How to Use the Server 2

Assignment Feedback

- Everyone should have it as comments on their commits on GitHub (and you should have gotten emails about it)
- There should be answer keys posted for all the assignments so far
- Make sure to answer all the questions
  - I'd rather see you try something than leave it blank
  - Please ask about previous assignments if you still don't know how to answer a question after getting feedback
- You can ask questions either via email or Slack outside of office hours
  - I won't answer right away
  - If you send it outside of normal working hours, I probably won't answer until the next morning

Miscellaneous

- To change your password, type `passwd` and you'll be prompted to enter a new password
- To close the connection, type `exit` or simply quit Terminal/PuTTy
- To cancel a command `Ctrl+C`
- Copying and pasting
  - Mac = exactly the same as normal ⌘C and ⌘V
  - Windows = highlight to copy, right click to paste

Input and Output (I/O)

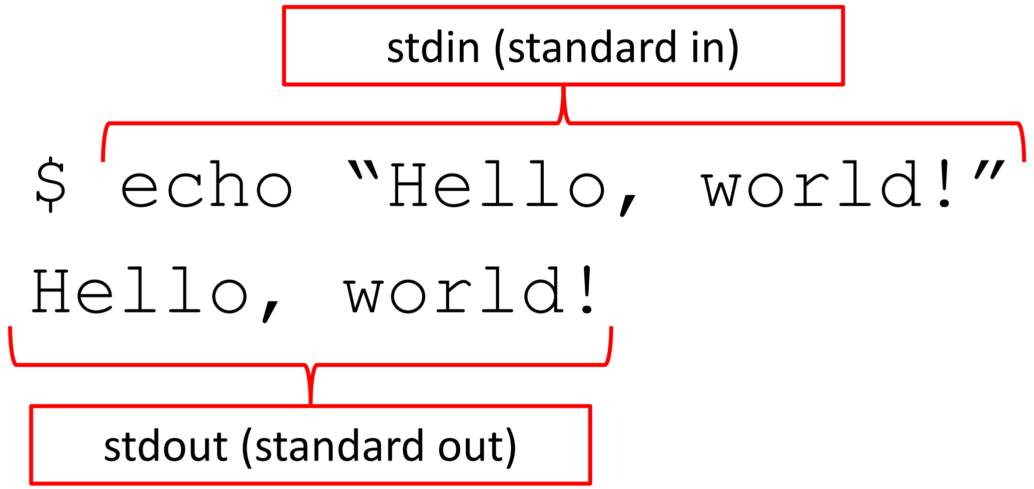
Input and Output Definitions

- Standard input (stdin) – information inputted into the terminal through the keyboard or input device

$$\texttt{\$ }\overbrace{\texttt{echo “Hello, world!”}}^{\boxed{\text{stdin (standard in)}}}$$

