Week 7: For Loops

Samira & Sophie, 2023

Adapted from Elena & Willa, 2022

Berkeley
UNIVERSITY OF CALIFORNIA

Who are we?

Sophie!
3rd year, Dev
Language and Cognitive Development Lab
sophie_regan@berkeley.edu

Samira (she/they)
4th year, Neuroscience (Cog Neuro)
Jagust & Weiner labs
smaboudian@berkeley.edu

Berkeley
UNIVERSITY OF CALIFORNIA

Today's agenda

- Intros + Warm-up
- For loops slides & demo
- Practice problems

What is a *loop*? What is a ***for*** *loop* for?

Loop - the action of doing something over and over again

For Loop - type of loop that has a pre-specified beginning and end; used when you want to repeat an action a certain number of times

Examples:

- “Set the table for 4 people”
- Send an invitation to everyone on a guest list
- Add one point to the grade of every student in your class

Example: Imagine you're giving instructions to someone who has never set a table before for 4

Do this 4 times:

- Put out placemat
- Put out plate
- Put out napkin
- Place fork on napkin on left side
- Place knife then spoon on right side of the plate
- Put glass on right side of placemat

Guest list example

I’m having a Halloween party. I need to make a guest list and send text messages to everyone to invite them.

I could do a group text, but I figured it would be more personal to send individual messages.

```
guest_list = Willa, Samira, Elena, Sophie
```

That’s it for now. I can always add more people later.

```
guest_list = Willa, Samira, Elena, Sophie
```

Now that I have my list, I'm going to write my message:

```
message = "hi [guest]! I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!"
```

Sweet! Now that I have my guest list and message, I'm ready to start sending text invites **to each person on the guest list**.

To Willa: send(message)
To Samira: send(message)
To Elena: send(message)
… etc.

For each person on the guest list:
send(message)

Goal:

sent

To: **345-943-0876**,
Hi **Willa**! I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!

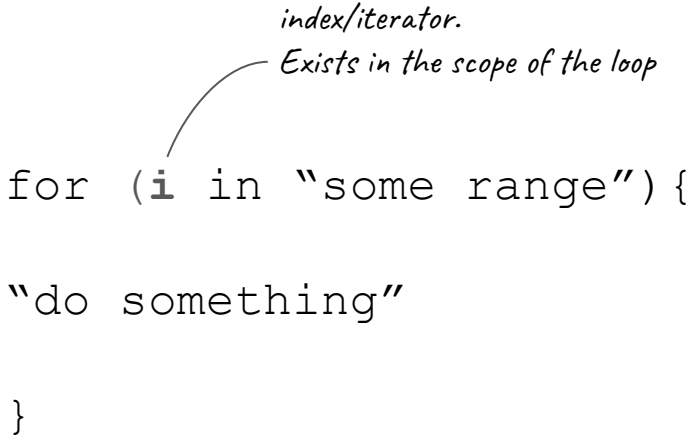
To: **206-738-9942**,
Hi **Elena**! I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!

To: **548-224-0381**,
Hi **Samira**! I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!

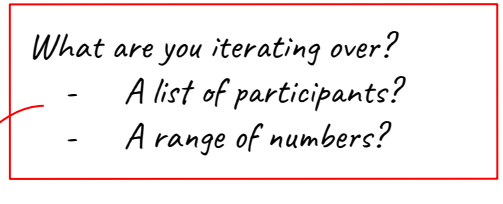
Anatomy of a for loop

```
for (i in "some range"){

"do something"

}
```

