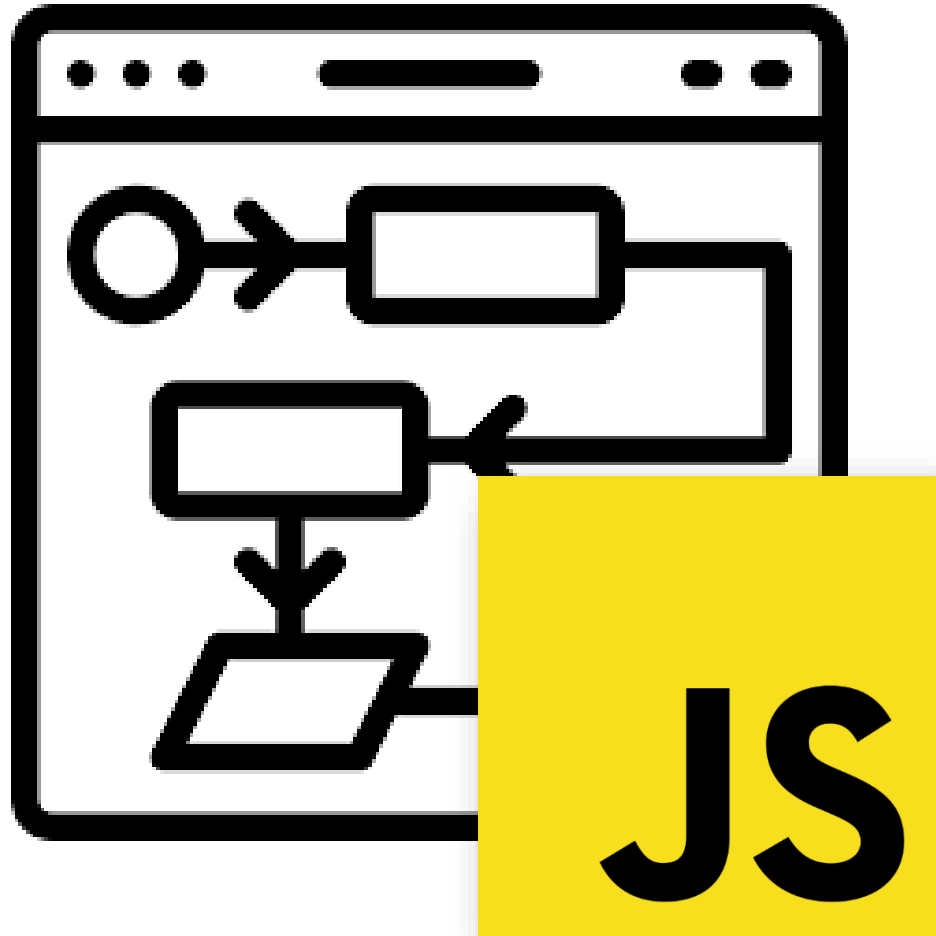
**Telegram**, а для  
обмена материалами  
и домашних заданий  
будем использовать

**GitHub**



**О чём курс?**

# О программировании и веб-разработке с применением языка JavaScript



Firestore



Поехали!