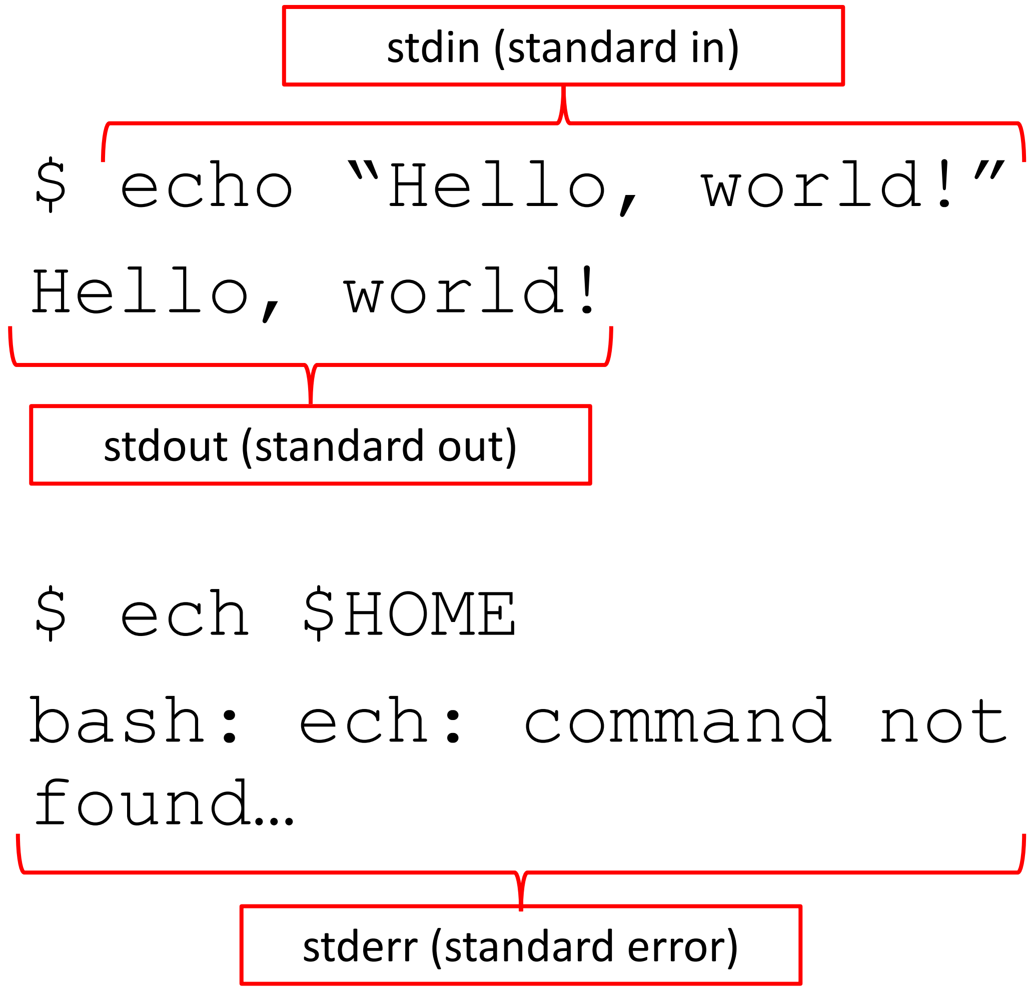
Input and Output Definitions

- Standard input (stdin) – information inputted into the terminal through the keyboard or input device
- Standard output (stdout) – information outputted after the process is run


```
$ echo “Hello, world!”
Hello, world!
```

Input and Output Definitions

- Standard input (stdin) – information inputted into the terminal through the keyboard or input device
- Standard output (stdout) – information outputted after the process is run
- Standard error (stderr) – an error message outputted by a failed process



```
$ echo “Hello, world!”
Hello, world!
```

```
$ ech $HOME
bash: ech: command not
found…
```

I/O Operators

- > = Redirect stdout to a file
- >> = append
- | = pipe

```
$ echo “Hello, world!” > hello.txt
$ cat hello.txt
Hello, world!
$ echo “This is how to use a Linux
server.” >> hello.txt
$ cat alphabet.txt | head
A is for aardvark
B is for bumblebee
C is for chihuahua
D is for donkey
E is for elephant shrew
F is for flamingo
G is for Galapagos tortoise
H is for hippopotamus
I is for iguana
J is for jackal
```

Loops in Bash

What’s a loop?

What’s a loop?

A loop is a piece of code that repeatedly executes a command for a certain condition until that condition is no longer true.

What’s a loop?

A loop is a piece of code that repeatedly executes a command for a certain condition until that condition is no longer true.

```
for i in *.fastq.gz;
   do echo $i;
done
```

What’s a loop?

A loop is a piece of code that repeatedly executes a command for a certain condition until that condition is no longer true.

```
for i in *.fastq.gz;
   do echo $i;
done
```

condition

What’s a loop?

A loop is a piece of code that repeatedly executes a command for a certain condition until that condition is no longer true.

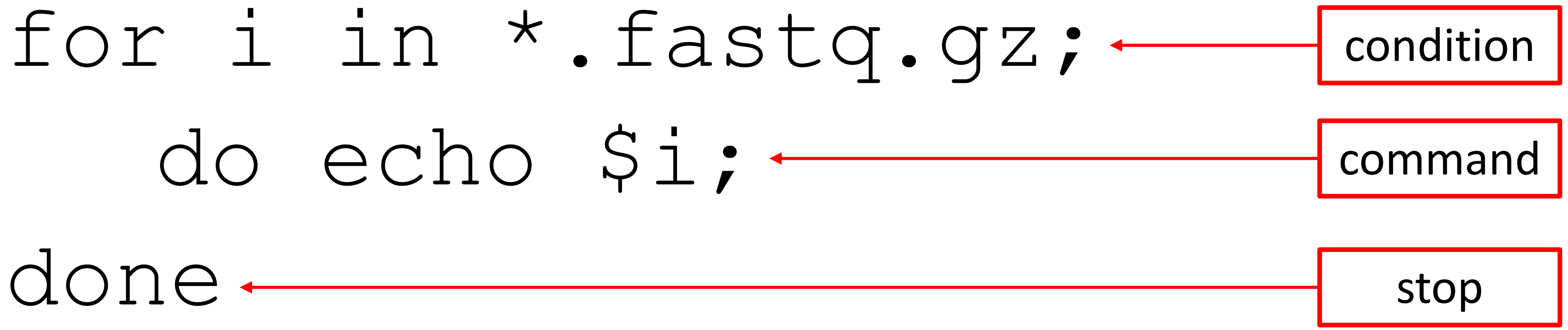
```
for i in *.fastq.gz;
    do echo $i;
done
```

condition

command

What’s a loop?

