«Массивы и циклы»

JavaScript
Courses

courses.dp.ua

О циклах

Если какие-либо действия нужно повторять, но заранее неизвестно сколько раз

```
<script>
    var password = prompt("Введите пароль: ");

    if((password == "12345")){
        alert("Access approved.");
    }else{
        alert("Wrong password!");
    }
</script>
```

Если пароль не подходит, то нужно повторно запросить его у пользователя, и так повторять до тех пор пока не будет введён правильный пароль.

*Т.е. нам нужен механизм который будет **повторять набор действий до тех пор пока будет верно условие** (например: пароль не равен «12345»)*

Циклы – способ многократно повторить фрагмент кода

*Цикл **while / do..while** выполняет фрагмент кода пока условие заданное в нём верно (истинно, **true**).*

```
1  <script>
2
3      do{
4          var password = prompt("Введите пароль: ");
5
6          if((password == "12345")){
7              alert("Access approved.");
8          }else{
9              alert("Wrong password!");
10         }
11
12     }while(password != "12345");
13
14 </script>
```

*В данном примере помимо проверки данных оператором **if** еще следует проверка данных оператором **while**.*

Циклы – способ многократно повторить фрагмент кода

*Цикл **while / do..while** выполняет фрагмент кода пока условие заданное в нём верно (истинно, true).*

```
1  <script>
2
3      do{
4          var password = prompt("Введите пароль: ");
5
6          if((password == "12345")){
7              alert("Access approved.");
8          }else{
9              alert("Wrong password!");
10         }
11
12     }while(password != "12345");
13
14 </script>
```

!!!В теле цикла должны происходить какие-либо изменения тех переменных которые находяться в условии, иначе цикл будет выполняться вечно!!!

Циклы в JavaScript

Циклы **while | do/while**

```
while(a > b){
    //... do something
}
```

```
do{
    //... do something
}while(a > b);
```

Циклы **while** и **do/while** – циклы с условием, они выполняются пока условие будет истинным (**true**). Отличие в том, что **while** проверяет условие на входе в каждый шаг цикл (и может возникнуть ситуация когда тело цикла ни разу не выполниться), а **do/while** на выходе из каждого шага цикла (тело цикла выполнится минимум 1 раз).

https://learn.javascript.ru/while-for

Цикл **for**

```
var n = 10;

for(var i = 0; i < n; i++){
    //... do something
}
```

Цикл **for** (или **цикл со счётчиком**) – хорош тем, что содержит удобную возможность подсчёта шагов (**итераций**) цикла. Этот цикл удобен для ситуация когда мы заранее знаем сколько шагов цикла нам понадобиться сделать.

https://learn.javascript.ru/while-for

Циклы **for..of** (ES-2015)

```
var mas = [7, 12, 5, 23, 56, 8];

for(var item of mas){
    //... do something
}
```

Цикл **for..of** (нововведение ECMAScript-2015) предназначен для перебора содержимого коллекций (не индексов, а именно содержимого).

https://learn.javascript.ru/iterator

Цикл **for..in**

```
var user = {
    name: "Ivan",
    phone: "+38 (067) 123-45-67",
    city: "Dnipro"
}

for(var key in user){
    //... do something
}
```

Цикл **for..in** предназначен для перебора ключей коллекции (на практике полезен в первую очередь для ассоциативных массивов).

https://learn.javascript.ru/object-for-in

Операторы
break / continue

Операторы break и continue

```
var i = 0;

while(i < 100){

    if(i % 3 == 0) continue;

    if(i % 7 === 0) break;

    i++;
}
```

Оператор **break** позволяет прервать цикл, оператор **continue** позволяет завершить текущий шаг (итерацию) цикла и перейти к следующей.

https://learn.javascript.ru/while-for#preryvanie-tsikla-break
https://learn.javascript.ru/while-for#continue

Немного практики #1

Кредитный калькулятор v.1

Заданы: Есть сумма кредита, годовая процентная ставка, и срок кредитования в месяцах. **Рассчитать** ежемесячные платежи (сколько в каждом месяце будет платить заёмщик, указав сколько из суммы ежемесячного платежа идёт на погашение тела кредита, а сколько на погашение процентов, а также, сколько остаётся долга по телу кредита) по **классической** схеме.

Коллекции в JavaScript

Коллекциями

Коллекциями в языках программирования называют структуры данных предназначенных для **хранения множества значений**. Коллекции можно разделить на те которые хранят пары **ключ=>значение** (*массив, ассоциативный массив, словарь – Map*) и просто хранящие значения (*множество – Set*).

