



Die Ruby-Dokumentation

Die Ruby-Dokumentation

<https://ruby-doc.org/>

- Listet und beschreibt Core-Klassen und Standardbibliothek
- Für jede aktuelle Ruby-Version verfügbar
- Oft mit zusätzlichen Code-Beispielen

Help and documentation for the Ruby programming language.



Core API

These are the API documents for the base classes and modules in the current stable release of Ruby 2.5.

- [2.5.1 core](#) - Core API docs for Ruby 2.5.1 This is the current official release.
- [2.6.0.preview2 core](#) - Core API docs for Ruby 2.6.0.preview2
- [2.5.0 core](#) - Core API docs for Ruby 2.5.0
- [2.4.4 core](#) - Core API docs for Ruby 2.4.4
- [2.4.3 core](#) - Core API docs for Ruby 2.4.3
- [2.4.2 core](#) - Core API docs for Ruby 2.4.2
- [2.4.1 core](#) - Core API docs for Ruby 2.4.1
- [2.4.0 core](#) - Core API docs for Ruby 2.4.0
- [2.3.7 core](#) - Core API docs for Ruby 2.3.7
- [2.3.6 core](#) - Core API docs for Ruby 2.3.6
- [2.3.5 core](#) - Core API docs for Ruby 2.3.5
- [2.3.4 core](#) - Core API docs for Ruby 2.3.4
- [2.3.3 core](#) - Core API docs for Ruby 2.3.3
- [2.3.2 core](#) - Core API docs for Ruby 2.3.2

Standard Library API

These are the API documents for the standard library classes and modules in version 2.5

- [The 2.5.1 standard library \(this is the current official release\)](#)
Additional libraries included with the standard Ruby distribution, such as CGI, OpenURI, and REXML
- [The 2.6.0.preview2 standard library](#)
- [The 2.5.0 standard library](#)
- [The 2.4.4 standard library](#)
- [The 2.4.3 standard library](#)
- [The 2.4.2 standard library](#)
- [The 2.4.1 standard library](#)
- [The 2.4.0 standard library](#)
- [The 2.3.7 standard library](#)
- [The 2.3.6 standard library](#)
- [The 2.3.5 standard library](#)
- [The 2.3.4 standard library](#)
- [The 2.3.3 standard library](#)

Ruby 2.5.1



Ruby 2.5.1

New to Ruby? You may find these links helpful: [syntax](#), [control expressions](#), [assignment](#), [methods](#), [modules + classes](#), and [operator precedence](#).

This is the API documentation for Ruby 2.5.1.

Files

[.lib/racc/rdoc/grammar.en.rdoc](#)

[.test/rdoc/test.ja.rdoc](#)

[doc/contributing.rdoc](#)

[doc/contributors.rdoc](#)

[doc/dtrace_probes.rdoc](#)

[doc/extension.ja.rdoc](#)

[doc/extension.rdoc](#)

[doc/globals.rdoc](#)

[doc/keywords.rdoc](#)

[doc/maintainers.rdoc](#)

[sample/drb/README.ja.rdoc](#)

[sample/drb/README.rdoc](#)

Classes

- C [ARGF](#)
- C [ArgumentError](#)
- C [Array](#)
- C [BasicObject](#)
- C [Binding](#)
- C [Class](#)
- C [ClosedQueueError](#)
- M [Comparable](#)
- C [Complex](#)
- C [Complex::compatible](#)
- C [ConditionVariable](#)
- C [Continuation](#)
- C [Data](#)
- C [Dir](#)
- C [ENV](#)
- C [EOFError](#)
- C [Encoding](#)