A loop is a piece of code that repeatedly executes a command for a certain condition until that condition is no longer true.



```
for i in *.fastq.gz;
    do echo $i;
done
```

How do you write a loop in bash?

How do you write a loop in bash?

```
for i in *.fastq.gz;
   do echo $i;
done
```

How do you write a loop in bash?

starting a loop

```
for i in *.fastq.gz;
    do echo $i;
done
```

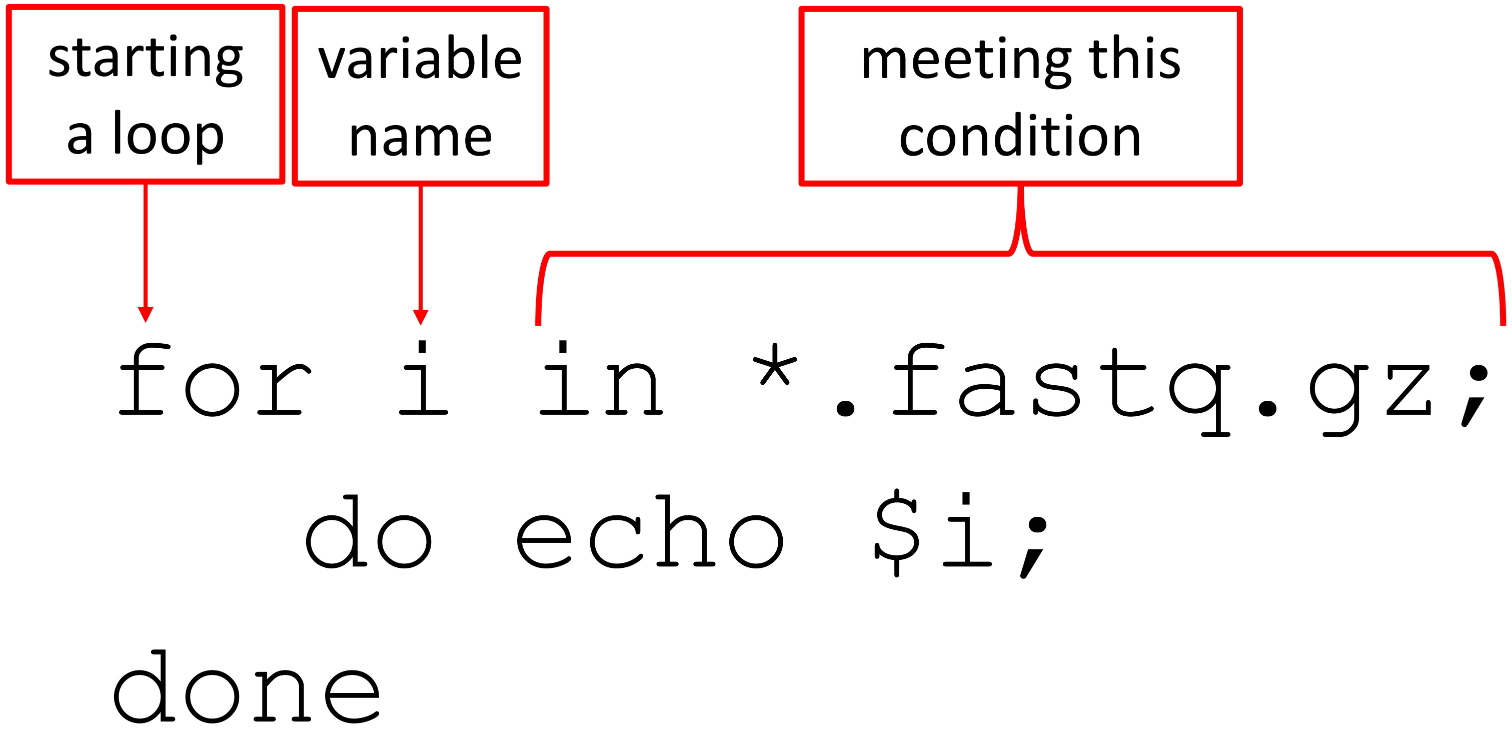
How do you write a loop in bash?