Массивы с числовыми индексами

```
var mas = ["Ivan", 42, "Dnipro", true];

mas[2] = "Test";

mas.push("Elena");
mas.unshift(12.75);

mas.length;
```

Массив с числовыми индексами (ключами) – *это нумерованный набор переменных. Оператор* ***[ ]*** *– основной признак массива, он позволяет обратиться по номеру к конкретному элементу массива.*

https://learn.javascript.ru/array
https://learn.javascript.ru/array-methods

Ассоциативные массивы

```
var user = {
    name: "Ivan",
    phone: "+38 (067) 123-45-67",
    city: "Dnipro"
}

user["name"] = "Elena";
console.log(user["name"]);
```

В ассоциативном массиве **индексом (ключом)** к элементу выступает не число, а **строка**. В ассоциативном массиве о длине массива как правило не говорят.

П.С. В JavaScript ассоциативные массивы и объекты это одно и тоже, но лишь потому, что объекты в JavaScript построены на базе концепции ассоциативных массивов;

https://learn.javascript.ru/object

Со строками можно работать как с массивом

```
var str = "Hello world!!!";

console.log(str[2], str[3], str[4]);
```

Строки по сути представляют собой коллекцию символов, к которым можно обратится по числовому индексу **[n]**, также строки имеют свойство **.length** показывающее их длину. Также возможен получение символа через метод **.charAt()** по номеру. Установка символа таким способом невозможна.

https://learn.javascript.ru/string

Немного практики #2

Алгоритм Луна

VISA *4916 5526 5398 1949*

mastercard *5357 6872 3409 1447*

Алгоритм Луна проверяет контрольную сумму числа, широко применяется для проверки корректности номера банковских карт.

Задача: пользователь вводит номер банковской карты, необходимо проверить не ошибся ли он.

https://uk.wikipedia.org/wiki/Алгоритм_Луна

Spread оператор

...

Spread оператор … (ES-2015)

```
var mas_1 = ["a", "b", "c"];
var mas_2 = [1, 2, 3];

var mas_3 = [...mas_1, ...mas_2];

console.log(mas_3);
```

```
var mas = [11, 33, 55, 77, 99];

var [ ,a, b, ...other] = mas;

console.log(a, b, other);
```

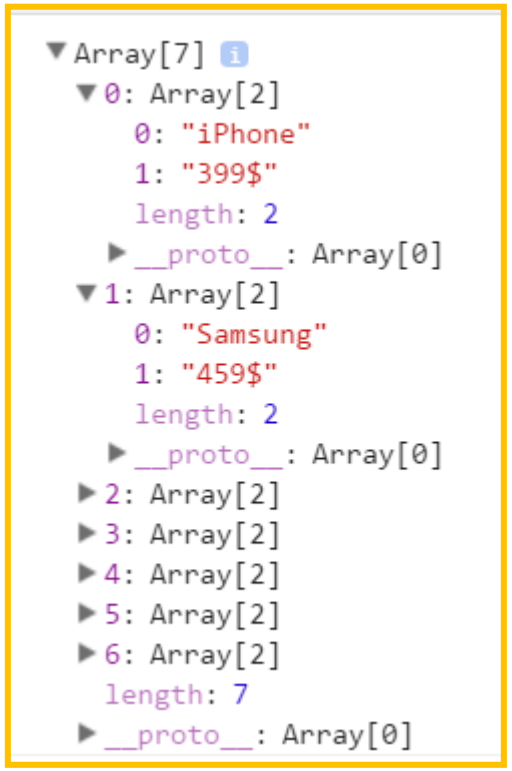
Spead *оператор (…) позволяет разложить коллекцию на элементы в неё входящие.*

Многомерные массивы

Многомерные массивы

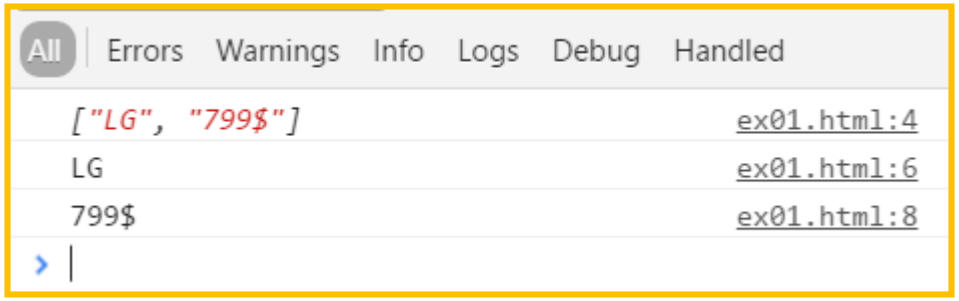
```
1  <script>
2      var arr = [["iPhone", "399$"], ["Samsung", "459$"], ["HTC",
       "150$"], ["LG", "799$"], ["Lenovo", "199$"], ["Sony", "430$"],
        ["Nokia", "39$"]];
3
4      console.dir(arr);
5  </script>
```

Многомерные массивы – массивы элементы которого сами являются массивами.



