Informatics on High-throughput Sequencing Data

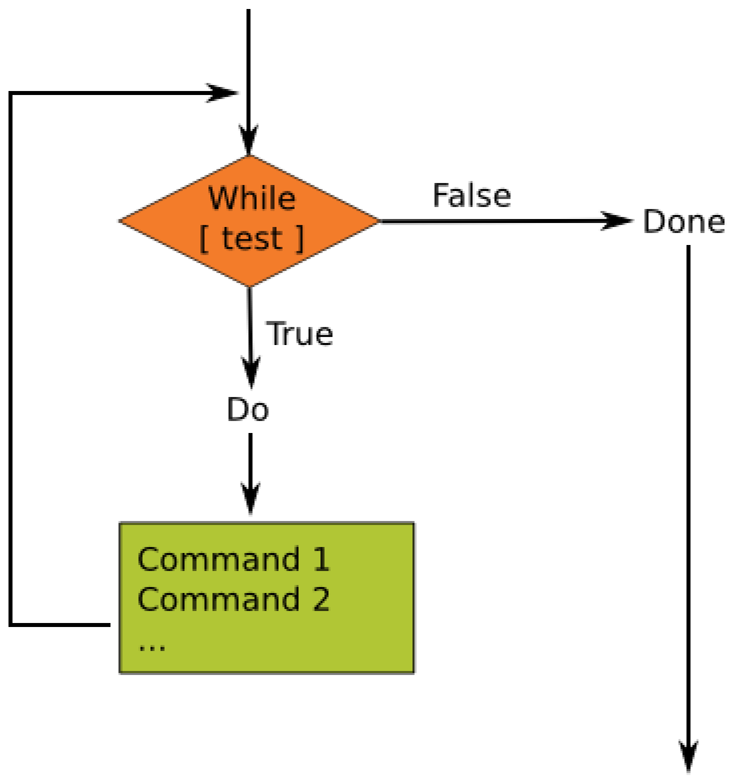
(Summer Course 2020)

Day 9

Agenda

- Shell Scripting
- Variables
- User Inputs
- Arithmetic
- Functions
- IF statements
- Loops

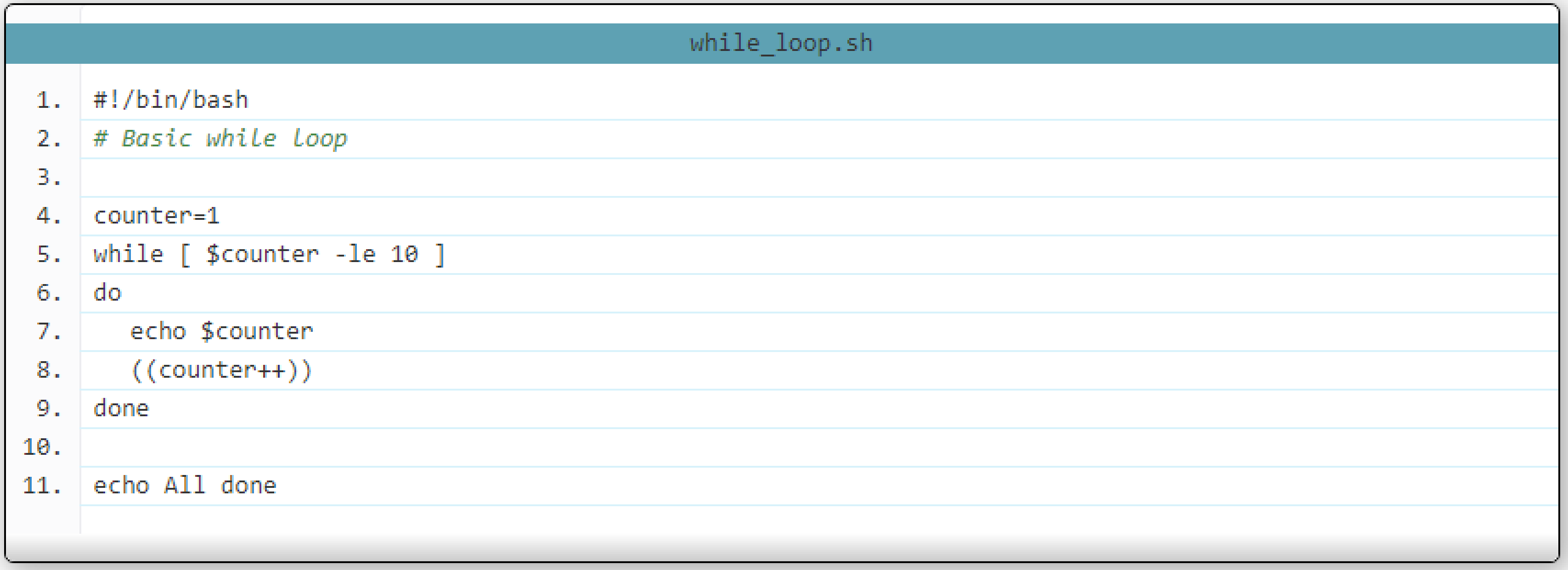
While Loops



```
while [ <some test> ]
do
    <commands>
done
```