starting a loop

variable name

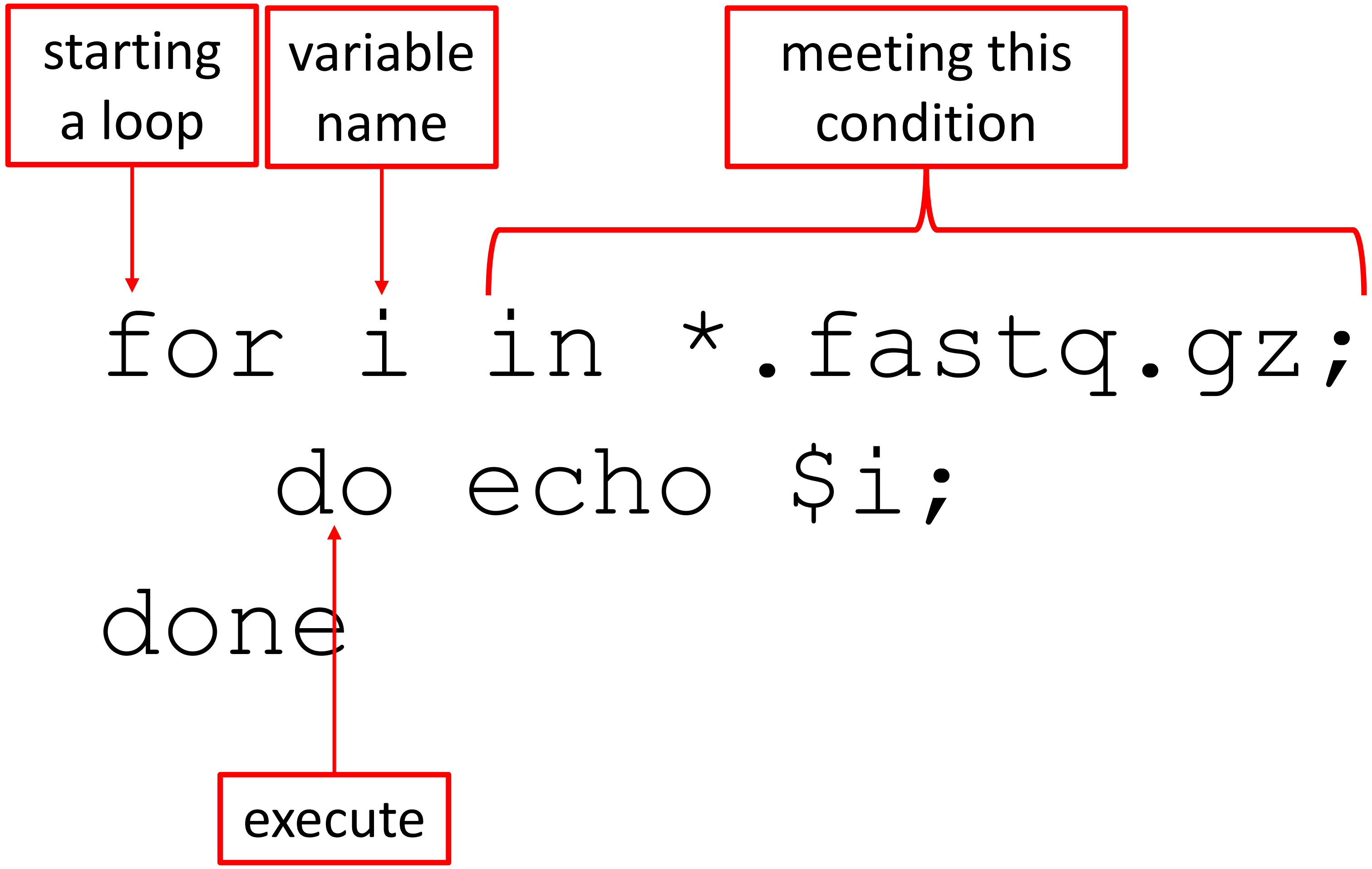
```
for i in *.fastq.gz;
   do echo $i;
done
```

How do you write a loop in bash?