Многомерные массивы

```
1  <script>
2      var arr = [["iPhone", "399$"], ["Samsung", "459$"], ["HTC",
       "150$"], ["LG", "799$"], ["Lenovo", "199$"], ["Sony", "430$"],
        ["Nokia", "39$"]];
3
4      console.log(arr[3]);
5
6      console.log(arr[3][0]);
7
8      console.log(arr[3][1]);
9  </script>
```

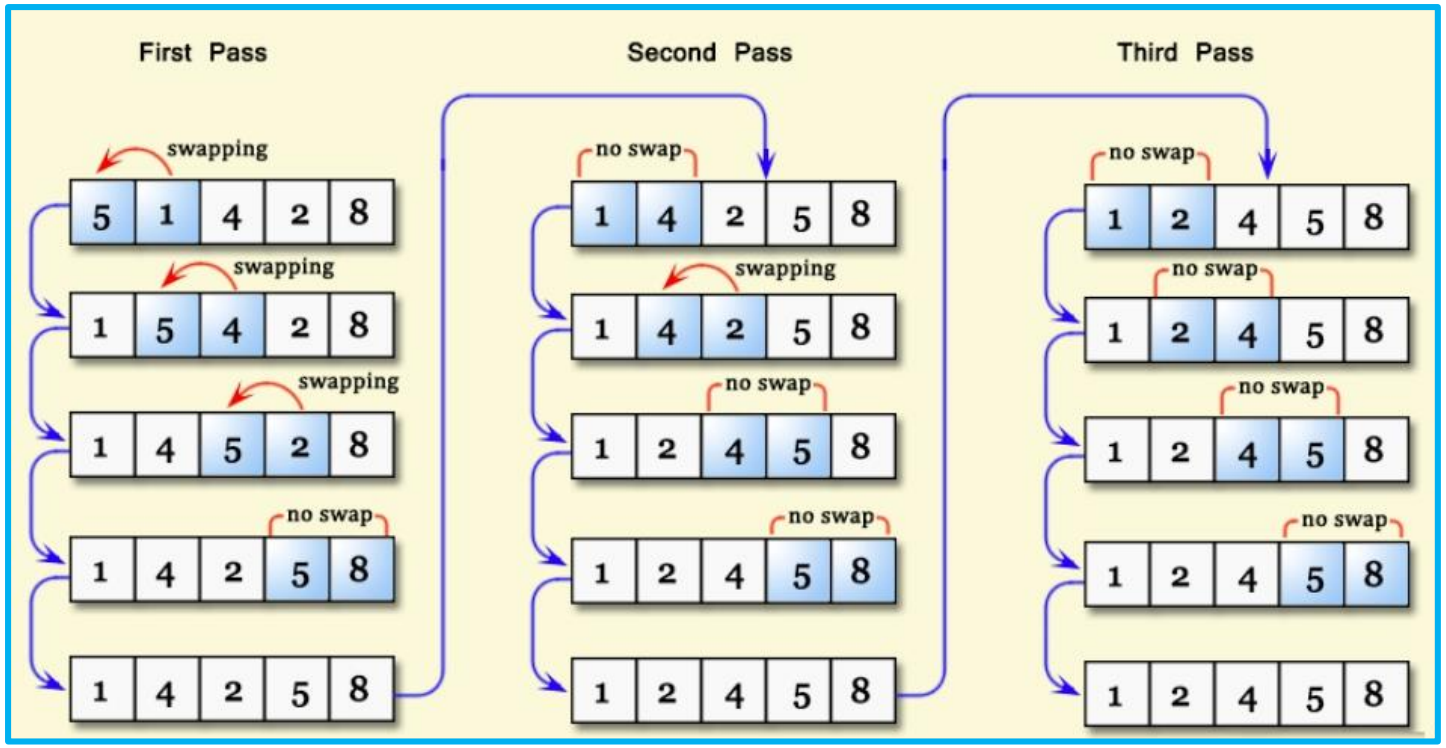


Многомерные массивы – массивы элементы которого сами являются массивами. Обращение к элементам осуществляется использованием нескольких пар скобок ***[][]***

Немного практики #3

Сортировка данных

Когда необходимо внести изменения в существующий набор данных.