# ES

**ECMAScript**

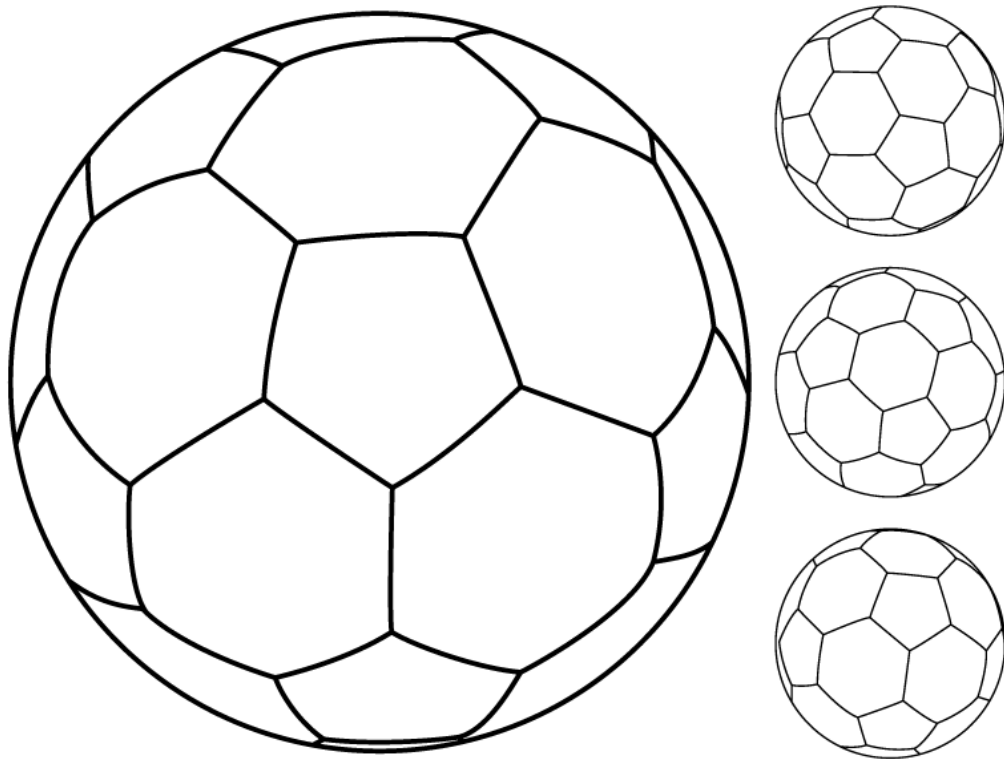
vs

# JS

**JavaScript**



# ECMAScript



*Спецификация...*

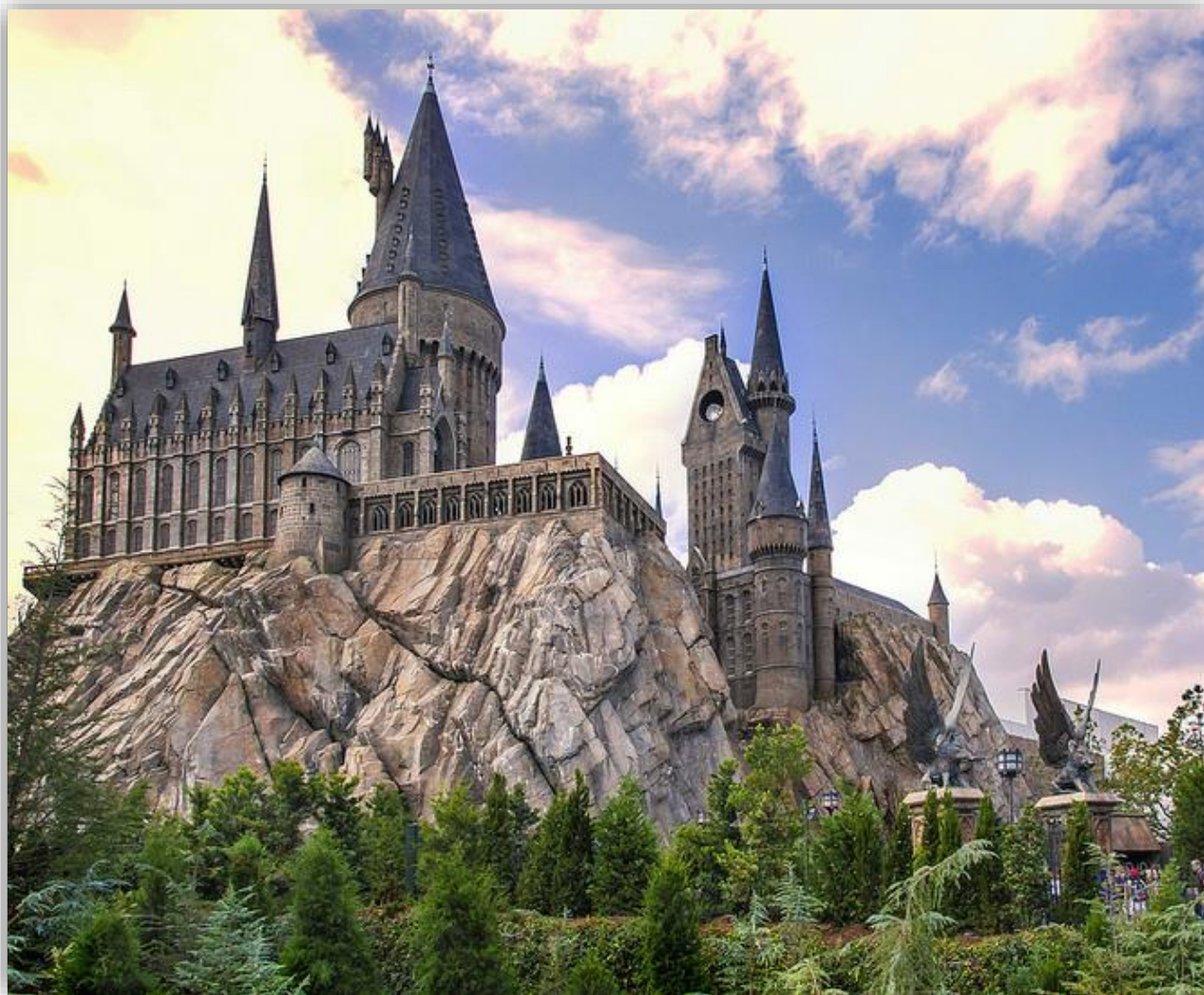
# JavaScript



*...и её реализация*



# Развитие JavaScript...



# 1. Export/Import (ES Modules)

# Процедура экспорта/импорта модулей (ES Modules)

```
JS main.js  x  JS lib.js
assets > js > JS main.js
1
2   import def from './lib.js';
3
4   import { pi as Pi, sum, config } from './lib.js';
5
6   console.log(def, Pi, sum, config);
7
8
10
11   <script src="./main.js" type='module'></script>
12
```

Директивы **export/import** по сути позволяют подключать сторонние (специальным образом подготовленные) *js-файлы* (**ES-модули**) с кодом непосредственно из *js-кода*. Для работы этого механизма первый файл (в котором импортируются другие) должен быть подключен с атрибутом **type='module'**.

```
JS main.js  JS lib.js  x
assets > js > JS lib.js > ...
1
2   const pi = 3.14;
3
4   function sum(a, b) {
5       return a + b;
6   }
7
8   let config = {
9       enable: true,
10      count: 42,
11      id: 'HX883'
12  }
13
14  export { pi, sum , config };
15
16  export default config;
17
```

## 2. Переменные и типы данных



# Переменные в JavaScript

```
1
2  var user_name    = "Elena";
3
4  let user_age     = 27;
5
6  const user_inn   = 3252873450;
7
8  console.log(user_name, typeof user_name);
9  console.log(user_age,  typeof user_age);
