

Ruby - Strings - Methods I

Text info can be read from varied sources and is often unsuitable for direct processing or usage by core functions. This necessitates methods for post-processing and data-fixing. In this tutorial, we'll learn how to remove flanking whitespace and newline from strings.

- `String.chomp(separator=$/)` : Returns a new string with the given separator removed from the end of the string (if present). If `$/` has not been changed from the default Ruby record separator, then `chomp` also removes carriage return characters (that is, it will remove `\n`, `\r`, and `\r\n`).

```
> "Hello World! \r\n".chomp
"Hello World! "
> "Hello World!".chomp("ord!")
"Hello W"
> "hello \n there".chomp
"hello \n there"
```

- `String.strip` - Returns a new string with the leading and trailing whitespace removed.

```
> "  hello  ".strip
"hello"
> "\tgoodbye\r\n".strip
"goodbye"
```

- `String.chop` - Returns a new string with the last character removed. Note that carriage returns (`\n`, `\r\n`) are treated as single character and, in the case they are not present, a character from the string **will be removed**.

```
> "string\n".chop
"string"
> "string".chop
"strin"
```

In this challenge, your task is to code a `process_text` method, which takes an array of strings as input and returns a single joined string with all *flanking* whitespace and new lines removed. Each string has to be separated by a single space.

```
> process_text(["Hi, \n", " Are you having fun?  "])
"Hi, Are you having fun?"
```