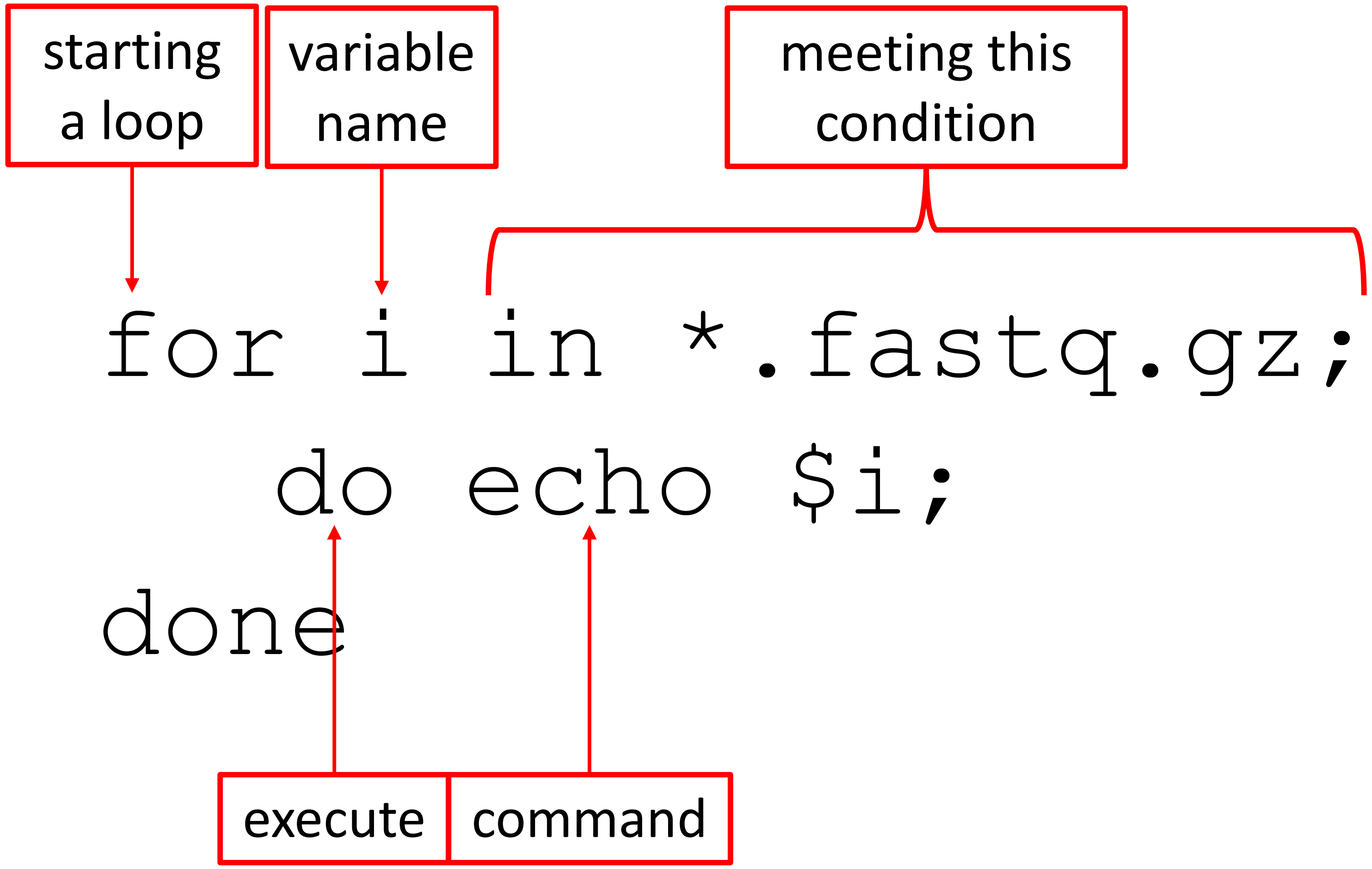
```
for i in *.fastq.gz;
   do echo $i;
done
```

How do you write a loop in bash?



