You are logged in as C. O'Keefe (Logout)

Core Ruby 41st Batch

Some useful Ruby methods

Method *
str * int --> string
The * instance method of class **String**, returns a new **String** containing int copies of the receiver. Here's an example:

```
"Ruby! " * 3 # => "Ruby! Ruby! Ruby! "
```

slice

The **String slice** method, if passed a single **Fixnum**, returns the code of the character at that position (**whereas in version 1.9, it returns the character at the position**). If passed two **Fixnum** objects, returns a substring starting at the offset given by the first, and a length given by the second.

See the following examples:

```
a = "hello there"
  a.slice(1) # => "e"
  a.slice(1,3) # => "ell"
  a.slice(-3,2) # => "er"
  a.slice(0) # => "h"
```

strip

str.strip -> string

Returns a copy of *str* with leading and trailing whitespace removed.

```
" hello ".strip #=> "hello"
```

My courses

Settings

Course administration

My profile settings

```
"\tgoodbye\r\n".strip #=> "goodbye"
```

count

The **Stringcount** method counts the number of occurrences of any of a set of specified characters.

```
str.count( string + ) -> int
```

Each string parameter defines a *set* of characters to count. The *intersection of these sets* defines the characters to count in str.

```
a = "hello world"a.count "lo" # => 5
a.count "lo", "o" # => 2
```

length

You (or your users) frequently misremember the name of a method. To reduce the confusion, you want to make the same method accessible under multiple names. Alternatively, you're about to redefine a method and you'd like to keep the old version available.

It's difficult to pick the perfect name for a method: you must find the word or short phrase that best conveys an operation on a data structure, possibly an abstract operation that has different "meanings" depending on context. Sometimes there will be no good name for a method and you'll just have to pick one; sometimes there will be too many good names for a method and you'll just have to pick one. In either case, your users may have difficulty remembering the "right" name of the method. You can help them out by creating aliases.

Some languages use length or len to find the length of a string, and some use size.

Do you remember how you can achieve this? Of course with **alias**. Ruby itself uses aliases in its standard library.

`str.length -> int`

Returns the length of str, where str is a string.

str.size `-> int`

Synonym for **String#length**.

Last modified: Friday, 1 June 2012, 10:08 PM

You are logged in as C. O'Keefe (Logout)

moodle