Anatomy of a for loop

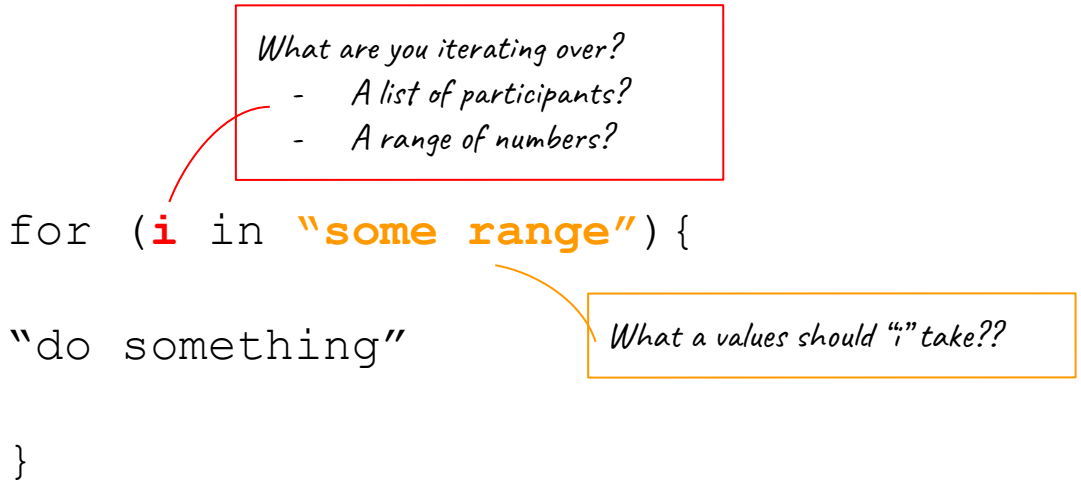


Anatomy of a for loop

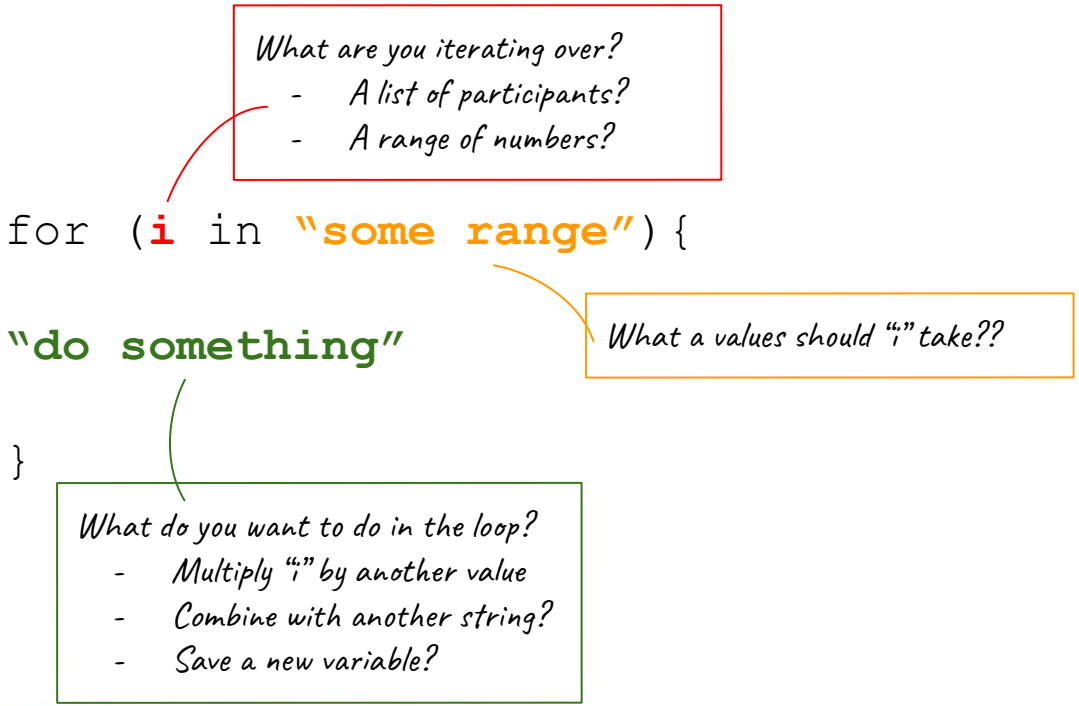



```
for (i in "some range"){

"do something"

}
```

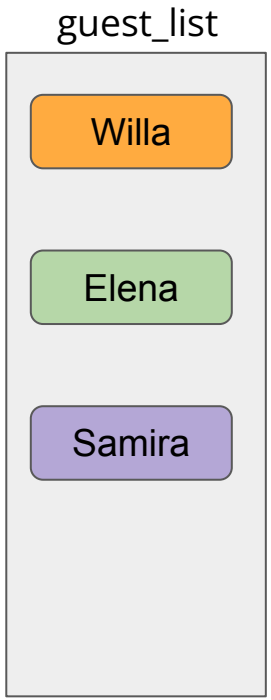
Anatomy of a for loop



Anatomy of a for loop



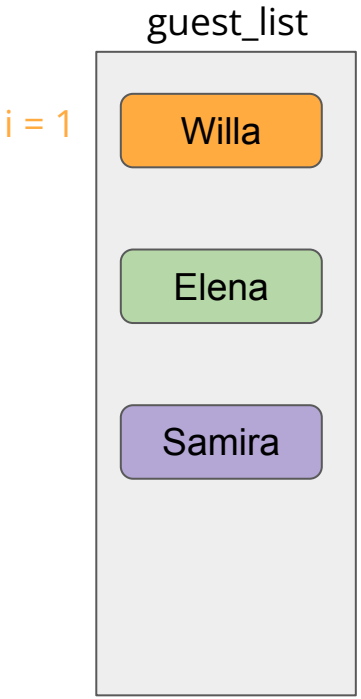
The logic of for loops



length(guest_list) = 3, so loop stops at i = 3
→ generally, you should use `1:length(guest_list)` *here instead of* `1:3`
incase your list length changes (you add/remove guests)

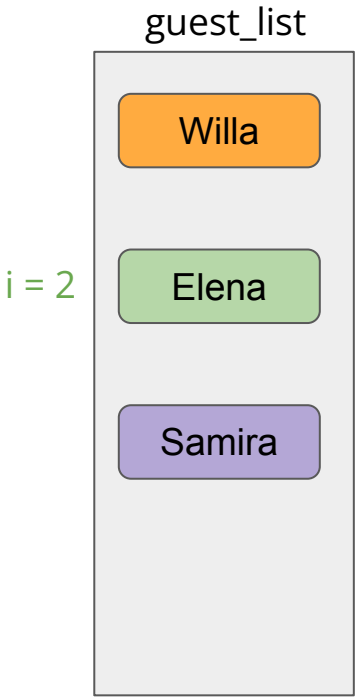
```
for (i in 1:3) {

print(guest_list[i])

}
```

The logic of for loops



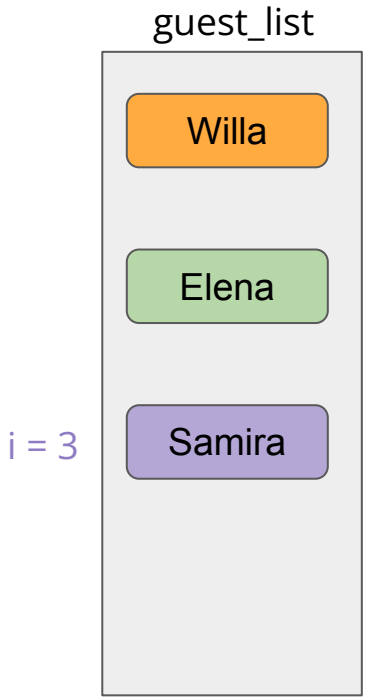

```
for (i in 1:3) {

print(guest_list[i])

}
```