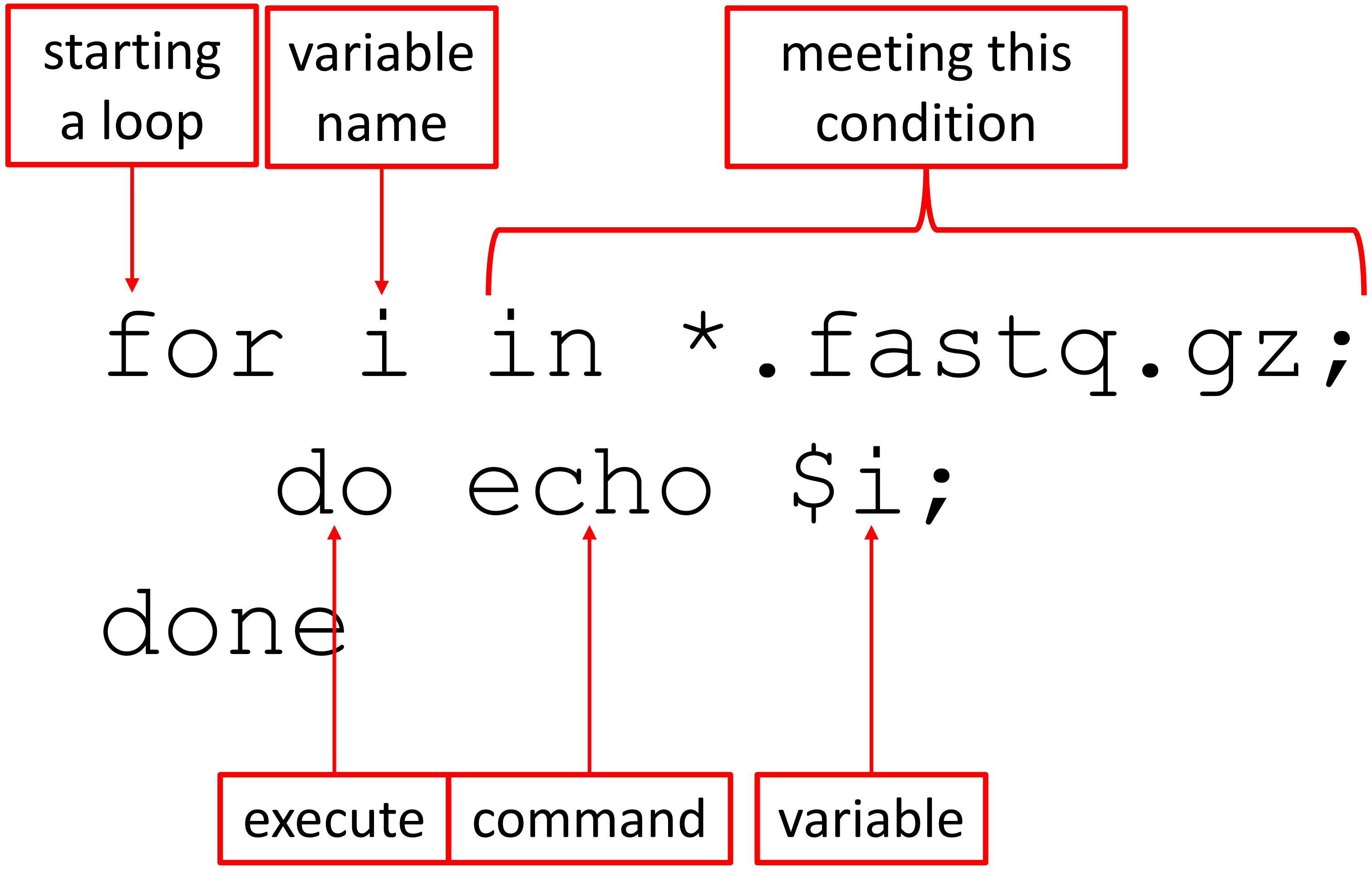
```
for i in *.fastq.gz;
    do echo $i;
done
```

How do you write a loop in bash?



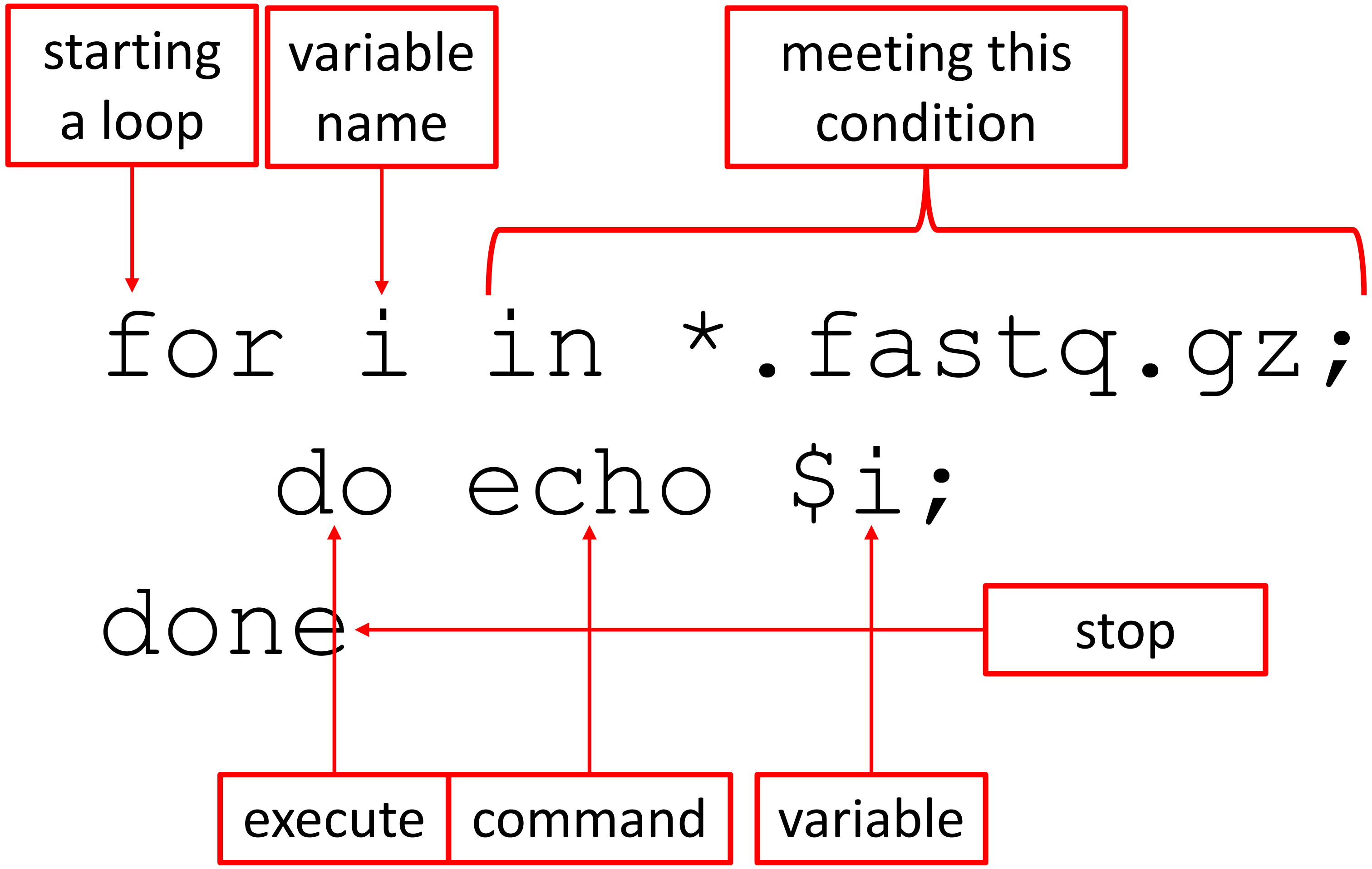
```
for i in *.fastq.gz;
    do echo $i;
done
```

