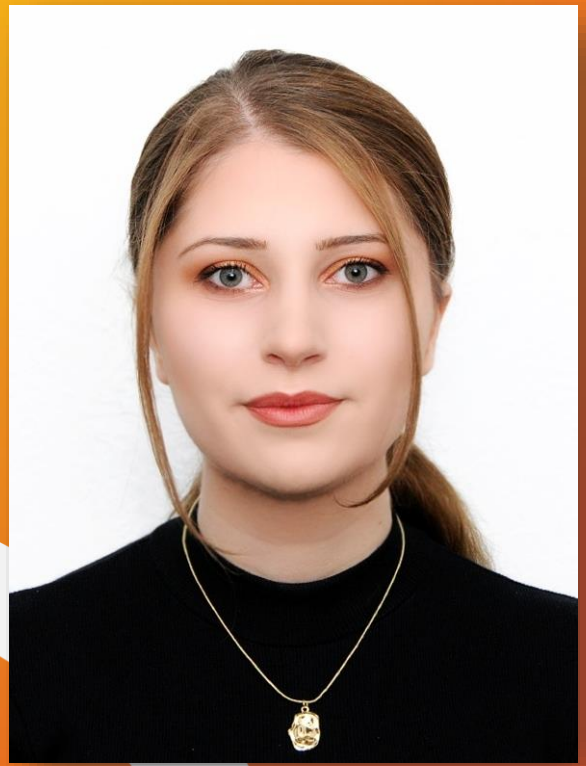
TEL-RAN
by Starta Institute

Frontend Basic
LESSON 8

Anna Khachaturyan



TEL-RAN
by Starta Institute

Анна Хачатурян

Front End/Gen Tech Teacher

- Since 2018 in IT
- Full Stack Developer at Web Magnat
- QA Engineer/Web Developer at Central Bank of RA
- Lecturer at Plekhanov Russian University of Economics
- TA at Picsart Academy
- Teacher at Tel-Ran

https://t.me/anny_khachaturyan

ВАЖНО:

- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

TEL-RAN
by Starta Institute

ПЛАН ЗАНЯТИЯ

1. Повторение изученного
2. Вопросы по повторению
3. Условное ветвление
4. Массивы
5. Циклы
6. Практика

TEL-RAN
by Starta Institute

1 ПОВТОРЕНИЕ

Повторение

TEL-RAN
by Starta Institute

- Чтобы включить в HTML-документ JavaScript-код из внешнего файла, нужно использовать атрибут **src** (source) тега <**script**>.

- **Консоль** — это инструмент разработчика, который помогает тестировать код. Если во время выполнения скрипта возникнет ошибка, в консоли появится сообщение о ней. А ещё в консоль можно выводить текстовые подсказки.
  Чтобы вывести сообщение в консоль, нужно использовать **console.log**.

- В JavaScript переменную можно создать/объявить, с помощью ключевого слова **let**.

- Функция **typeof** возвращает тип аргумента. Это полезно, когда мы хотим обрабатывать значения различных типов по-разному или просто хотим сделать проверку.

- Система преобразования типов в JavaScript очень проста, но отличается от других языков. Всего есть три преобразования:

    1. *Строковое преобразование.*
    2. *Численное преобразование.*
    3. *Преобразование к логическому значению.*

TEL-RAN
by Starta Institute

2 ВОПРОСЫ ПО ПОВТОРЕНИЮ

TEL-RAN
by Starta Institute

3 УСЛОВНОЕ ВЕТВЛЕНИЕ

Условное ветвление

TEL-RAN
by Starta Institute

Иногда нам нужно выполнить различные действия в зависимости от условий.

Для этого мы можем использовать инструкцию if.

Инструкция if(…) вычисляет условие в скобках и, если результат true, то выполняет блок кода.

Например:

```
let year = prompt('В каком году была опубликована спецификация ECMAScript-2015?');

if (year == 2015){
    console.log(‘Да');
}
```

Преобразование к логическому типу

Инструкция if (…) вычисляет выражение в скобках и преобразует результат к логическому типу.

Правила преобразования типов:

- Число 0, пустая строка "", null, undefined и NaN становятся false. Из-за этого их называют «ложными» («falsy») значениями.
- Остальные значения становятся true, поэтому их называют «правдивыми» («truthy»).

Таким образом, код при таком условии никогда не выполнится:

```
if (0) { // 0 is falsy
    ...
}
```

…а при таком – выполнится всегда:

```
if (1) { // 1 is truthy
    ...
}
```

Блок «else»

Инструкция if может содержать необязательный блок «**else**» («**иначе**»). Он выполняется, когда условие ложно.