The logic of for loops



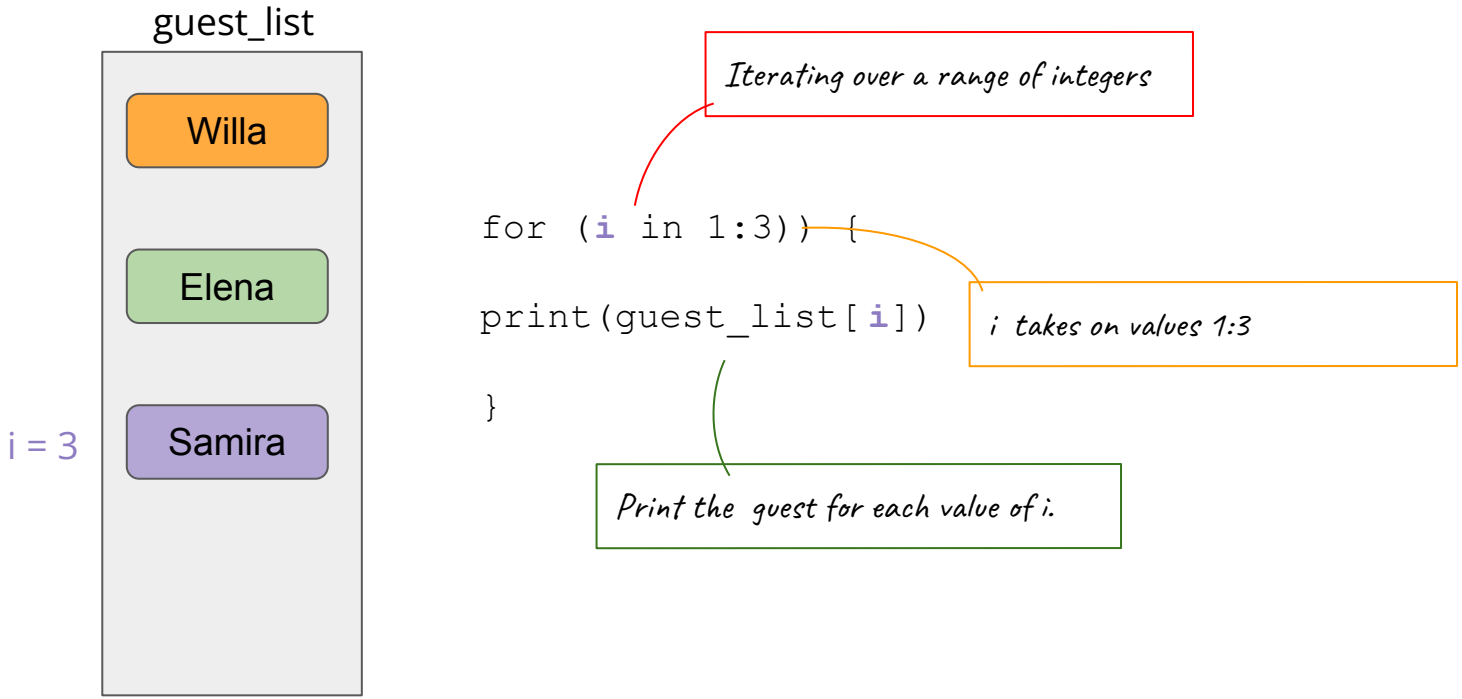

```
for (i in 1:3) {

print(guest_list[i])

}
```

The logic of for loops

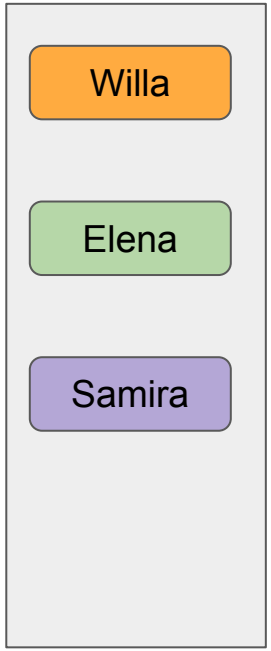



```
for (i in 1:3) {

print(guest_list[i])

}
```

The logic of for loops



Let's use this idea to write a personalized message

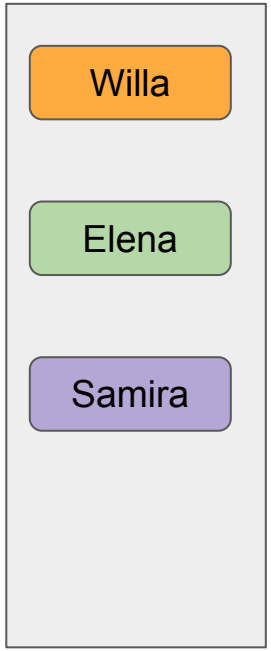


Print a message that says `"hi <guest name>. "I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!"`

What needs to change each time?

What stays the same?

Let's use this idea to write a personalized message

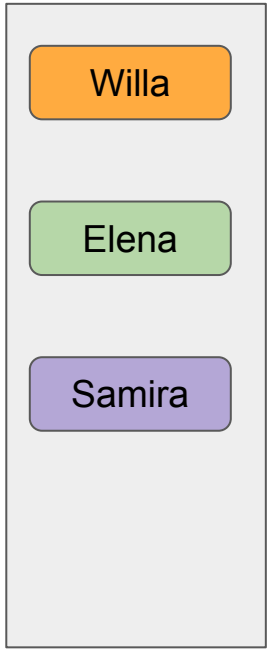


iterate over guest list

Print a message that says "hi **<guest name>**. "I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!"

The message stays the same

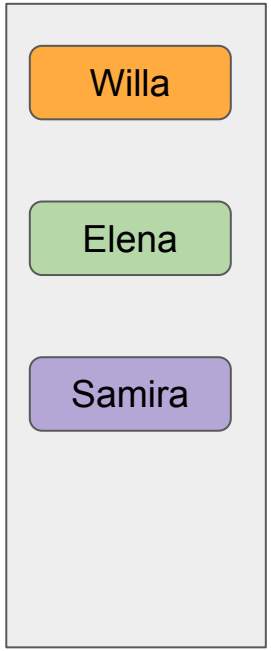
We can use indexing to personalize the message




```
message = "I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!"

for (i in 1:3) {

print(paste0("hi", guest_list[ i], "message")

}
```

We can use indexing to personalize the message

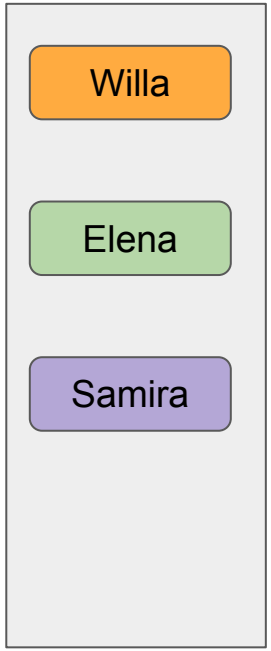
i = 1




```
message = "I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!"

for (i in 1:3) {

print(paste0("hi", guest_list[ i], message")

}
```

Hi **Willa**! I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!