How do you write a loop in bash?



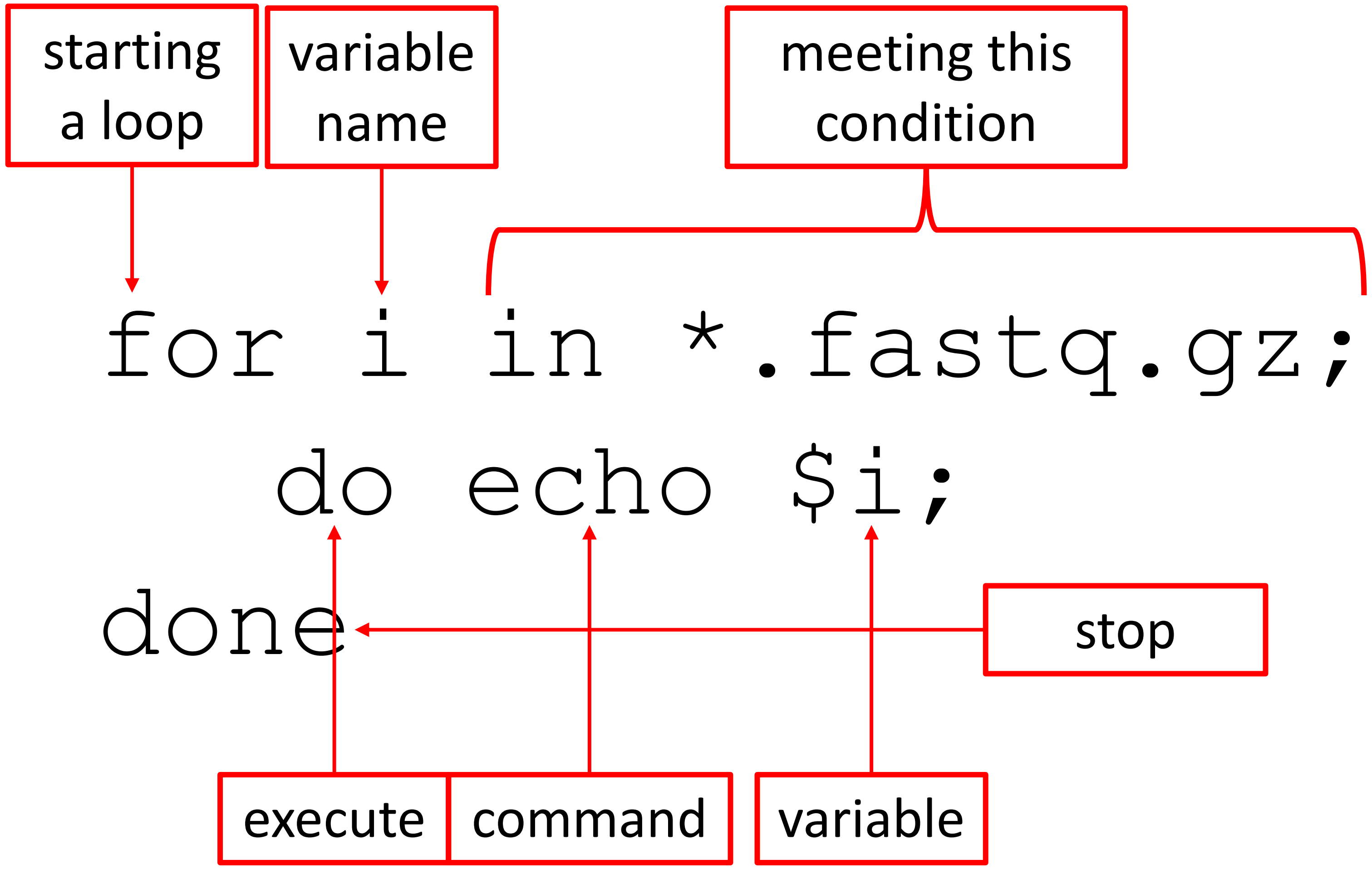
```
for i in *.fastq.gz;
   do echo $i;
done
```

