

## Выполнение заданий

- 1) Открыть браузер (например, Chrome)
- 2) Нажать F12 (вызвать консоль разработчика)
- 3) Перейти на вкладку «Console»



Рисунок 1 – Вызов консоли

Исходный код набирать либо в редакторе и вставлять в консоль, либо прямо в консоли. Enter – выполнить код, shift +Enter – перейти на новую строку.

Для очистки консоли есть специальная кнопка либо ПКМ – очистить консоль (в разных браузерах называется по-разному).

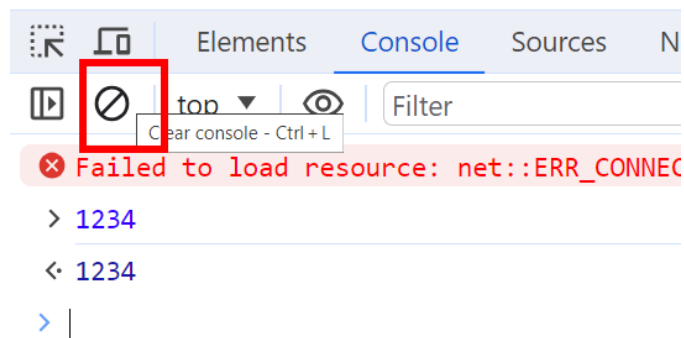


Рисунок 2 – Очистить результат вывода консоли

# Вариант 1

Смотреть раздел [Выполнение заданий](#)

## Задание 1

Создать массив `a = [1, 2, 'task 1']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

## Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 1']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

## Задание 3

Дан исходный массив `a = ['k1', 'K12', 'nasdBc', 'bac', 'BCq', 'task1']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k1» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

## Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 1'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 2

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 2']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 2']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k2', 'K22', 'nasdBc', 'bac', 'BCq', 'task2']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k2» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 2'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 3

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 3']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 3']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k3', 'K32', 'nasdBc', 'bac', 'BCq', 'task3']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «к3» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 3'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 4

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 4']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 4']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k4', 'K42', 'nasdBc', 'bac', 'BCq', 'task4']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k4» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 4'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 5

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 5']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 5']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k5', 'K52', 'nasdBc', 'bac', 'BCq', 'task5']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k5» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 5'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 6

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 6']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 6']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k6', 'K62', 'nasdBc', 'bac', 'BCq', 'task6']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k6» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 6'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 7

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 7']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 7']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k7', 'K72', 'nasdBc', 'bac', 'BCq', 'task7']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «к7» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 7'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```



## Вариант 8

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 8']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 8']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k8', 'K82', 'nasdBc', 'bac', 'BCq', 'task8']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k8» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 8'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 9

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 9']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 9']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k9', 'K92', 'nasdBc', 'bac', 'BCq', 'task9']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k9» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 9'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 10

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 10']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 10']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k10', 'K102', 'nasdBc', 'bac', 'BCq', 'task10']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k10» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 10'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

# Вариант 11

Смотреть раздел [Выполнение заданий](#)

## Задание 1

Создать массив `a = [1, 2, 'task 11']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

## Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 11']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

## Задание 3

Дан исходный массив `a = ['k11', 'K112', 'nasdBc', 'bac', 'BCq', 'task11']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k11» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

## Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 11'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 12

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 12']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 12']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k12', 'K122', 'nasdBc', 'bac', 'BCq', 'task12']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k12» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 12'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 13

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 13']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 13']`:

```
▲ ▶ 1
```

```
▲ ▶ 2
```

```
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 13']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
```

```
◀ ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k13', 'K132', 'nasdBc', 'bac', 'BCq', 'task13']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k13» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
```

```
> b
```

```
◀ ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 13'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
```

```
◀ ▶ (2) ['key1', 'key2']
```

```
◀ ▶ (2) [111, 'task 1']
```

```
jsonStr
```

```
'{"key1":111,"key2":"task 1"}'
```