We can use indexing to personalize the message

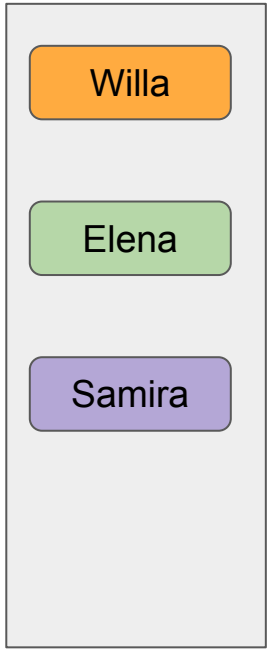


i = 2

```
message = "I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!"

for (i in 1:3) {

print(paste0("hi", guest_list[ i], message")

}
```

Hi **Elena**! I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!

We can use indexing to personalize the message

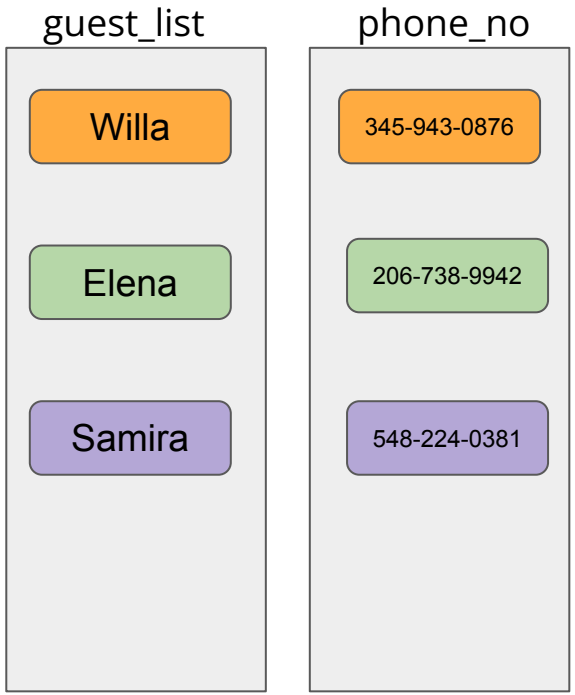


i = 3

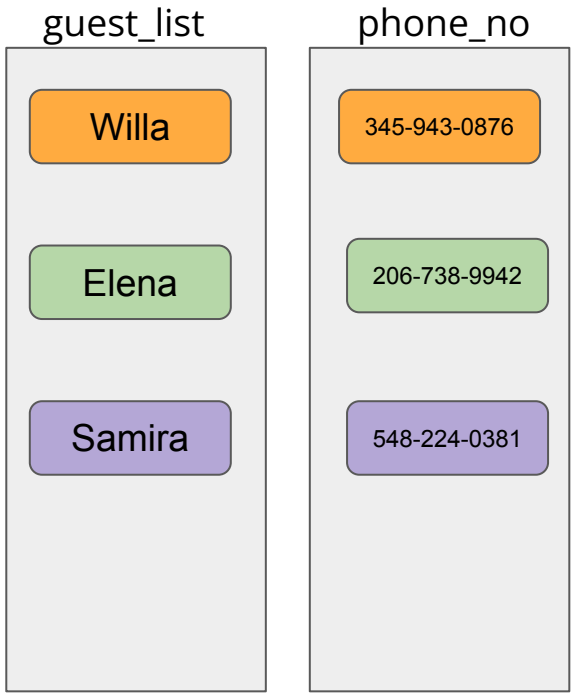
```
message = "I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!"

for (i in 1:3) {

print(paste0("hi", guest_list[ i], message")

}
```

Hi **Samira**! I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!