Классический алгоритм «пузырьковой» сортировки.

https://habrahabr.ru/post/204600/

Немного практики #4

Игра «Угадай число»

Необходимо написать скрипт который загадает число, в диапазон от 1 до 1000 включительно. И даст пользователю 10 попыток на угадывание. Если пользователь во время попытки не угадал число, ему даётся подсказка в виде «число которое я загадал больше чем ваш вариант» или «...меньше...». Функция **Math.random()** вам поможет.

JSON | JavaScript Object Notation

JSON (JavaScript Object Notation)

JSON *- текстовый формат обмена данными,* удобный для чтения и написания как человеком, так и компьютером. *Основан на синтаксисе (правилах записи) массивов в* ***JavaScript****. Формат поддерживается практически во всех современных языках программирования.*

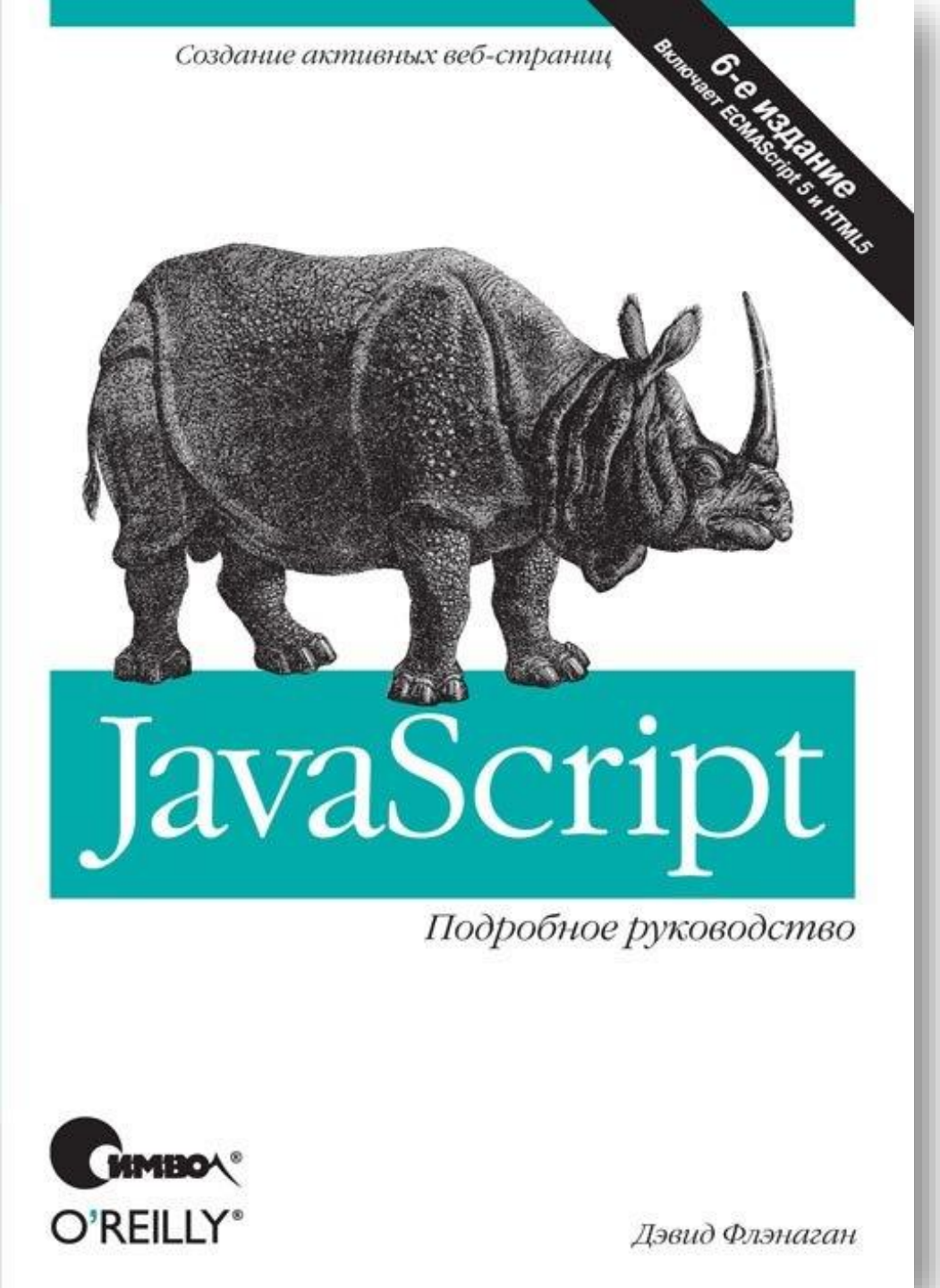

```
[{"ccy":"USD","base_ccy":"UAH","buy":"28.05000","sale":"28.25000"},
{"ccy":"EUR","base_ccy":"UAH","buy":"31.95000","sale":"32.45000"},
{"ccy":"RUR","base_ccy":"UAH","buy":"0.41500","sale":"0.43500"},
{"ccy":"BTC","base_ccy":"USD","buy":"6143.7724","sale":"6790.4852"}]
```


