

Лабораторная работа № 13. `Array.prototype.includes()`, инфиксный оператор возведения в степень

Цель: изучить метод объектов `Array.prototype.includes()` для типа `Array`; изучить инфиксный оператор для возведения в степень, заменяющий метод `Math.pow()`.

Теория

`Array.prototype.includes()`

Метод **`includes()`** — это метод объектов типа `Array`, который определяет, есть ли в массиве искомый элемент (он, в отличие от *`indexOf`*, подходит и для работы со значениями *`NaN`*).

Синтаксис:

```
arr.includes(searchElement[, fromIndex = 0])
```

Параметры:

- *`searchElement`* – искомый элемент
- *`fromIndex`* (необязательный) – позиция в массиве, с которой начинать поиск элемент *`searchElement`*.

При отрицательных значениях поиск производится начиная с индекса *`arr.length + fromIndex`* по возрастанию. Значение по умолчанию равно 0. Метод возвращает значение типа `boolean` (`true` or `false`).

Разница при использовании методов *`indexOf()`* и *`includes()`*:

```
var numbers = [1, 2, 13, 8, 6];
if (numbers.indexOf(1) !== -1) {
  // ...
}
//тоже самое при использовании includes()
if (numbers.includes(1)) {
  // ...
}
```

Пример кода, если массив содержит значение *`NaN`*:

```
var numbers = [1, 2, 13, 8, NaN];
if (numbers.indexOf(NaN) !== -1) {
  // Код не будет работать, вернёт значение '-1'
}
//тоже самое при использовании includes()
if (numbers.includes(NaN)) {
  // Код будет работать, вернёт значение 'true'
}
```

Инфиксный оператор возведения в степень

Математические операции, вроде сложения и вычитания, реализуются в JavaScript с помощью инфиксных операторов, таких, как, соответственно, «+» и «-». Существует и нашедший широкое применение в программировании инфиксный оператор, который используется для возведения в степень. Такой оператор, выглядящий как «**», был представлен в ECMAScript 2016, он может служить заменой *Math.pow()*.

Пример кода при работе с *Math.pow()* и инфиксным оператором:

```
Math.pow(5,2)    // 25  
5**2             // 25
```

Задания к лабораторной работе № 13

Задание 1. Создать массив со случайными числами. Проверить, имеется ли у массива *n*-й элемент, введенный пользователем. Результат вывести на экран.

Задание 2. Создать массив со случайными числами и со значением *NaN*. Проверить, имеется ли у массива элемент *NaN* с помощью *InstanceOf* и *Array.prototype.includes*. Результат вывести на экран.

Задание 3. Создать несколько массивов со случайными символами. Проверить, имеется ли у массивов введенный пользователем символ. Результат вывести на экран.

Задание 4. Найти значение арифметических выражений. Использовать инфиксный оператор возведения в степень. Результат округлить до целого и вывести на экран.

Пример 1. $1234^2 * 23/5 + 234$

Пример 2. $28 * 765^2 / 5 * 43^2$

Пример 3. $3734^6 - (434^6 + 1024^4) / 5$

Пример 4. $6543^{35} - 965732$

Пример 5. $(1000^{10} + 100000^{10}) / 10000$