The concept of *indexing* allows us to include information from more vectors

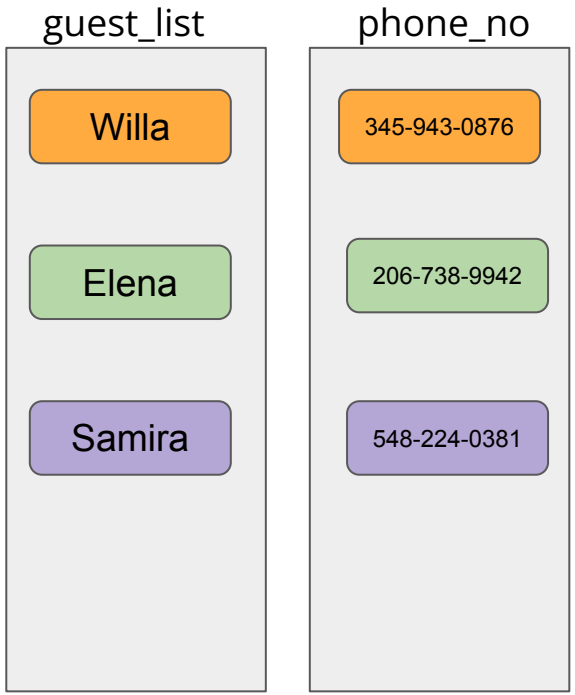


The concept of *indexing* allows us to include information from more vectors



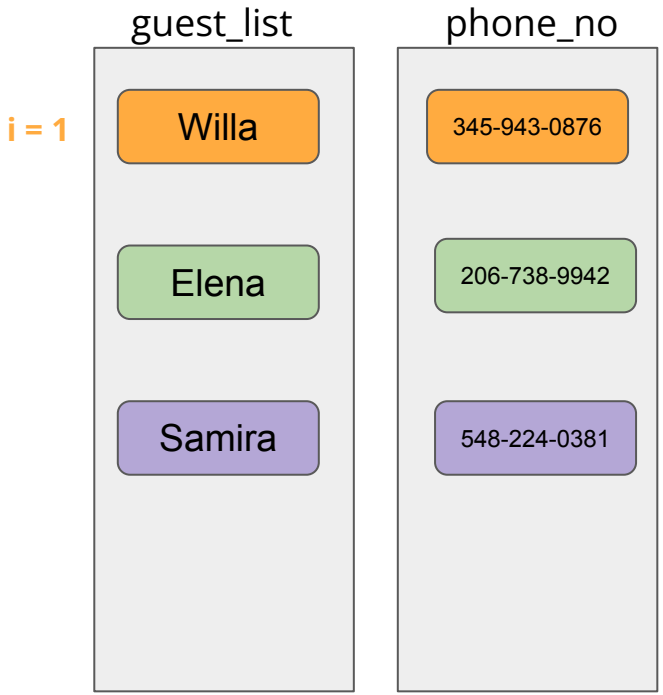
To: **345-943-0876**,
Hi **Willa**! I'm having a Halloween Party on 10/31! The party starts at 9pm. I hope to see you there!

The concept of *indexing* allows us to include information from more vectors




```
for (i in 1:3) {
print(paste0("To:", phone_no[i],
"hi", guest_list[i], message")
}
```

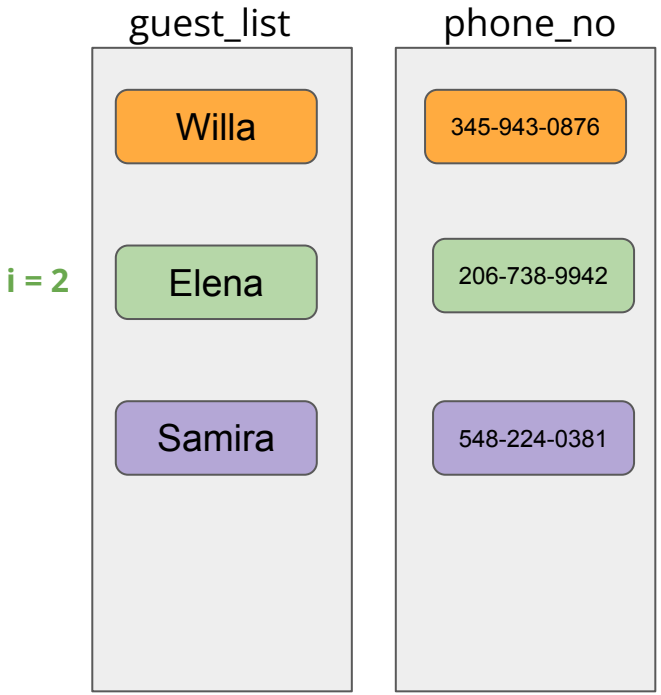
The concept of *indexing* allows us to include information from more vectors




```
for (i in 1:3) {

print(paste0("To:", phone_no[i],

"hi", guest_list[i], message")

}
```

To: **345-943-0876**,
Hi **Willa**! I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!

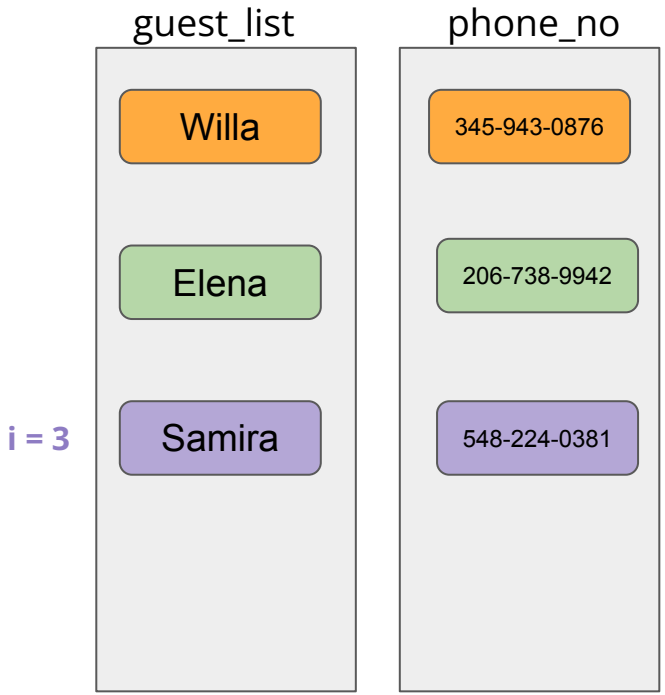
The concept of *indexing* allows us to include information from more vectors




```
for (i in 1:3) {

print(paste0("To:", phone_no[i],

"hi", guest_list[i], message")

}
```

To: **206-738-9942**,
Hi **Elena**! I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!