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

While Loops

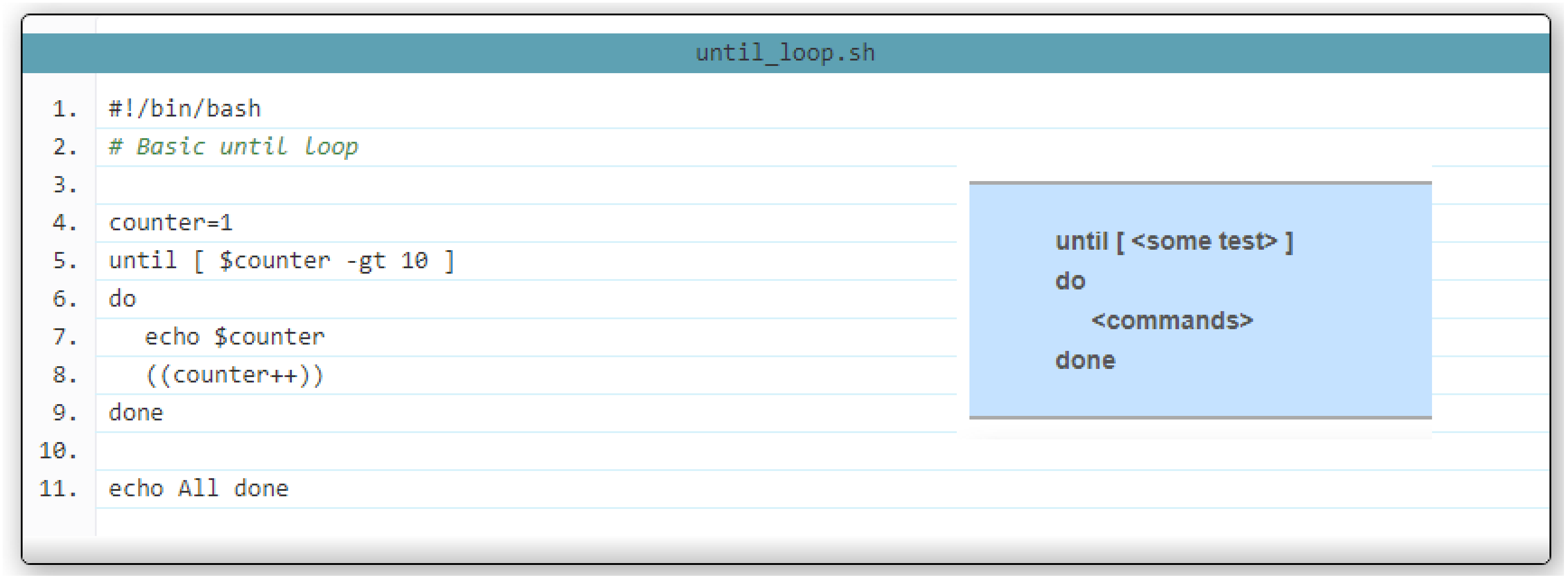


while_loop.sh

```
#!/bin/bash
# Basic while loop

counter=1
while [ $counter -le 10 ]
do
   echo $counter
   ((counter++))
done

echo All done
```