Например:

```
let year = prompt('В каком году была опубликована спецификация ECMAScript-2015?');

if (year == 2015) {
    console.log('Да вы знаток!');
}
else {
    console.log('А вот и неправильно!'); // любое значение, кроме 2015
}
```

Несколько условий: «else if»

Иногда, нужно проверить несколько вариантов условия. Для этого используется блок else if.

Например:

```
let year = prompt('В каком году была опубликована спецификация ECMAScript-2015?');

if (year < 2015) {
    console.log('Это слишком рано...');
}
else if (year > 2015) {
    console.log('Это поздновато');
}
else {
    console.log('Верно!');
}
```

В приведённом выше коде JavaScript сначала проверит year < 2015. Если это неверно, он переходит к следующему условию year > 2015. Если оно тоже ложно, тогда сработает последний console.log.

Блоков else if может быть и больше. Присутствие блока else не является обязательным.

TEL-RAN
by Starta Institute

4 МАССИВЫ

Массивы

Массив (Array) в JavaScript является глобальным объектом, который используется для создания массивов; которые представляют собой высокоуровневые спископодобные объекты.

Создание массива

```
let fruits = ['Яблоко', 'Банан'];

console.log(fruits.length); //длина массива
// 2
```

Доступ к элементу массива по индексу

```
let first = fruits[0];
// Яблоко

let last = fruits[fruits.length - 1];
// Банан
```

Добавление элемента в конец массива

```
fruits.push('Апельсин');
// ["Яблоко", "Банан", "Апельсин"]
```

Удаление последнего элемента массива

```
fruits.pop(); // удалим Апельсин (из конца)
// ["Яблоко", "Банан"];
```

Удаление первого элемента массива

```
fruits.shift(); // удалим Яблоко (из начала)
// ["Банан"];
```

Добавление элемента в начало массива

```
fruits.unshift('Клубника') // добавляет в начало
// ["Клубника", "Банан"];
```

Поиск номера элемента в массиве

```
fruits.push('Манго');
// ["Клубника", "Банан", "Манго"]
let pos = fruits.indexOf('Банан');
// 1
```

Удаление элемента с определённым индексом

```
let removedItem = fruits.splice(pos, 1); // так можно удалить элемент

// ["Клубника", "Манго"]
```

Удаление нескольких элементов, начиная с определённого индекса

```
let vegetables = ['Капуста', 'Репа', 'Редиска', 'Морковка'];
console.log(vegetables);
// ["Капуста", "Репа", "Редиска", "Морковка"]
let pos = 1, n = 2;

let removedItems = vegetables.splice(pos, n);
// так можно удалить элементы, n определяет количество элементов для удаления,
// начиная с позиции(pos) и далее в направлении конца массива.

console.log(vegetables);
// ["Капуста", "Морковка"] (исходный массив изменён)

console.log(removedItems);
// ["Репа", "Редиска"]
```

Программа, которая считывает три числа через prompt и добавляет их в массив.

```
let arr = [];
let num1 = Number(prompt('Enter the number.'));
let num2 = Number(prompt('Enter the number.'));
let  num3  =  Number(prompt('Enter  the  number.'));
arr.push(num1, num2, num3);
console.log(arr);
```

Инкремент и декремент

Из языка Си в JavaScript перекочевали две операции: **инкремент** ++ и **декремент** --, которые очень часто встречаются вместе с циклами. Эти унарные операции увеличивают и уменьшают на единицу число, записанное в переменную:

```
let i = 0;
i++; // 0
i++; // 1

i--; // 2
i--; // 1
```

Кроме постфиксной формы, у них есть и префиксная:

```
let i = 0;
++i; // 1
++i; // 2

--i; // 1
--i; // 0
```

При использовании префиксной нотации сначала происходит изменение переменной, а потом возврат.

При использовании постфиксной нотации — наоборот: можно считать, что сначала происходит возврат, а потом изменение переменной.

TEL-RAN
by Starta Institute

5 | ЦИКЛЫ

Циклы

Последовательность инструкций, предназначенная для многократного исполнения, называется **телом цикла**. Единичное выполнение тела цикла называется **итерацией**. Выражение, определяющее, будет в очередной раз выполняться итерация или цикл завершится, называется **условием выхода** или условием окончания цикла (либо условием продолжения в зависимости от того, как интерпретируется его истинность – как признак необходимости завершения или продолжения цикла). Переменная, хранящая текущий номер итерации, называется **счётчиком** итераций цикла или просто счётчиком цикла. Цикл не обязательно содержит счётчик, счётчик не обязан быть один – условие выхода из цикла может зависеть от нескольких изменяемых в цикле переменных, а может определяться внешними условиями (например, наступлением определённого времени), в последнем случае счётчик может вообще не понадобиться.