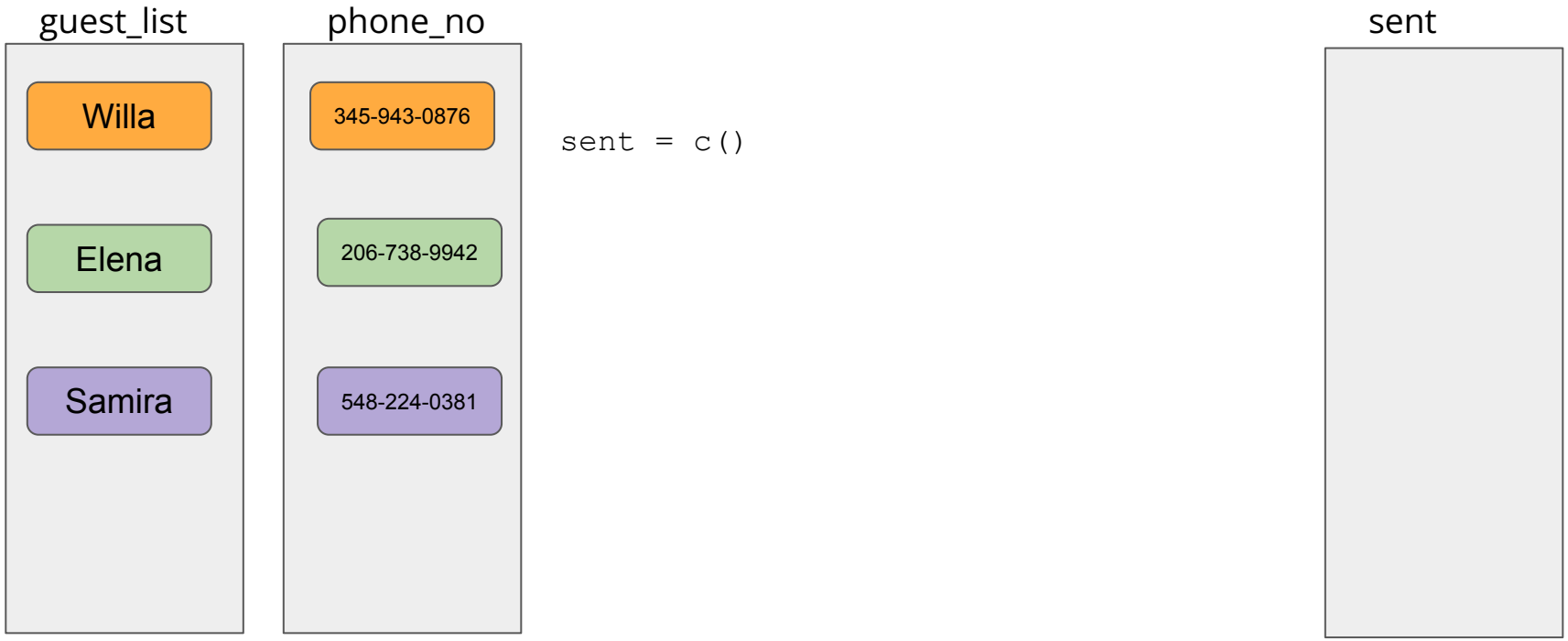
The concept of *indexing* allows us to include information from more vectors



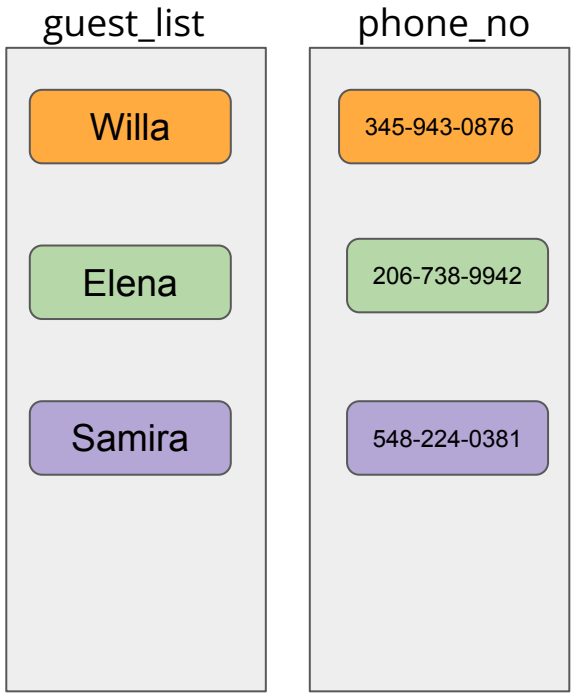

```
for (i in 1:3) {

print(paste0("To:", phone_no[i],

"hi", guest_list[i], message")

}
```

To: **548-224-0381**,
Hi **Samira**! I'm having a Halloween Party on 10/31!
The party starts at 9pm. I hope to see you there!

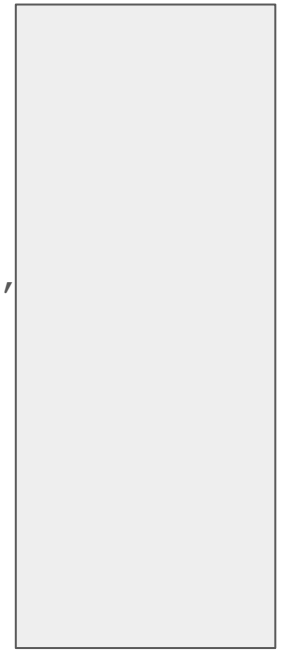
We can also use indexing to save the output to a new vector



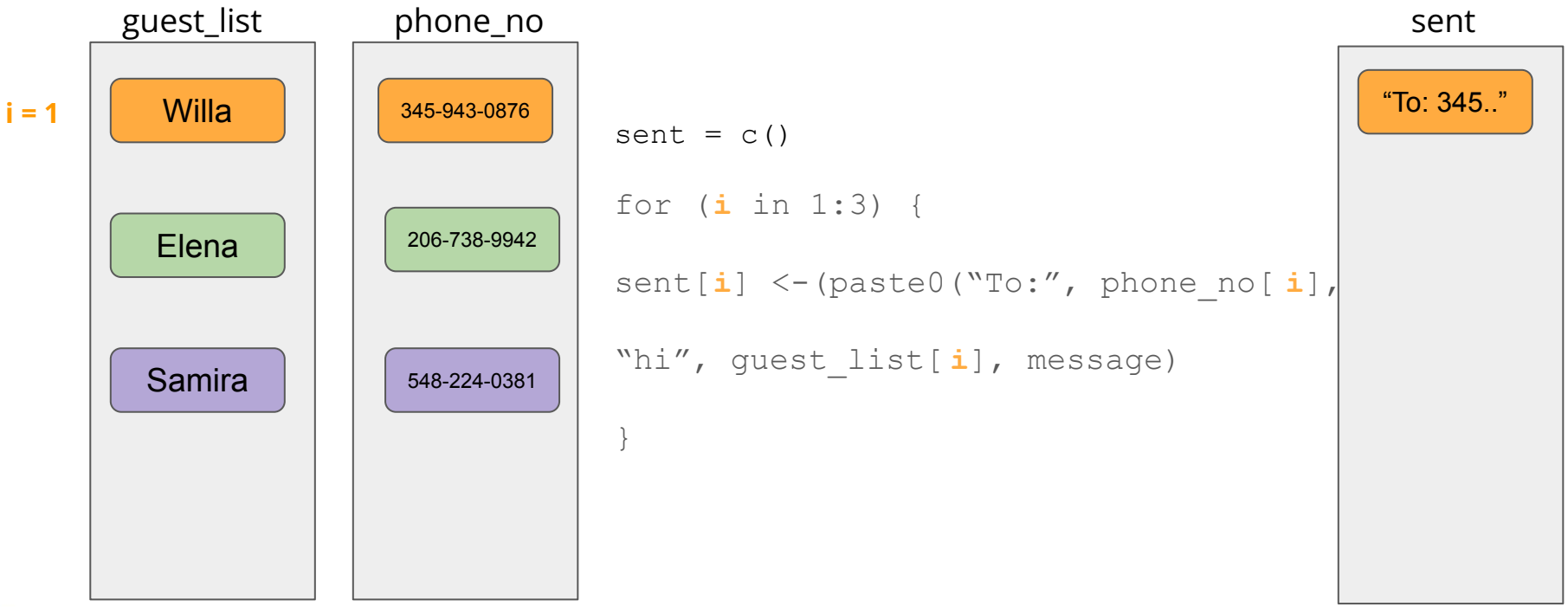
We can also use indexing to save the output to a new vector




```
sent = c()

for (i in 1:3) {

sent[i] <-(paste0("To:", phone_no[ i],

"hi", guest_list[ i], message")

}
```