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

Until Loops

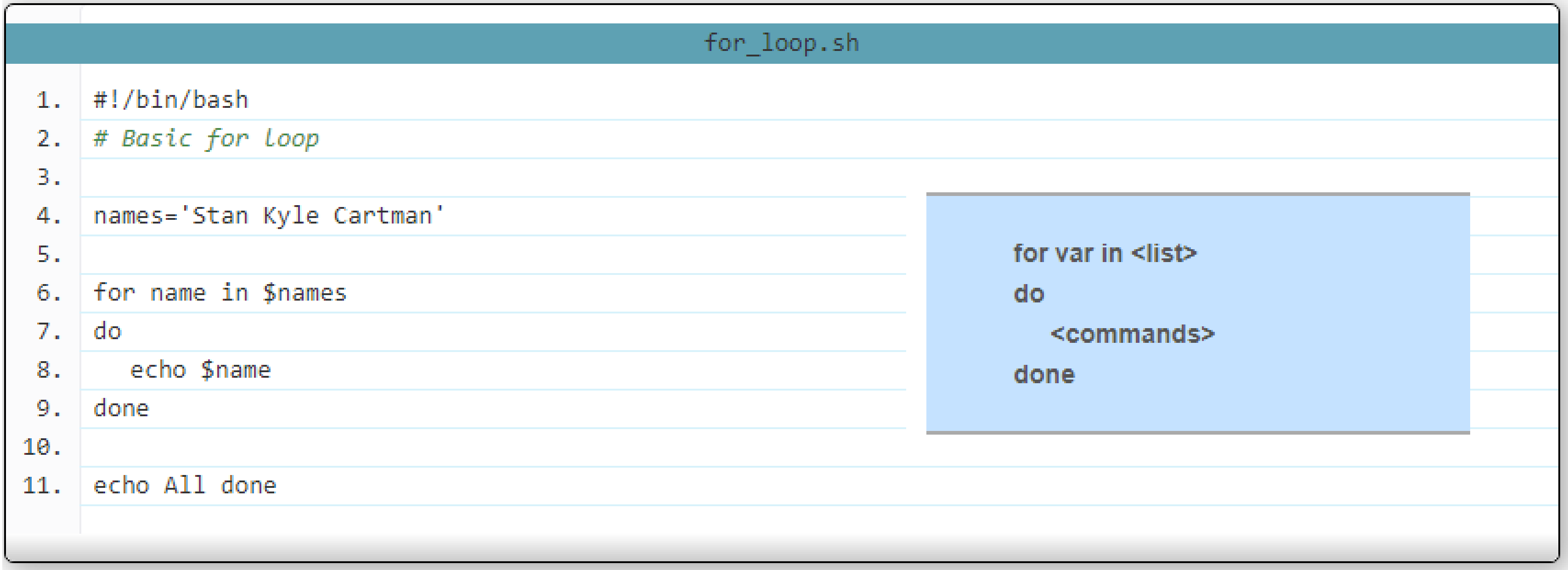

until_loop.sh

```
#!/bin/bash
# Basic until loop

counter=1
until [ $counter -gt 10 ]
do
   echo $counter
   ((counter++))
done

echo All done
```

```
until [ <some test> ]
do
    <commands>
done
```