Methods

- :: [=== \(SystemCallError\)](#)
- :: [DEBUG \(Thread\)](#)
- :: [DEBUG= \(Thread\)](#)
- :: [\[\] \(Array\)](#)
- :: [\[\] \(Dir\)](#)
- :: [\[\] \(ENV\)](#)
- :: [\[\] \(Hash\)](#)
- :: [\[\]= \(ENV\)](#)
- :: [_id2ref \(ObjectSpace\)](#)
- :: [abort \(Process\)](#)
- :: [abort_on_exception \(Thread\)](#)
- :: [abort_on_exception= \(Thread\)](#)
- :: [absolute_path \(File\)](#)
- :: [acos \(Math\)](#)
- :: [acosh \(Math\)](#)
- :: [add_stress_to_class \(GC\)](#)
- :: [aliases \(Encoding\)](#)

[sample/drb/README.ja.rdoc](#)

[sample/drb/README.rdoc](#)

Classes **Arr**

[C](#) [Array](#)

Methods **filter...**

- [:: === \(SystemCallError\)](#)
- [:: DEBUG \(Thread\)](#)
- [:: DEBUG= \(Thread\)](#)
- [:: \[\] \(Array\)](#)
- [:: \[\] \(Dir\)](#)
- [:: \[\] \(ENV\)](#)
- [:: \[\] \(Hash\)](#)
- [:: \[\]= \(ENV\)](#)
- [:: _id2ref \(ObjectSpace\)](#)
- [:: abort \(Process\)](#)
- [:: abort_on_exception \(Thread\)](#)
- [:: abort_on_exception= \(Thread\)](#)
- [:: absolute_path \(File\)](#)
- [:: acos \(Math\)](#)
- [:: acosh \(Math\)](#)
- [:: add_stress_to_class \(GC\)](#)
- [:: aliases \(Encoding\)](#)



Class: Array (Ruby 2.5.1)

Returns a new **array**. In the first form, if no arguments are sent, the new **array** will be empty. When a size and an optional default are sent, an **array** is created with size copies of default.



<https://ruby-doc.org/core/Array.html>

class Array - Documentation for Ruby 2.0.0

Returns a new **array**. In the first form, if no arguments are sent, the new **array** will be empty. When a size and an optional obj are sent, an **array** is created with size copies of obj.



<https://docs.ruby-lang.org/en/2.0.0/Array.html>

Ruby - Arrays - tutorialspoint.com

Ruby arrays are ordered, integer-indexed collections of any object. Each element in an **array** is associated with and referred to by an index. **Array** indexing starts at 0, as in C or Java. A negative index is assumed relative to the end of the **array** --- that is, an index of -1 indicates the last ...



https://www.tutorialspoint.com/ruby/ruby_arrays.htm

How to Create Arrays in Ruby - ThoughtCo

Storing variables within variables is a common thing in **Ruby** and is often referred to as a "data structure." There are many varieties of data structures, the most simple of which is the **array**. Programs often have to manage collections of variables. For example, a program that manages your calendar ...



<https://www.thoughtco.com/how-to-create-arrays-in-ruby-2908192>

Ruby Arrays - RubyLearning

Ruby Arrays <Ruby Blocks | TOC | Ranges> An **Array** is just a list of items in order (like mangoes,

Array

Ruby

Arrays are ordered, integer-indexed collections of any object.

Array indexing starts at 0, as in C or Java. A negative index is assumed to be relative to the end of the array--- that is, an index of -1 indicates the last element of the array, -2 is the next to last element in the array, and so on.





[More at ruby-doc.org](#)

[Feedback](#)

Send feedback

Home Classes Methods

In Files

 array.c
 pack.c

Parent

Object

Methods

`::[]`
`::new`
`::try_convert`
`#&`
`#*`
`#+`
`#-`
`#<<`
`#<=>`
`#==`
`#[]`
`#[]=`
`#any?`
`#append`

Array

Arrays are ordered, integer-indexed collections of any object.

Array indexing starts at 0, as in C or Java. A negative index is assumed to be relative to the end of the array—that is, an index of -1 indicates the last element of the array, -2 is the next to last element in the array, and so on.