ПриватБанк

https://api.privatbank.ua/p24api/pubinfo?json&exchange&coursid=5

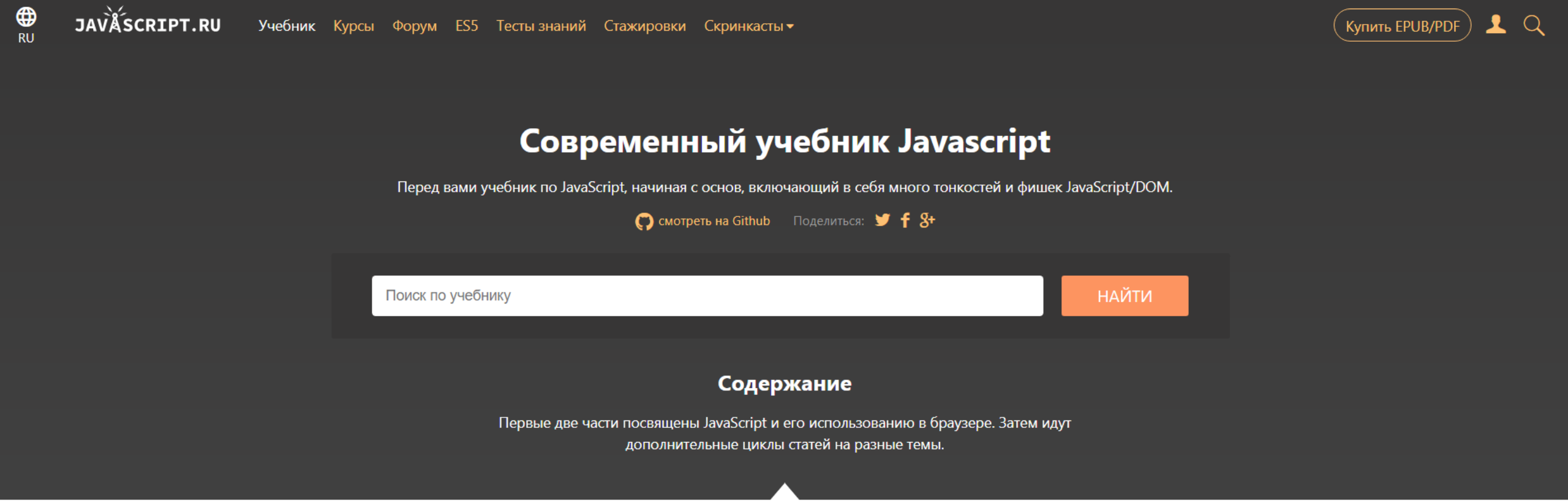
http://www.json.org/json-ru.html

Домашнее задание
/узнать



Дэвид Флэнаган *JavaScript. Подробное руководство*

Предварительные знания – лучший помощник в обучении, поэтому к следующему занятию жду, что **вы прочтёте 8 главу («Функции»), а также еще раз разделы 7.8 и 7.9 седьмой главы.**



Предварительные знания – лучший помощник в обучении, поэтому к следующему занятию жду, что **пройдёте разделы 2.17-2.20, 4.8, 4.9, 7.2, 7.3, 10.2-10.5, 10.10**.

http://learn.javascript.ru/

Домашнее задание | узнать

Узнать подробнее о деструктуризация массива

https://learn.javascript.ru/destructuring#massiv

Узнать о коллекции Map (ES-2015)

```
var mas = new Map();

mas.set("1", "AAA");
mas.set(1, "BBB");
mas.set(true, "CCC");

mas.get("1");
mas.get(1);
mas.get(true);
```