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

For Loops



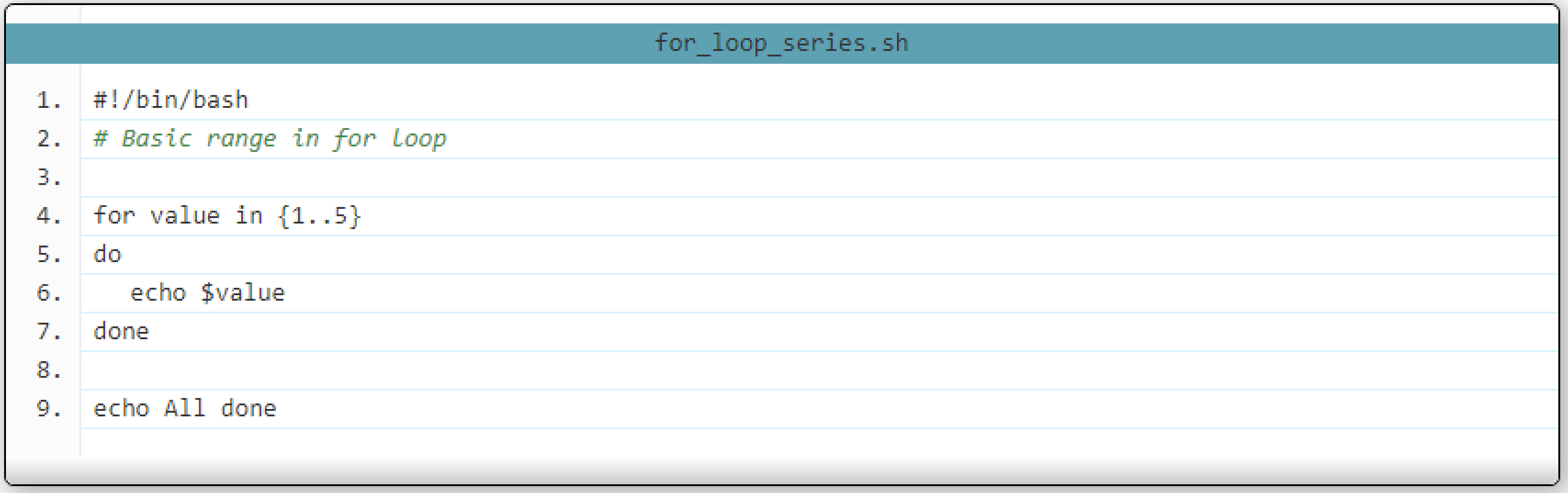
for_loop.sh

```
#!/bin/bash
# Basic for loop

names='Stan Kyle Cartman'

for name in $names
do
   echo $name
done

echo All done
```

```
for var in <list>
do
    <commands>
done
```

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

Ranges

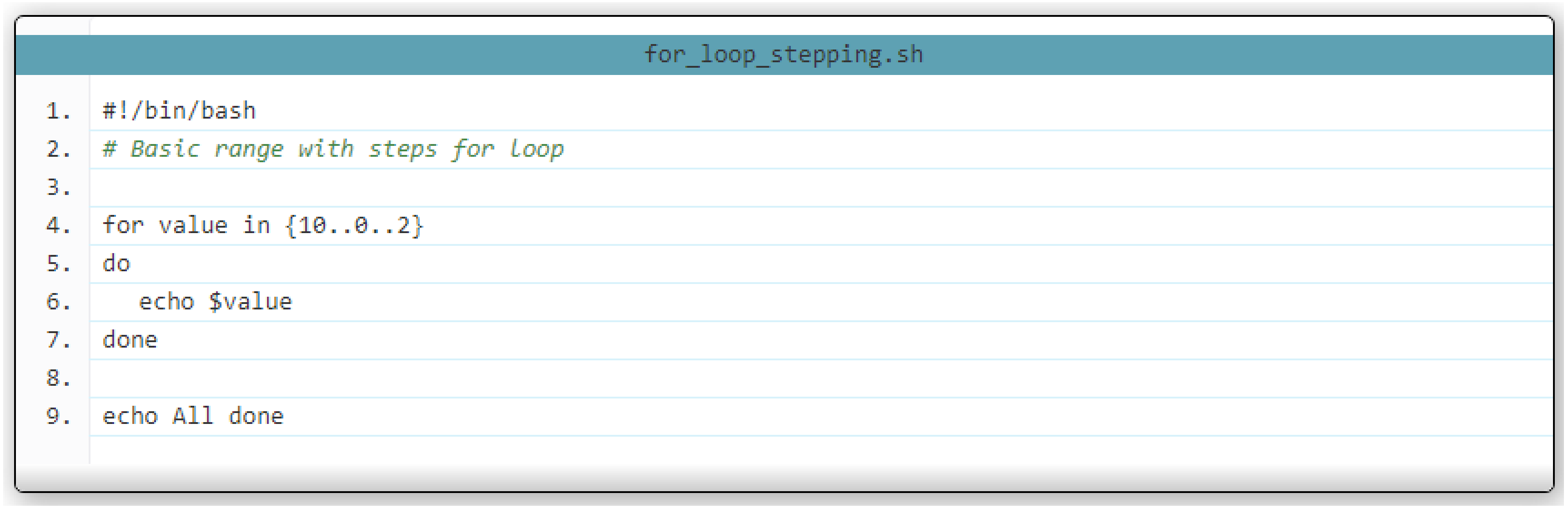


for_loop_series.sh

```bash
#!/bin/bash
# Basic range in for loop

for value in {1..5}
do
   echo $value
done

echo All done
```