10 console.log(user_inn,  typeof user_inn);
11
```

Переменные объявляются при помощи ключевых слов **var**, **let** и **const**. Первые два способа отличаются областью видимости переменной которая создаётся. Третий создаёт переменную у которой нельзя заменить значения после инициализации.

# Типы данных в JavaScript

```
3
4  undefined    //undefined
5
6  number       //42, -35.783, 4e18, NaN, Infinity ...
7
8  string       //'Hello', "World", `!!!` ...
9
10 boolean      //true, false
11
12 object       //null, { prop:'value', ... } ...
13
14 symbol       //Symbol('marker'), Symbol.for('label')
15
16 bigint       //35n, 9999999999999999999999999999999n
17
18 function     //function(...){...}, (...)=> ...
19
```

Переменные могут хранить значение одного из поддерживаемых типов данных. В ходе выполнения кода может меняться как содержимое переменной так и его тип.

**Тип влияет на то какие операции могут быть выполнены с переменной.** Тип переменной можно получить при помощи оператора/функции **typeof**.

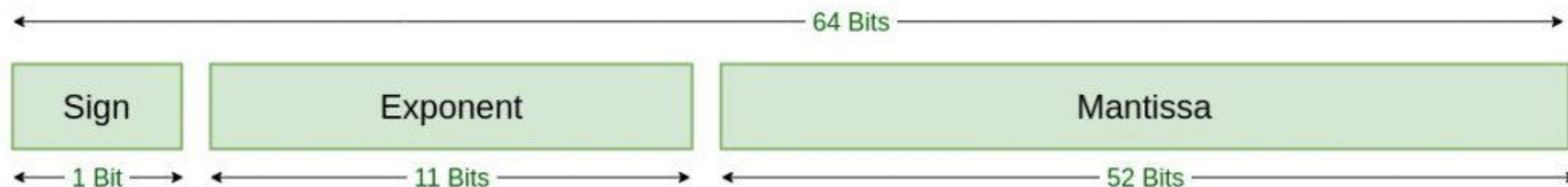
# Преобразование типов

Несмотря на наличие механизма автоматического приведения типов может возникать ситуации требующие принудительного преобразования типов (чаще всего **string** к **number**), для этого есть ряд возможностей. В первую очередь при помощи соответствующих функций-конструкторов **Number()**, **String()**, **Boolean()**, **BigInt()**, **Symbol()** и т.д.