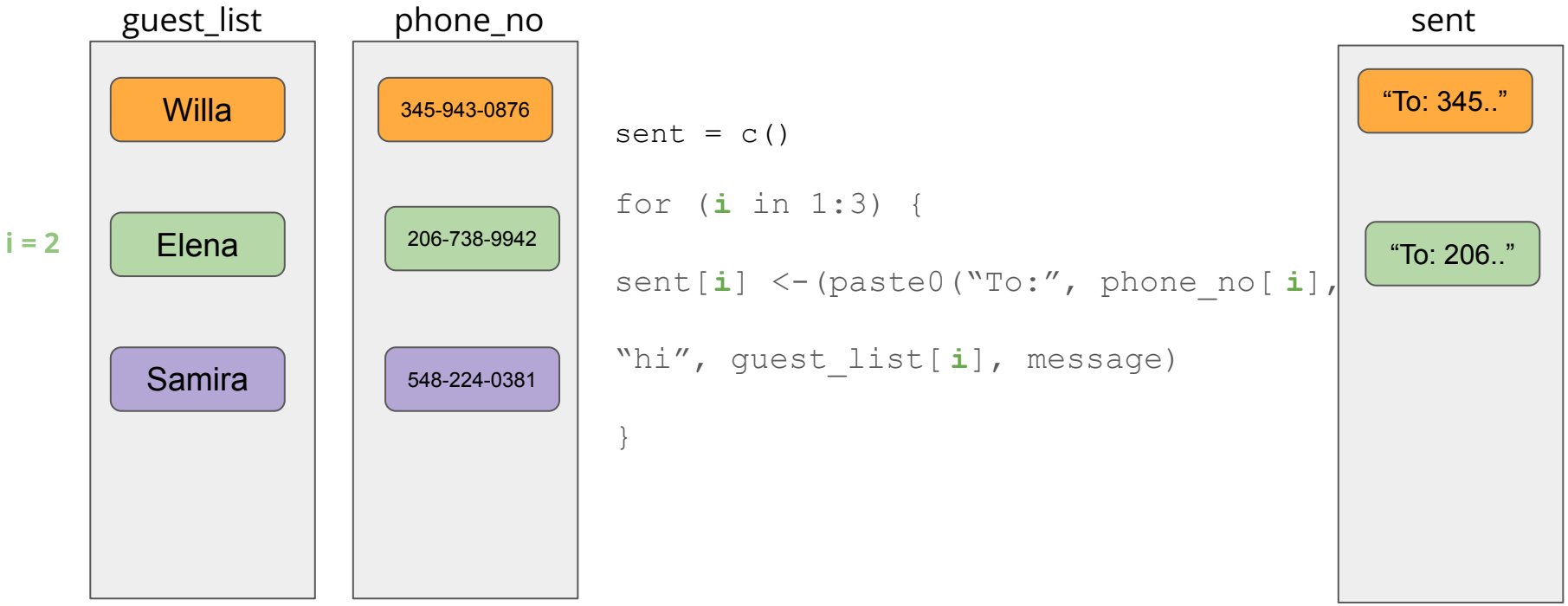


We can also use indexing to save the output to a new vector

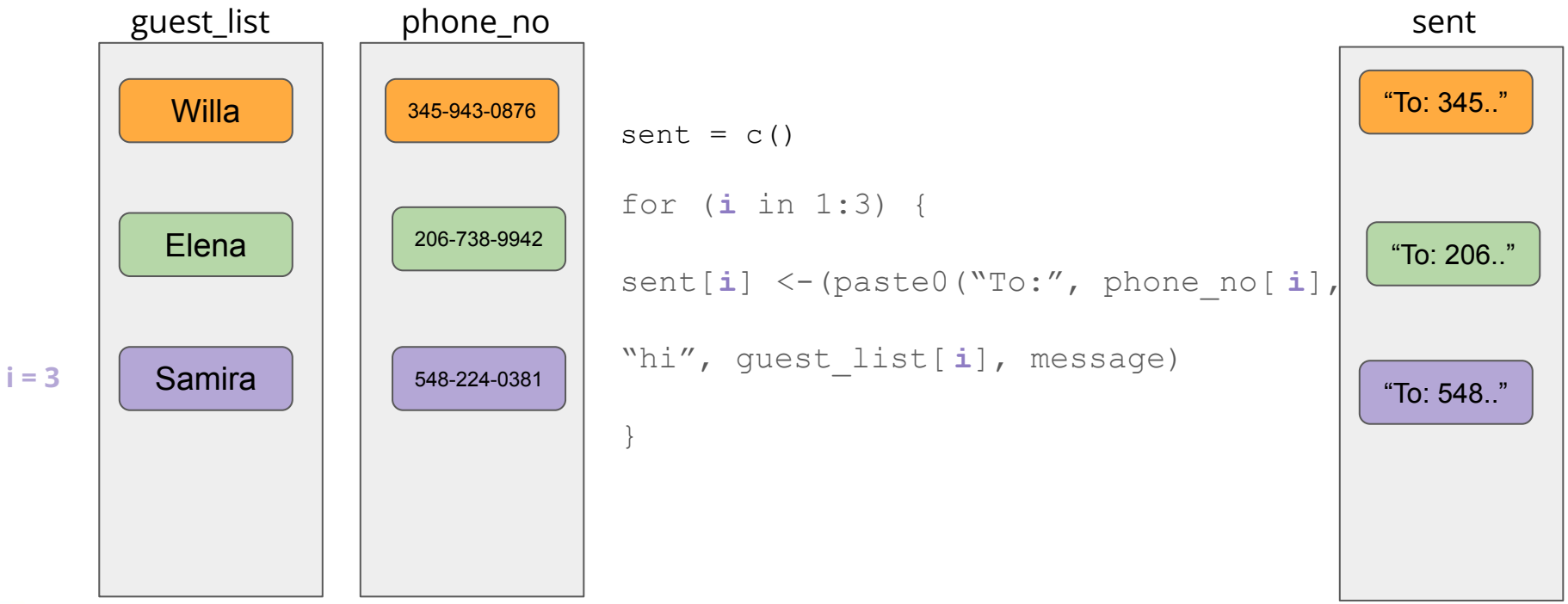



```
sent = c()

for (i in 1:3) {

sent[i] <-(paste0("To:", phone_no[ i],

"hi", guest_list[ i], message)

}
```