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

Ranges


for_loop_stepping.sh

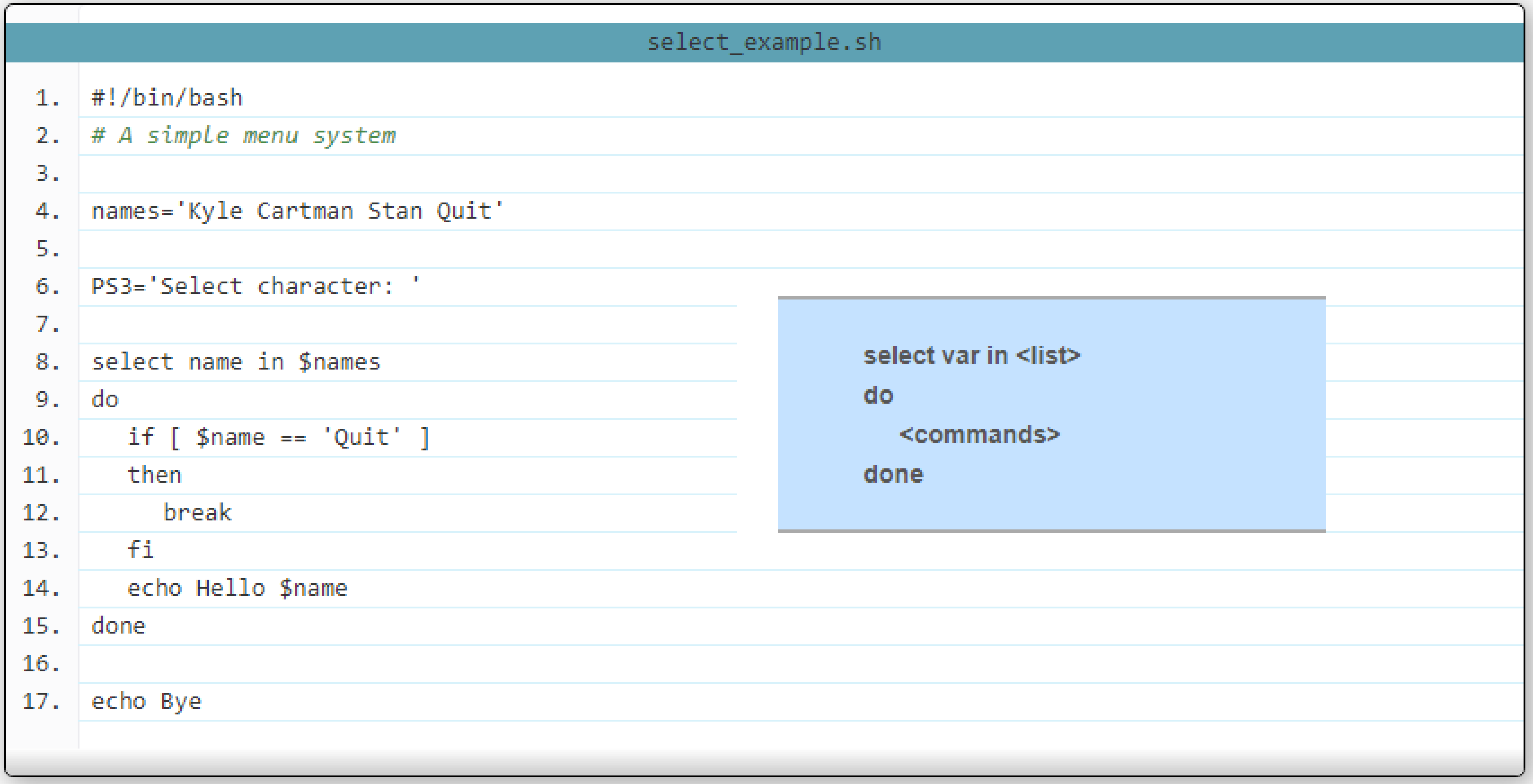
```
#!/bin/bash
# Basic range with steps for loop

for value in {10..0..2}
do
   echo $value
done

echo All done
```

- ❑ The break statement tells Bash to leave the loop straight away.
- ❑ The continue statement tells Bash to stop running through this iteration of the loop and begin the next iteration.