Исполнение любого цикла включает первоначальную инициализацию переменных цикла, проверку условия выхода, исполнение тела цикла и обновление переменной цикла на каждой итерации. Кроме того, большинство языков программирования предоставляет средства для досрочного управления циклом, например, операторы завершения цикла, то есть выхода из цикла независимо от истинности условия выхода (break) и операторы пропуска итерации (continue).

Синтаксис СИ подобного цикла.

```
// Цикл FOR

// Синтаксис
for(Начало; Условие; Шаг){
    // тело цикла
    // тут будет выполняться код
}
```

Пример:

```
for(let num = 0; num < 5; num++){
    console.log(num);
}

// Работа цикла for:
// 1) Выполняется начало - let num = 0
// 2) Выполняется условие - num < 5
// 3) Если условие true выполняется
//    тело цикла - console.log(num)
// 4) Выполняется шаг - num++
// Повтор начиная с пункта 2)
```

Break - Досрочно прекращает работу

```
for(let num = 0; num < 5; num++){
    console.log(num);
    if (num == 2){
        break;
    }
}
```

Continue - Инструкция continue прерывает выполнение текущей итерации текущего или отмеченного цикла, и продолжает его выполнение на следующей итерации.

```
for(let num = 0; num < 5; num++){
    if (num == 2){
        continue;
    }
    console.log(num);
}
```

Пример вывода всех элементов массива при помощи СИ подобного цикла:


```
let arr = [];

for (let num = 0; num < 3; num++){
    arr.push(Number(prompt('Array')));
}

console.log(arr);
```

Пример: Написать цикл, который выводит только положительные числа из массива

```
let arr = [1, 0, 13, -40, 0, -1, 3, 6];

for(let num = 0; num < arr.length; num++){
    if(arr[num] >= 0){
        console.log(arr[num]);
    }
}
```

Синтаксиса цикла от большего к меньшему

Числа от 10 до 0

```
for (let num = 10; num > 0; num--){
    console.log(num);
}
```

TEL-RAN
by Starta Institute

6 ПРАКТИКА

Задание. Написать программу, которая получает два числа и выводит наибольшее.

Задание. Написать программу, которая считывает через prompt одно число и выводит одну из строк “число положительное”, “число отрицательное”, “число равно нулю”

Задание: Написать программу, в которой объявлен массив с 5 числовыми элементами. Программа должна заполнить новый пустой массив квадратами чисел из первого массива.

Пример:

Исходный массив [1, 4, 2, 5, 3]

Итоговый массив [1, 16, 4, 25, 9]

Задание:

1. Создайте массив styles с элементами «Джаз» и «Блюз».
2. Добавьте «Рок-н-ролл» в конец.
3. Замените значение в середине на «Классика».
4. Удалите первый элемент массива и покажите его.
5. Вставьте Рэп и Регги в начало массива.

Массив по ходу выполнения операций:

1. Джаз, Блюз
2. Джаз, Блюз, Рок-н-ролл
3. Джаз, Классика, Рок-н-ролл
4. Классика, Рок-н-ролл
5. Рэп, Регги, Классика, Рок-н-ролл

Домашнее задание

1. Составьте программу, которая присваивает переменной d значение 7, а затем выводит на экран: в первой строке - это значение, во второй – квадрат этого значения, в третьей – куб этого значения.
2. Составьте программу, которая принимает с клавиатуры целое число и, если оно положительное, увеличивает его вдвое.
   Программа должна выводить на экран новое значение.
3. Составьте программу, которая принимает с клавиатуры два целых числа и, если первое больше второго, выводит на экран их сумму, если же наоборот – выводит на экран их произведение. В случае же, если числа одинаковы, программа выводит на экран сообщение "числа одинаковые".
4. Составьте программу, которая принимает с клавиатуры целое число и выводит на экран его квадрат – но только в том случае, если введенное число отрицательно.
   В противном случае – на экран выводится сообщение "ошибка".
5. Составьте программу, которая принимает с клавиатуры два числа: первое – количество учеников в классе, второе – количество стульев в кабинете.
   Программа проверит соответствие между этими двумя значениями и выведет на экран соответствующую информацию.
   ввод: 24, 28 ⇒ вывод: стульев хватает; ввод: 24, 22 ⇒ вывод: стульев не хватает)
6. Составьте программу, которая выводит на экран все однозначные положительные числа в возрастающем порядке.
   Перед началом вывода на экран следует вывести "старт", а после окончания вывода чисел – "финиш".
   вывод: старт, 1, ... 9, финиш)
7. Составьте программу, которая выводит на экран все двузначные положительные числа, делящиеся без остатка на 5 (начиная с наименьшего).
8. Написать цикл, который выводит те числа из массива, которые больше или равны 15.
9. Написать цикл, который выводит только нечетные числа
10. Вывести только те значения массива, индекс которых кратен трем

Ребята!
Вы
просто
молодцы!