Подробнее: <https://learn.javascript.ru/types-conversion>



# О проблемах с «точностью» при работе с **number**



Double Precision  
IEEE 754 Floating-Point Standard

<https://www.youtube.com/watch?v=U0U8Ddx4TgE>

# **3. Немного практики**

# Задача #1

**Задача:** Тарифы банка за перевод средств с карты на карту: **1%** за счёт личных средств и **4%** в счёт кредитного лимита. Скрипт должен рассчитывать сумму комиссии за перевод (который хочет выполнить пользователь), и определять возможно ли выполнить перевод (достаточно ли средств).

# Задача #2

**«Задача банкомата»** Программа спрашивает у пользователя сумму, а в ответ сообщает купюры каких номиналов, и в каком количестве необходимо выдать. При этом *суммарное количество купюр должно быть минимально возможным*. Для простоты будем считать, что в банкомате есть только купюры по **1, 5, 20, 50** гривен (при этом из количество не ограничено).

*Например: Пользователь вводит сумму: 552 грн.*

*В ответ программа выдаёт:*

50 грн. x 11;

20 грн. 0;

5 грн. x 0;

1 грн. x 2;

**«Источник знаний»**

O'REILLY®

7-е  
издание

# JavaScript Полное руководство

Справочник по самому популярному  
языку программирования



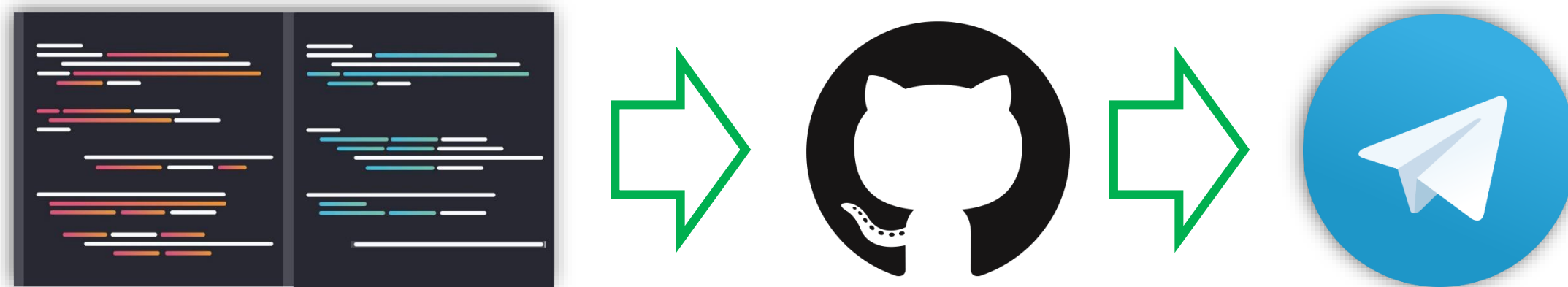
Дэвид Флэнаган

Дэвид Флэнаган

**JavaScript:** Полное  
руководство, 7-е издание

**Домашнее задание**  
**/сделать**

Каждое домашнее задание оформляйте в виде отдельного репозитория на GitHub, в названии которого **укажите номер задания** (например: **a1-federal-tax**)



Если есть проблемы, вопросы, трудности, делаем тоже самое – код с проблемой заливаем на **GitHub** и ссылку на него, с описанием проблемі в **группу**.



# Домашнее задание #А.1



Есть в США такой вид налога как **Federal Income Tax**, ваша задача написать налоговый калькулятор, который будет рассчитывать сумму налогов в зависимости от годового дохода человека. За основу взять ставки налога для доходов полученных за **2022 г.** , и для простоты - расчёт выполнять **только** для лиц не состоящих в браке (**For Single Filers**) и не упустите **налоговые вычеты (Deduction)**.

<https://taxfoundation.org/2022-tax-brackets/>

В репозитории занятия **есть тестер**: `./src/homework-tester` для сверки. Расхождением в **~1 доллар** можно пренебречь.

О прогрессивном налогообложении в целом, с примерами:

<http://allfi.biz/glossary/eng/P/progressive-taxation.php>

# Домашнее задание #А.1



**Пример** расчёта для суммы в **\$100 000** (для **Single Filers**)

1) Налоговый вычет, согласно таблице вычетов (**deduction**)  $\$100\,000 - \$12\,950 = \$87\,050$  (эта сумма и будет облагаться налогом);

2) В какие бакеты попадает налогооблагаемая сумма (согласно таблице):

В первый полностью (от **\$0** до **\$10 275**) и от этой части берётся **10%**;

Во второй полностью (более **\$10 275** и до **\$41 775**) и от этой части берётся **12%**;

В третий бакет попадает частично (более **\$41 775** и до **\$87 050**, нашей облагаемой суммы) и от этой части берётся **22%**;

3) Считаём

В первом бакете  $(\$10\,275 - \$0) * 10\% = \$1\,027.5$ ;

Во втором бакете  $(\$41\,775 - \$10\,275) * 12\% = \$3\,780$ ;

В третьем бакете  $(\$87\,050 - \$41\,775) * 22\% = \$9\,960.5$ ;

Складываем сумму по всем бакетам:  $\$1\,027.5 + \$3\,780 + \$9\,960.5 = \$14\,768$ .