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

Select

- The select mechanism allows you to create a simple menu system.



select_example.sh

```bash
#!/bin/bash
# A simple menu system

names='Kyle Cartman Stan Quit'

PS3='Select character: '

select name in $names
do
   if [ $name == 'Quit' ]
   then
      break
   fi
   echo Hello $name
done

echo Bye
```

```
select var in <list>
do
    <commands>
done
```

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

Functions

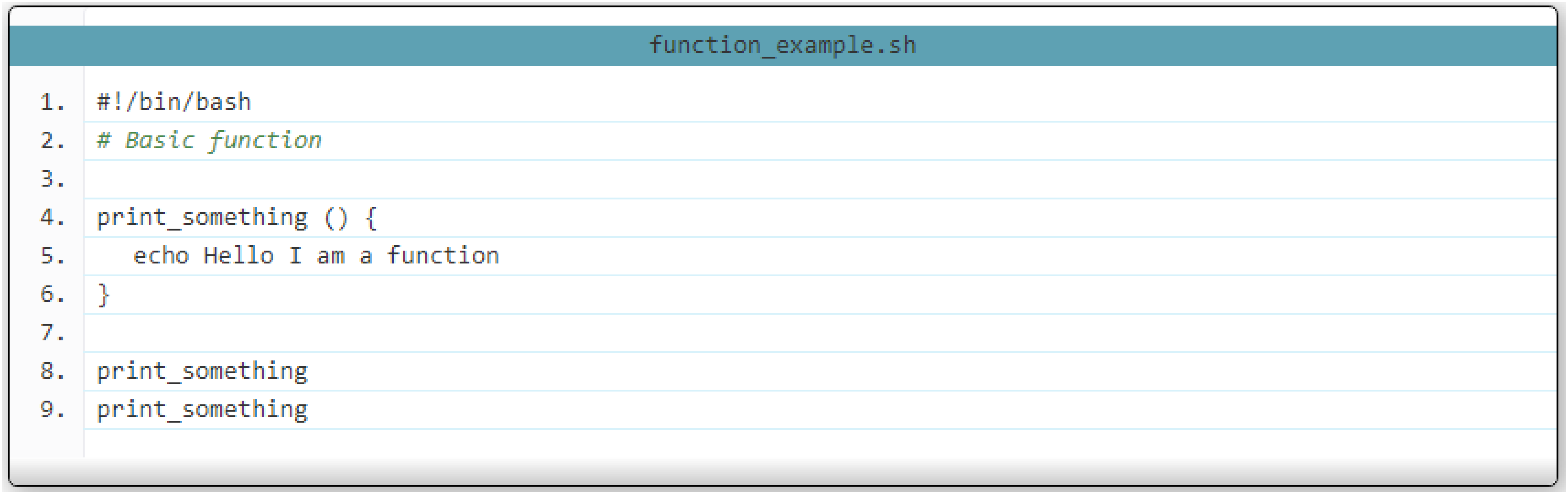
- Either of the below methods of specifying a function is valid.
- In other programming languages it is common to have arguments passed to the function listed inside the brackets ().
- In Bash they are there only for decoration and you never put anything inside them.
- The function definition (the actual function itself) must appear in the script before any calls to the function.

```
function_name () {
    <commands>
}
```

```
function function_name {
    <commands>
}
```