We can also use indexing to save the output to a new vector

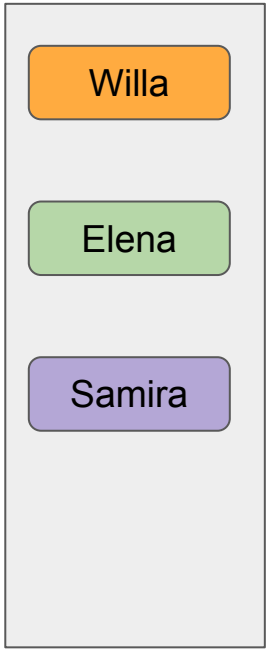


We can also use indexing to save the output to a new vector



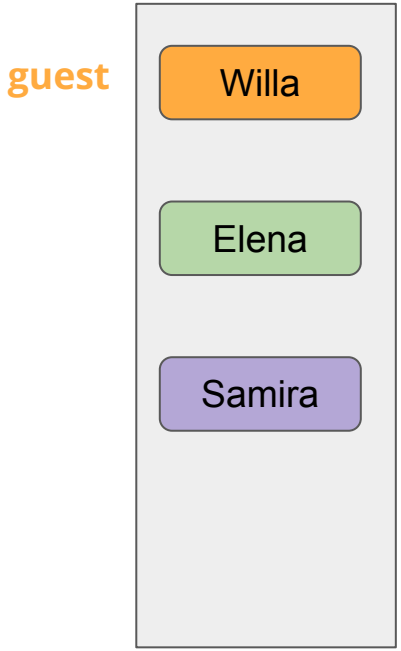
“ i “ is the standard we use for indexing, but you can use anything that makes sense to you.

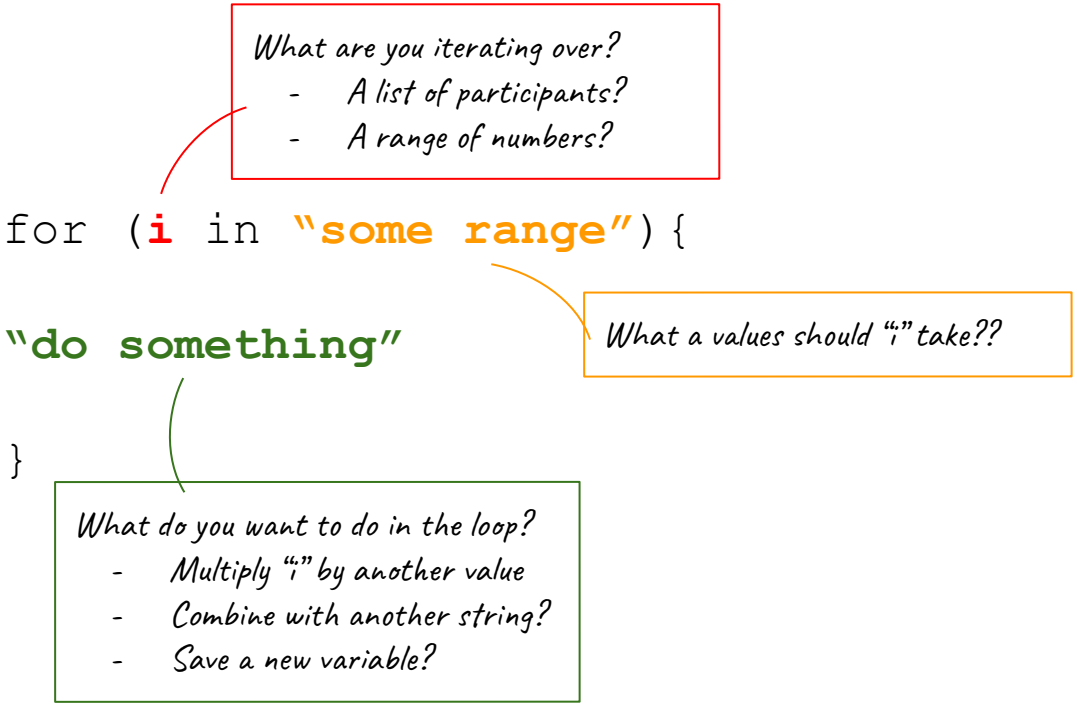
guest_number =1




```
for (guest_num in 1:3) (

print(guest_list[guest_num])

)
```

When we only have one vector we can use an even simpler syntax: loop through the elements instead of the index




```
for (guest in guest_list) (

print(guest)

)
```

Anatomy of a for loop