

Getting to know glue

INTERMEDIATE REGULAR EXPRESSIONS IN R



Angelo Zehr
Instructor

Pasting is tedious

```
username <- "Adam"  
paste("Hi", username)
```

Will print: Hi Adam

```
library("glue")  
username <- "Adam"  
glue("Hi {username}")
```

Will also print: Hi Adam

What you pass to glue

```
username_1 <- "Adam"  
username_2 <- NA  
  
glue(  
  "Hi {username_1} and {username_2}",  
  .na = ""  
)
```

Will print: Hi Adam and

What is a template string?

```
username_1 <- "Adam"  
username_2 <- "Eve"  
  
glue("Hi {username_1} and {username_2}")
```

Will print: Hi Adam and Eve

`glue("{2 + 2}")` will print `"4"`

Temporary variables

```
glue(  
  "The train is {length} meters long.",  
  length = 50  
)
```

Will print:

```
"The train is 50 meters long."
```

Glue inside data frames

```
df %>%  
  mutate(  
    new_column = glue("Hi {username}")  
  )
```

Will return:

	username	new_column
1	"Adam"	"Hi Adam"

Let's practice!

INTERMEDIATE REGULAR EXPRESSIONS IN R

Collapsing multiple elements into a string

INTERMEDIATE REGULAR EXPRESSIONS IN R



Angelo Zehr
Data Journalist

Introducing glue_collapse

```
usernames <- c("Adam", "Betty", "Cora", "David")  
  
glue_collapse(usernames)
```

Will print AdamBettyCoraDavid

More elaborate example

```
glue_collapse(  
  usernames,  
  sep = ", ",  
  last = ", and ",  
  width = 27  
)
```

Will print Adam, Betty, Cora, and Dav...

Passing glue collapse to glue

```
glue(  
  "Hello {users}.",  
  users = glue_collapse(  
    usernames,  
    sep = ", ",  
    last = ", and "  
  )  
)
```

Will print Hello Adam, Betty, Cora, and David.

Collapsing columns of a data frame

Data frame `df`

```
x  y
1  4
2  5
3  6
```

`glue_collapse(df$x)` will print `123`

Let's practice!

INTERMEDIATE REGULAR EXPRESSIONS IN R

Gluing regular expressions

INTERMEDIATE REGULAR EXPRESSIONS IN R



Angelo Zehr
Data Journalist

Collapsing with the pipe

```
pattern = "Nemo|Harmony|Dory"
```

can also be created like this:

```
names <- c("Nemo", "Harmony", "Dory")
```

```
pattern = glue_collapse(names, sep = "|")
```

A quick refresh

Character Class	Name	Example
<code>\\d</code>	Digit	<code>0, 1, 2, 3, ...</code>
<code>\\w</code>	Word	<code>a, b, c..., 1, 2, 3..., _</code>
<code>\\s</code>	Space	<code>" "</code> , tabs and line breaks
<code>[A-Za-z]</code>	Letter	<code>A, B, C, ..., a, b, c, ...</code>

A quick refresh

Multiplier	Repetitions
+	One or more repetitions
*	Zero or more repetitions

Break up complex patterns

```
api_response <- "payload: 'Adam, 5, 3', headers: 'Auth...'"
```

```
str_match(api_resopnse, pattern = "[A-Za-z]+, \\d+, \\d+")
```

Will match: Adam, 5, 3

```
pattern = glue_collapse(c(
  "name" = "[A-Za-z]+",
  ", ",
  "attempts" = "\\d+",
  ", ",
  "logins" = "\\d+"
))
```

Let's practice!

INTERMEDIATE REGULAR EXPRESSIONS IN R