В коллекции **Map**, в отличии от ассоциативных массивов, ключом может выступать любое значение, любого типа. В отличии от массива с числовыми индексами количество элементов в **Map** можно узнать через свойство **.size**.

https://learn.javascript.ru/set-map

Узнать о коллекции Set (ES-2015)

```
var mas = new Set();

mas.add("AAA");
mas.add("BBB");
mas.add("AAA");
mas.add("CCC");

mas.size;
```

В коллекции **Set** предназначена для хранения множества значений, причём каждое значение может встречаться лишь один раз, в отличии от других коллекций ключей в **Set** нет. В отличии от массива с числовыми индексами количество элементов в **Set** можно узнать через свойство **.size**.

https://learn.javascript.ru/set-map

Где брать задачки для тренировки?

Чтобы научиться программировать – нужно тренироваться…
Чтобы тренироваться нужны задачи…
Чтобы были задачи нужно уметь программировать…

Д. М. Златопольский
Сборник задач по программированию

Prometheus CS50

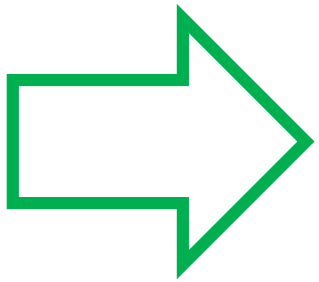
Лучшие основы программирования!

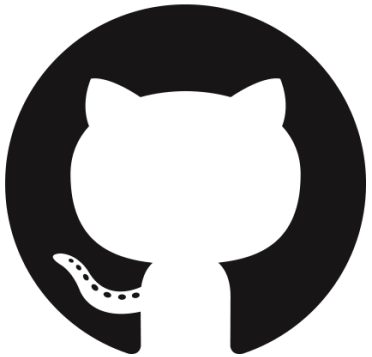
Домашнее задание
/сделать

Каждое домашнее задание оформляйте в виде отдельного репозитория на GitHub, в названии которого должно быть указан код задание (например: B3) код домашнего задания должен быть в ветке master

НЕ ЗАБЫВАЕМ ОБ ОТСТУПАХ В КОДЕ!!!

Ссылку на репозиторий сбрасывайте в Slack в канал #homeworks





Если есть проблемы, вопросы, трудности, делаем тоже самое – код с проблемой заливаем на **GitHub** и ссылку на него, с описанием вопроса в **Slack**, но в канал **#trouble**

Домашнее задание #B.1

```
<script>

    var phones = ["Apple", "Nokia", "Samsung", "LG", "HTC", "Lenovo", "Sony", "Fly",
    "Microsoft", "Huawei"].map(function(element){
        return [element, (Math.random() * 1000).toFixed(0) + "$" ];
    });

    phones.forEach(function(element){ console.log(element);});

    //phones содержит массив со списком телефонов, каждый элемент которого тоже массив
    состоящий их названия и цены телефона. Вам необходимо написать код который
    отсортирует массив phones по возврастанию цены. Цены генерируются автоматически
    (каждый раз разные). Ваш код можно писать только в отведённом месте, править
    остальной код нельзя. Вид массива [["Apple", "233$"], ["Nokia", "315$"]... и т.д.]
    //Код можно писать после этой строки//


    //Код можно писать до этой строки//

    console.log("Вывод после сортировки: ");
    phones.forEach(function(element){ console.log(element);});

</script>
```

Заготовка лежит в репозитории: ./homework/b1.html

Задача: отсортировать массив со списком телефонов по возрастанию цены.

Домашнее задание #B.2

Разработать скрипт, проверяющий знания (умение) таблицы умножения двузначных чисел. Скрипт должен задать пользователю 12 задач на умножение **двузначных** чисел. По результатам проверки, пользователю выставляется оценка (по 12 бальной шкале), а также выводиться два списка: верных ответов, и ошибочных ответов, с указанием какой ответ был правильный.

Домашнее задание #B.3

Кредитный калькулятор v.2

Заданы: сумма кредита, годовая процентная ставка, и срок кредитования в месяцах. **Рассчитать** ежемесячные платежи (сколько в каждом месяце будет платить заёмщик, указав сколько из суммы ежемесячного платежа идёт на погашение тела кредита, а сколько на погашение процентов) по **аннуитетной** схеме.