## Вариант 14

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 14']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 14']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k14', 'K142', 'nasdBc', 'bac', 'BCq', 'task14']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k14» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 14'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 15

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 15']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 15']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k15', 'K152', 'nasdBc', 'bac', 'BCq', 'task15']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k15» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 15'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```



## Вариант 16

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 16']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)  
for .. in  
for .. of  
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1  
⚠ ▶ 2  
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 16']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])  
⏪ ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k16', 'K162', 'nasdBc', 'bac', 'BCq', 'task16']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k16» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']  
> b  
⏪ ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 16'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}  
⏪ ▶ (2) ['key1', 'key2']  
⏪ ▶ (2) [111, 'task 1']  
jsonStr  
'{"key1":111,"key2":"task 1"}'
```

## Вариант 17

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 1']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 1']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k17', 'K172', 'nasdBc', 'bac', 'BCq', 'task17']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k17» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 17'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 18

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 18']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 18']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k18', 'K182', 'nasdBc', 'bac', 'BCq', 'task18']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k18» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 18'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 19

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 19']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
▲ ▶ 1
▲ ▶ 2
▲ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 19']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k19', 'K192', 'nasdBc', 'bac', 'BCq', 'task19']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k19» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 19'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```

## Вариант 20

Смотреть раздел [Выполнение заданий](#)

### Задание 1

Создать массив `a = [1, 2, 'task 20']`. Вывести значения (индексы выводить не надо) в `console.warn()` при помощи четырех типов циклов:

```
for (let i = 0; i < ...)
for .. in
for .. of
forEach
```

В отчет предоставить исходный код четырех циклов и результат выполнения (скриншот консоли с кодом цикла и его результатом). Результат вывода будет одинаковым. Пример для массива `[1, 2, 'task 1']`:

```
⚠ ▶ 1
⚠ ▶ 2
⚠ ▶ task 1
```

### Задание 2

Написать функцию `userReverse(arr)` (аналог встроенной `reverse()`). Функция принимает на вход массив `arr`, возвращает новый массив (исходный не изменяется) с обратным порядком элементов. Можно использовать цикл `for` () с итерацией от `(arr.length - 1)` до 0, метод `push`. Массив для проверки `[1, 2, 'task 20']`.

В отчет исходный код и скриншот консоли.

Пример вывода для массива `[1, 2, 3]`:

```
> userReverse([1, 2, 3])
< ▶ (3) [3, 2, 1]
```

### Задание 3

Дан исходный массив `a = ['k20', 'K202', 'nasdBc', 'bac', 'BCq', 'task20']`.

Создать массив `b`, который содержит такие элементы из массива `a`, значение которых включают в строку «k20» без учета регистра. Реализовать либо циклом, либо при помощи `filter()`. В лекции есть подходящий метод для поиска вхождения строки.

В отчет исходный код и скриншот консоли.

Пример для массива `['abc', 'uyu', 'nasdBc', 'bac', 'BCq']` и строки `bc`:

```
> a = ['abc', 'uyu', 'nasdBc', 'bac', 'BCq']
> b
< ▶ (3) ['abc', 'nasdBc', 'BCq']
```

### Задание 4

Создать объект `t` с парами ключ-значение: `key1: 111, key2: 'task 20'`.

Вывести в консоль ключи этого объекта, затем вывести значения этого объекта.

Преобразовать в строку формата JSON и присвоить ее переменной `jsonStr`. Вывести ее в консоль. Вывести в консоль длину строки `jsonStr`.

В отчет исходный код и скриншот консоли.

Пример выводов для объекта `key1: 111, key2: 'task 1'`:

```
▶ {key1: 111, key2: 'task 1'}
< ▶ (2) ['key1', 'key2']
< ▶ (2) [111, 'task 1']
jsonStr
'{"key1":111,"key2":"task 1"}'
```