https://ryanstutorials.net/bash-scripting-tutorial/bash-functions.php

Functions

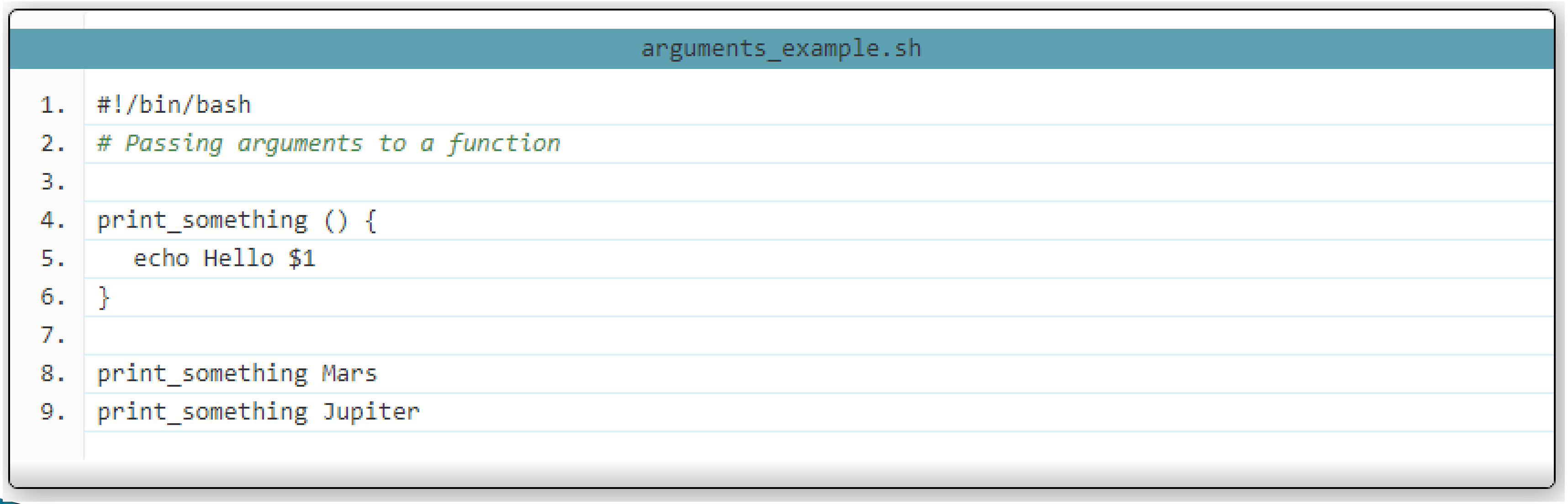

function_example.sh

```bash
#!/bin/bash
# Basic function

print_something () {
   echo Hello I am a function
}

print_something
print_something
```

https://ryanstutorials.net/bash-scripting-tutorial/bash-functions.php

Functions

- **We may send data to the function in a similar way to passing command line arguments to a script.**
- **We supply the arguments directly after the function name. Within the function they are accessible as $1, $2, etc.**



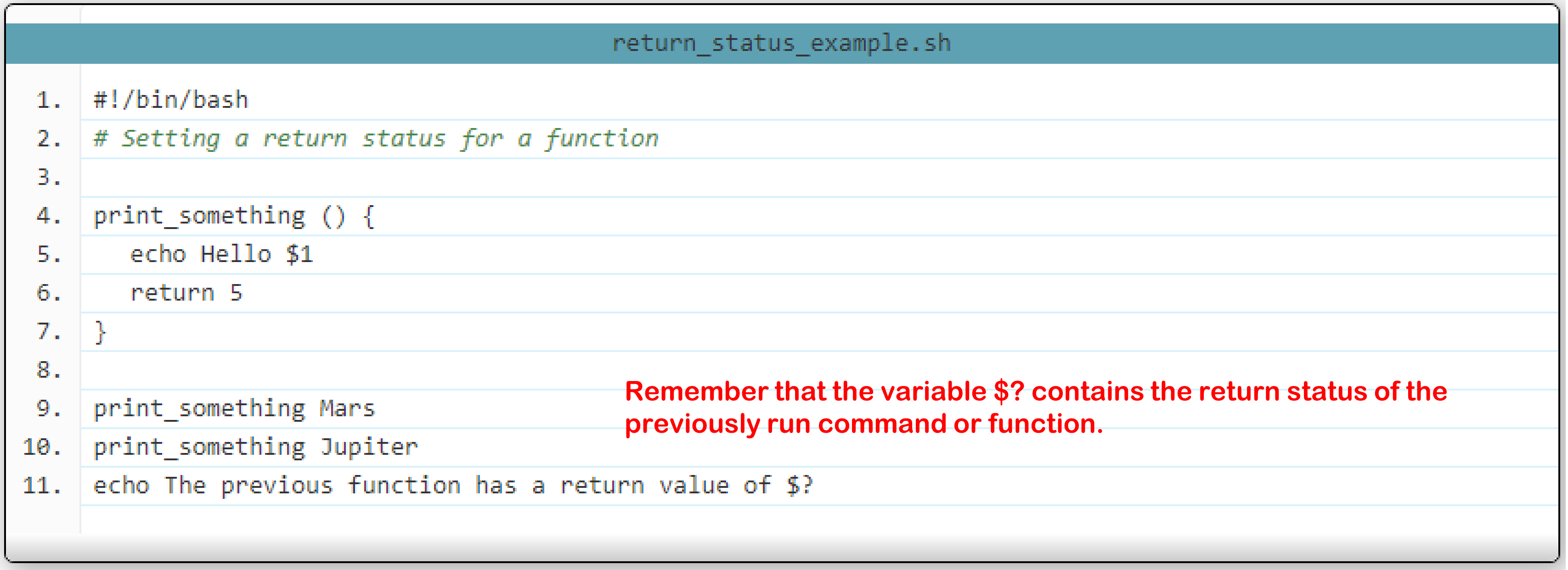
arguments_example.sh

```bash
#!/bin/bash
# Passing arguments to a function

print_something () {
   echo Hello $1
}

print_something Mars
print_something Jupiter
```