Creating Arrays ¶ ↑

A new array can be created by using the literal constructor `[]`. Arrays can contain different types of objects. For example, the array below contains an Integer, a String and a Float:

```
ary = [1, "two", 3.0] #=> [1, "two", 3.0]
```

An array can also be created by explicitly calling `::new` with zero, one (the initial size of the Array) or two arguments (the initial size and a default object).

```
ary = Array.new      #=> []  
Array.new(3)         #=> [nil, nil, nil]  
Array.new(3, true)   #=> [true, true, true]
```


Methods

`::[]`
`::new`
`try_convert`
`#&`
`#*`
`#+`
`#-`
`#<<`
`#<=>`
`#==`
`#[]`
`#[]=`
`#any?`
`#append`
`#assoc`
`#at`
`#bsearch`
`#bsearch_index`
`#clear`
`#collect`
`#collect!`
`#combination`
`#compact`
`#compact!`
`#concat`
`#count`
`#cycle`
`#delete`
`#delete_at`
`#delete_if`

contain different types of objects. For example, the array below contains an Integer, a String and a Float:

```
ary = [1, "two", 3.0] #=> [1, "two", 3.0]
```

An array can also be created by explicitly calling `::new` with zero, one (the initial size of the Array) or two arguments (the initial size and a default object).

```
ary = Array.new      #=> []  
Array.new(3)         #=> [nil, nil, nil]  
Array.new(3, true)   #=> [true, true, true]
```

Note that the second argument populates the array with references to the same object. Therefore, it is only recommended in cases when you need to instantiate arrays with natively immutable objects such as Symbols, numbers, true or false.

To create an array with separate objects a block can be passed instead. This method is safe to use with mutable objects such as hashes, strings or other arrays:

```
Array.new(4) { Hash.new }  #=> [{}, {}, {}, {}]  
Array.new(4) {|i| i.to_s } #=> ["0", "1", "2", "3"]
```

This is also a quick way to build up multi-dimensional arrays:

```
empty_table = Array.new(3) { Array.new(3) }  
#=> [[nil, nil, nil], [nil, nil, nil], [nil, nil, nil]]
```

#slice.
#sort
#sort!
#sort_by!
#sum
#take
#take_while
#to_a
#to_ary
#to_h
#to_s
#transpose
#uniq
#uniq!
#unshift
#values_at
#zip
#|

Included Modules

Enumerable

Files

grammar.en.rdoc
test.ja.rdoc
contributing.rdoc
contributors.rdoc
dtrace_probes.rdoc

`drop` does the opposite of `take`, by returning the elements after `n` elements have been dropped:

```
arr.drop(3) #=> [4, 5, 6]
```

Obtaining Information about an Array ¶ ↑

Arrays keep track of their own length at all times. To query an array about the number of elements it contains, use `length`, `count` or `size`.

```
browsers = ['Chrome', 'Firefox', 'Safari', 'Opera', 'IE']  
browsers.length #=> 5  
browsers.count #=> 5
```

To check whether an array contains any elements at all

```
browsers.empty? #=> false
```

To check whether a particular item is included in the array

```
browsers.include?('Konqueror') #=> false
```

Adding Items to Arrays ¶ ↑

Items can be added to the end of an array by using either `push` or `<<`

```
arr = [1, 2, 3, 4]  
arr.push(5) #=> [1, 2, 3, 4, 5]  
arr << 6    #=> [1, 2, 3, 4, 5, 6]
```

`unshift` will add a new item to the beginning of an array.

Array#select

 **select { |item| block } → new_ary**

 **select → Enumerator**

Returns a new array containing all elements of `ary` for which the given `block` returns a true value.

If no `block` is given, an Enumerator is returned instead.

```
[1,2,3,4,5].select { |num| num.even? }  #=> [2, 4]
```

```
a = %w{ a b c d e f }
```

```
a.select { |v| v =~ /[aeiou]/ }  #=> ["a", "e"]
```

See also Enumerable#select.



Die Ruby-Dokumentation