Lesson 5, Week 1: Strings I (literal basics)

AIM

— to understand what a literal string is

— to learn to combine the string operators `*` and `^`

— to learn how to enter Unicode characters at the REPL

— to learn how to use `+` and `-` as operators on characters and integers

After this lesson, you will be able to

* build literal strings using characters, other strings, and operators
* insert some characters by another means than finding them on your keyboard
* do some very simple character arithmetic

String literals

Consider assignment of the string value `"abcdef"` to the variable `var1`, which as you know, is done with the code
`var1 = "abcdef"`

As you also know, this means that the string value `"abcdef"` is physically stored somewhere on your computer, where it takes up a small amount of memory. To be precise, the memory of your computer is modified so that in some part of it, the character `'a'` is followed by the character `'b'` and so on[1].

A string value stored in memory this way is a string literal.

The string operators `*` and `^`

When used as a string operator, `*` combines two strings. [DEMO: some more examples]
For example, `"Julia " * "programming"`

[1]Of course, what is actually stored is a sequence of 0s and 1s that Julia knows should be interpreted in a particular way.

The technical term is *concatenate*. Thus the in the code `"Hello" * ", " * "world"`, three strings are concatenated into one.

Note that for the purpose of concatenating strings, characters can also be used.

DEMO

What happens with code like `"a" * 'b'` is that first the character `'b'` is converted to the string `"b"` and then the two strings are joined.

[DEMO: more examples]

The string operator `^` is a bit different: its input is one string and one integer. Consider the code `"my" * "my" * "my"`
Here, a string literal is repeated three times. Instead of typing it out three times, one can just use `"my"^3`.

Note that these operators can also be used with variables whose values are string literals:

```
x, y, z = "string 1 ", "this is a string ", "but this?  what's this?"
x * y * z
(y*z)^3
```

Characters not directly from the keyboard

For example: you can get Greek letters from an standard keyboard that shows only Roman letters. The letter α is available as follows: at the REPL, type `\alpha` and then press Tab.

In fact, you can type only `\alp` and press Tab. The REPL then shows you all the options that start like that. Pick the one you want, type in the next letter, and try again. Continue until it completes correctly, then press Tab again.

Which characters are available by this method? There is no easy answer. If you know the LaTeXtypesetting system, you can try any character code from there, it should be available. You can also try to explore the system by typing just one or two letters after the backslash, and look at all the options[2].

Finally, for those of you who want more, some of Unicode is available. If you know a UTF-8 code of a character, you can try it. You still use a pair of single quotes, and inside it you put `\u...` where the dots stand for an integer with one to four hexadecimal digits. We won't use any of these characters on this course.

WARNING: some of these characters may be available at the REPL, but not in your text editor. In that case, you won't be able to use them in a `.jl` file. This is one of the differences between the REPL and code files to watch out for.

[2]Gets tedious quite quickly!

Some very simple character arithmetic

For completeness[3], we mention that `+` and `-` can act like operators on a character, for example `'a' + 1`.

As you see, this shifts the character to a nearby character—the distance is determined by the integer you add or subtract. Note that the order doesn't matter[4]: `'Z' + 22` gives the same result as `22 + 'Z'`.

Review and summary

* A string literal is simply all the actual characters of the string, all together in one place in the computer's memory
* The operator `*` concatenates strings
* The operator `^` makes a new string by repeating one string (inputs: string, number of repeats)
* Characters not on an international keyboard[5] are available via `\`, followed by some letters, followed by Tab.

[3] That is, we don't use this feature on this course, but it feels like a gap to leave it out.
[4] As opposed to `*` when it is used as a string operator.
[5] None of these are used on this course, except as examples.