Console clear()

❮ Previous ❮ Console Object Reference Next ❯

Example

Clear all messages in the console:

```
console.clear();
```

Try it Yourself »

Description

The `clear()` method clears the console.

The `clear()` method also write "Console was cleared" in the console.

Syntax

```
console.clear()
```

JavaScript For In

❮ Previous Next ❯

The For In Loop

The JavaScript `for in` statement loops through the properties of an Object:

Syntax

```
for (key in object) {
  // code block to be executed
}
```

Example

```
const person = {fname:"John", lname:"Doe", age:25};

let text = "";
for (let x in person) {
  text += person[x];
}
```

Try it Yourself »

- Each iteration returns a **key** (x)
- The key is used to access the **value** of the key
- The value of the key is **person[x]**

For In Over Arrays

The JavaScript `for in` statement can also loop over the properties of an Array:

Syntax

```
for (variable in array) {
  code
}
```

Example

```
const numbers = [45, 4, 9, 16, 25];

let txt = "";
for (let x in numbers) {
  txt += numbers[x];
}
```

Try it Yourself »

Do not use **for in** over an Array if the index **order** is important.

The index order is implementation-dependent, and array values may not be accessed in the order you expect.

Array.forEach()

The `forEach()` method calls a function (a callback function) once for each array element.

Example

```
const numbers = [45, 4, 9, 16, 25];

let txt = "";
numbers.forEach(myFunction);

function myFunction(value, index, array) {
  txt += value;
}
```

Try it Yourself »

Note that the function takes 3 arguments:

- The item value
- The item index
- The array itself

The example above uses only the value parameter. It can be rewritten to:

Example

```
const numbers = [45, 4, 9, 16, 25];
```

JavaScript For Of

❮ Previous Next ❯

The For Of Loop

The JavaScript `for of` statement loops through the values of an iterable object.

It lets you loop over iterable data structures such as Arrays, Strings, Maps, NodeLists, and more:

Syntax

```
for (variable of iterable) {
  // code block to be executed
}
```

variable - For every iteration the value of the next property is assigned to the variable. *Variable* can be declared with `const`, `let`, or `var`.

iterable - An object that has iterable properties.

Browser Support

For/of was added to JavaScript in 2015 (ES6)

Chrome 38	Edge 12	Firefox 51	Safari 7	Opera 25
Oct 2014	Jul 2015	Oct 2016	Oct 2013	Oct 2014

For/of is not supported in Internet Explorer.

Looping over an Array

Example

```
const cars = ["BMW", "Volvo", "Mini"];

let text = "";
for (let x of cars) {
  text += x;
}
```

Try it Yourself »

Looping over a String

Example

```
let language = "JavaScript";

let text = "";
for (let x of language) {
```

JavaScript parseFloat()

❮ Previous JavaScript Global Methods Next ❯

Examples

Parse different values:

```
parseFloat(10);
parseFloat("10");
parseFloat("10.33");
parseFloat("34 45 66");
parseFloat("He was 40");
```

Try it Yourself »

More examples below.

Description

The `parseFloat()` method parses a value as a string and returns the first number.

Notes

JavaScript parseInt()

❮ Previous JavaScript Global Methods Next ❯

Example

Parse different values:

```
parseInt("10");
parseInt("10.00");
parseInt("10.33");
parseInt("34 45 66");
parseInt(" 60 ");
parseInt("40 years");
parseInt("He was 40");
```

Try it Yourself »

Description

The `parseInt` method parses a value as a string and returns the first integer.

A radix parameter specifies the number system to use:

2 = binary, 8 = octal, 10 = decimal, 16 = hexadecimal.

JavaScript Number toFixed()

❮ Previous JavaScript Number Reference Next ❯

Examples

```
let num = 5.56789;
let n = num.toFixed();
```

Try it Yourself »

```
let num = 5.56789;
let n = num.toFixed(2);
```

Try it Yourself »

More examples below

Description

The `toFixed()` method converts a number to a string.

The `toFixed()` method rounds the string to a specified number of decimals.

If the number of decimals are higher than in the number, zeros are added.

Syntax

```
number.toFixed(x)
```

Parameters

Parameter	Description
x	Optional. Number of decimals. Default is 0 (no decimals)

Return Value

Type	Description
A string	The representation of a number with (or without) decimals.

More Examples

Round to 10 decimals

```
let num = 5.56789;
let n = num.toFixed(10);
```

JavaScript Array push()

❮ Previous JavaScript Array Reference Next ❯

Examples

Add a new item to an array:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Kiwi");
```

Try it Yourself »

Add two new items to the array:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Kiwi", "Lemon");
```

Try it Yourself »

Description

The `push()` method adds new items **to the end** of an array.

The `push()` method changes the length of the array.