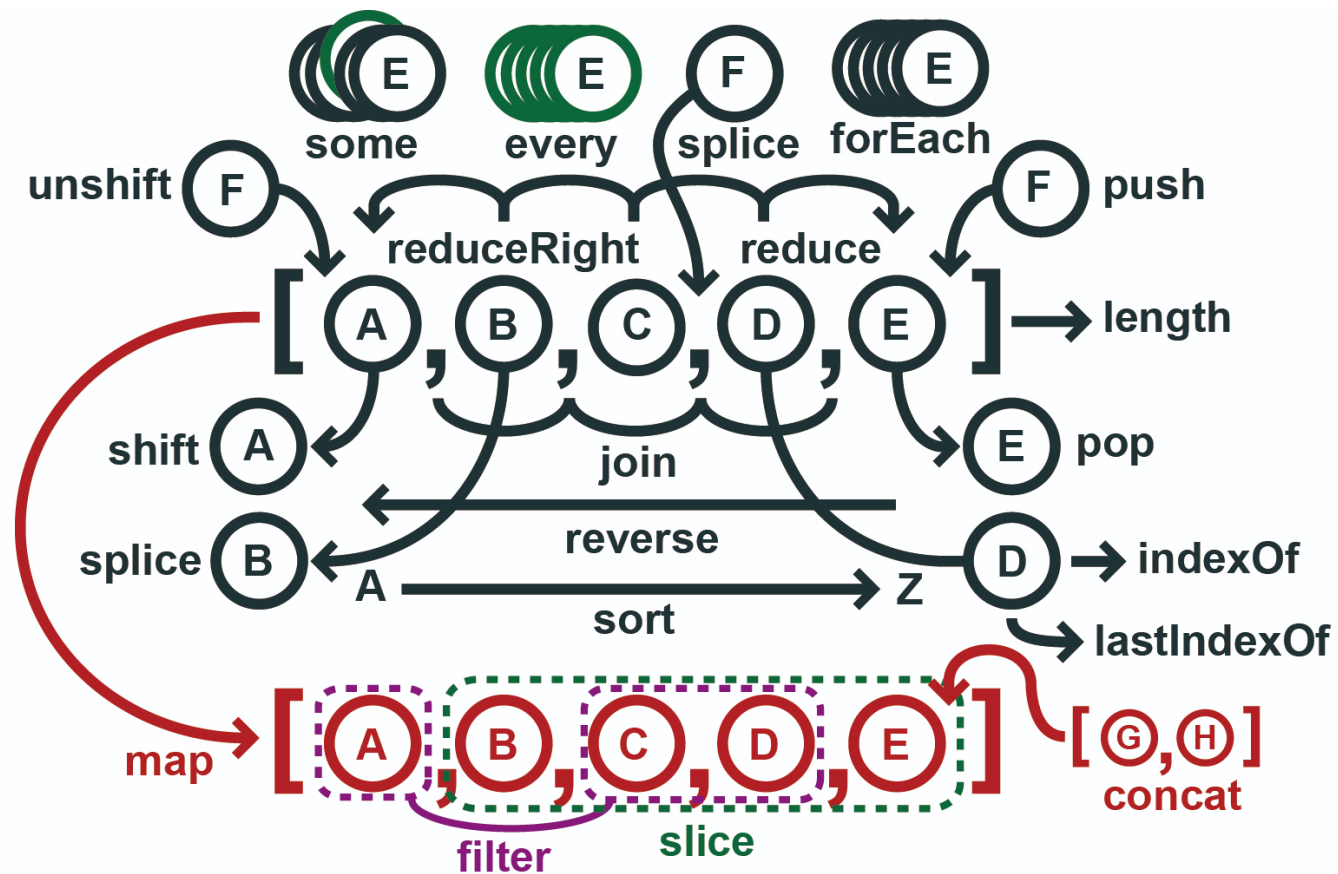
Итого нужно заплатить **\$14 768** налогов.

П.С. цикл и массив тут могут очень помочь (но можно и без них).

<https://taxfoundation.org/2022-tax-brackets/>

**На следующем занятии...**

# На следующем занятии



Коллекции в JavaScript и работа с ними