How do you write a loop in bash?



```
for i in *.fastq.gz;
    do echo $i;
done
```

How do you write a loop in bash?



```
for i in *.fastq.gz;
    do echo $i;
done
```

Now go try this loop on the server

Running Processes

Running Processes in `tmux`

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
    - tmux new–session –s session1

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed

```
[kkeith]$ tmux new-session -s
session1
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1

```
[kkeith]$









barbarbarbarbarbarbarbarbarbarbarbar
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1

```
[kkeith]$ pwd
/home/kkeith/data/practice_directory
```

```
barbarbarbarbarbarbarbarbarbarbarbar
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1
  - Ctrl + B, then d to detach

```
[kkeith]$ pwd
/home/kkeith/data/practice_directory
```

1. Hit Ctrl + B
2. Hit d

```
barbarbarbarbarbarbarbarbarbarbarbar
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1
  - Ctrl + B, then d to detach

```
[kkeith]$ tmux new–session -s session1
[detached (from session session1)]
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1
  - Ctrl + B, then d to detach
  - tmux ls

```
[kkeith]$ tmux new–session –s session1
[detached (from session session1)]
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
    - tmux new–session –s session1
    - Ctrl + B, then d to detach
    - tmux ls
    - tmux attach –t session1

```
[kkeith]$ tmux new-session -s session1
[detached (from session session1)]
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
[kkeith]$ tmux attach -t session1
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1
  - Ctrl + B, then d to detach
  - tmux ls
  - tmux attach –t session1

```
[kkeith]$ pwd
/home/kkeith/data/practice_directory
```

```
barbarbarbarbarbarbarbarbarbarbarbar
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
    - tmux new–session –s session1
    - Ctrl + B, then d to detach
    - tmux ls
    - tmux attach –t session1
    - tmux kill-session –t session1

```
[kkeith]$ tmux new-session -s session1
[detached (from session session1)]
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
[kkeith]$ tmux attach -t session1
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that’s running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1
  - Ctrl + B, then d to detach
  - tmux ls
  - tmux attach –t session1
  - tmux kill-session –t session1

```
[kkeith]$ tmux new-session –s session1
[detached (from session session1)]
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
[kkeith]$ tmux attach –t session1
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
  - tmux new–session –s session1
  - Ctrl + B, then d to detach
  - tmux ls
  - tmux attach –t session1
  - tmux kill-session –t session1

```
[kkeith]$ tmux new-session -s session1
[detached (from session session1)]
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
[kkeith]$ tmux attach -t session1
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
[kkeith]$ tmux kill-session -t session1
```

Running Processes in the Background

- **process** = a program in execution
- When you log out of the server, anything that's running is killed
- **tmux** (terminal multiplexer) = program that allows you to create extra, persistent windows
    - tmux new–session –s session1
    - Ctrl + B, then d to detach
    - tmux ls
    - tmux attach –t session1
    - tmux kill-session –t session1

```
[kkeith]$ tmux new–session –s session1
[detached (from session session1)]
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
[kkeith]$ tmux attach –t session1
[kkeith]$ tmux ls
session1: 1 windows (created Mon Nov 18 16:42:27
2019)
[kkeith]$ tmux kill-session –t session1

[kkeith]$ tmux ls

no server running on /private/tmp/tmux-
669394417/default
```