JavaScript Object Properties

Accessing JavaScript Properties

The syntax for accessing the property of an object is:

```
// objectName.property
let age = person.age;
```

or

```
//objectName["property"]
let age = person["age"];
```

or

```
//objectName[expression]
let age = person[x];
```

Examples

```
person.firstname + " is " + person.age + " years old.";
```

Try it Yourself »

```
person["firstname"] + " is " + person["age"] + " years old.";
```

Try it Yourself »

Pro Teams Pricing Documentation

npm

Sign Up Sign In

Search packages Search

readline-sync DT

1.4.10 • Public • Published 6 years ago

Readme

Code Beta

0 Dependencies

4,194 Dependents

65 Versions

readlineSync

npm v1.4.10 issues 0 open dependencies No dependency license MIT

Synchronous **Readline** for interactively running to have a conversation with the user via a console(TTY).

readlineSync tries to let your script have a conversation with the user via a console, even when the input/output stream is redirected like `your-script <foo.dat >bar.log` .

Basic Options	Utility Methods	Placeholders

- Simple case:

```
var readlineSync = require('readline-sync');
```

```
// Wait for user's response.
var userName = readlineSync.question('May I have your name? ');
console.log('Hi ' + userName + '!');

// Handle the secret text (e.g. password).
var favFood = readlineSync.question('What is your favorite food? ', {
  hideEchoBack: true // The typed text on screen is hidden by `*` (def
});
console.log('Oh, ' + userName + ' loves ' + favFood + '!');
```

```
May I have your name? CookieMonster
Hi CookieMonster!
What is your favorite food? ****
Oh, CookieMonster loves tofu!
```

- Get the user's response by a single key without the Enter key:

```
var readlineSync = require('readline-sync');
if (readlineSync.keyInYN('Do you want this module?')) {
  // 'Y' key was pressed.
  console.log('Installing now...');
  // Do something...
} else {
  // Another key was pressed.
  console.log('Searching another...');
  // Do something...
}
```

- Let the user choose an item from a list:

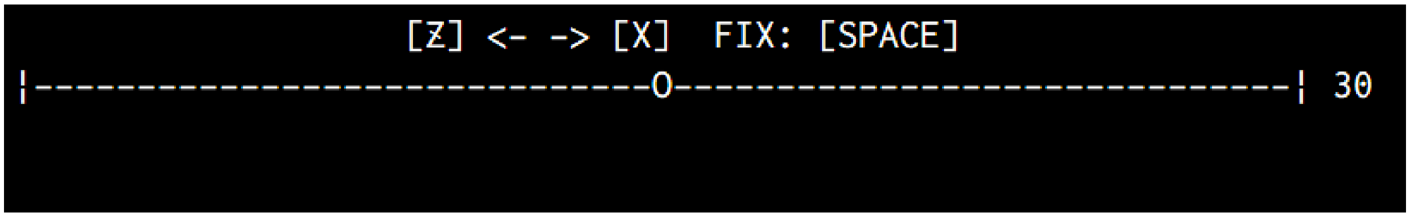
```
var readlineSync = require('readline-sync'),
  animals = ['Lion', 'Elephant', 'Crocodile', 'Giraffe', 'Hippo'],
  index = readlineSync.keyInSelect(animals, 'Which animal?');
console.log('Ok, ' + animals[index] + ' goes to your room.');
```

```
[1] Lion
[2] Elephant
[3] Crocodile
[4] Giraffe
[5] Hippo
[0] CANCEL

Which animal? [1...5 / 0]: 2
Ok, Elephant goes to your room.
```

- An UI like the Range Slider:
  (Press `Z` or `X` key to change a value, and Space Bar to exit)

```
var readlineSync = require('readline-sync'),
  MAX = 60, MIN = 0, value = 30, key;
console.log('\n\n' + (new Array(20)).join(' ') +
  '[Z] <- -> [X]  FIX: [SPACE]\n');
while (true) {
  console.log('\x1B[1A\x1B[K|' +
    (new Array(value + 1)).join('-') + 'O' +
    (new Array(MAX - value + 1)).join('-') + '| ' + value);
  key = readlineSync.keyIn('',
    {hideEchoBack: true, mask: '', limit: 'zx '});
  if (key === 'z') { if (value > MIN) { value--; } }
  else if (key === 'x') { if (value < MAX) { value++; } }
  else { break; }
}
console.log('\nA value the user requested: ' + value);
```



```
        [Z] <- -> [X]  FIX: [SPACE]
¦------------------------------O------------------------------¦ 30
```

- Handle the commands repeatedly, such as the shell interface:

```
readlineSync.promptCLLoop({
  add: function(target, into) {
    console.log(target + ' is added into ' + into + '.');
    // Do something...
  },
  remove: function(target) {
    console.log(target + ' is removed.');
    // Do something...
  },
  bye: function() { return true; }
});
console.log('Exited');
```

```
> add pic01.png archive
pic01.png is added into archive.
> delete pic01.png
Requested command is not available.
> remove pic01.png
pic01.png is removed.
> bye
Exited
```

Installation

```
npm install readline-sync
```