

```

@new = ();
push(@new, $+) while $text =~ m{
    "([^\\"\\]*(?:\\\[^\\"\\]*)*)",? # groups the phrase inside the quotes
    | ([^,]+),?
    | ,
}gx;
push(@new, undef) if substr($text,-1,1) eq ',';

```

If you want to represent quotation marks inside a quotation-mark-delimited field, escape them with backslashes (eg, "like \"this\"").

Alternatively, the `Text::ParseWords` module (part of the standard Perl distribution) lets you say:

```

use Text::ParseWords;
@new = quotewords(",", 0, $text);

```

There's also a `Text::CSV` (Comma-Separated Values) module on CPAN.

#### 20.4.14 How do I strip blank space from the beginning/end of a string?

Although the simplest approach would seem to be

```
$string =~ s/^\s*(.*?)\s*$/$1/;
```

not only is this unnecessarily slow and destructive, it also fails with embedded newlines. It is much faster to do this operation in two steps:

```

$string =~ s/^\s+//;
$string =~ s/\s+$//;

```

Or more nicely written as:

```

for ($string) {
    s/^\s+//;
    s/\s+$//;
}

```

This idiom takes advantage of the `foreach` loop's aliasing behavior to factor out common code. You can do this on several strings at once, or arrays, or even the values of a hash if you use a slice:

```

# trim whitespace in the scalar, the array,
# and all the values in the hash
foreach ($scalar, @array, @hash{keys %hash}) {
    s/^\s+//;
    s/\s+$//;
}

```

#### 20.4.15 How do I pad a string with blanks or pad a number with zeroes?

In the following examples, `$pad_len` is the length to which you wish to pad the string, `$text` or `$num` contains the string to be padded, and `$pad_char` contains the padding character. You can use a single character string constant instead of the `$pad_char` variable if you know what it is in advance. And in the same way you can use an integer in place of `$pad_len` if you know the pad length in advance.

The simplest method uses the `sprintf` function. It can pad on the left or right with blanks and on the left with zeroes and it will not truncate the result. The `pack` function can only pad strings on the right with blanks and it will truncate the result to a maximum length of `$pad_len`.