https://ryanstutorials.net/bash-scripting-tutorial/bash-functions.php

Functions

- Bash functions don't allow us to return a value.
- They do however allow us to set a return status.
- Typically a return status of 0 indicates that everything went successfully. A non zero value indicates an error occurred.



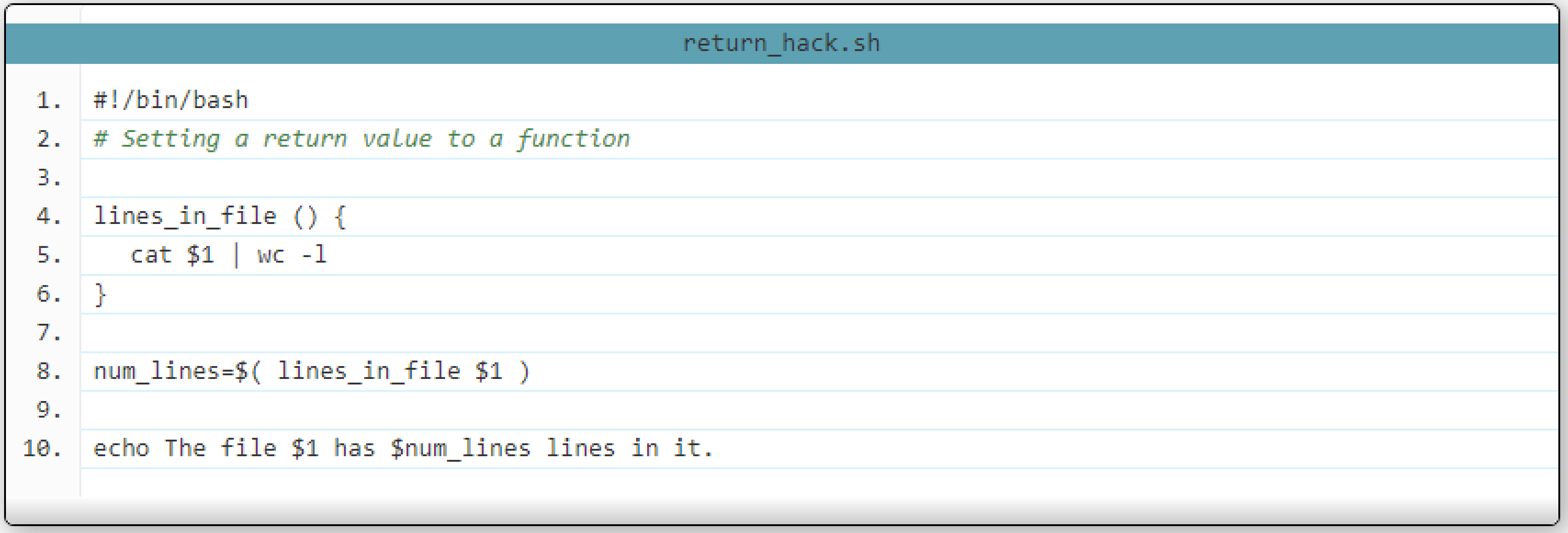
return_status_example.sh

```bash
#!/bin/bash
# Setting a return status for a function

print_something () {
   echo Hello $1
   return 5
}

print_something Mars
print_something Jupiter
echo The previous function has a return value of $?
```

Remember that the variable $? contains the return status of the previously run command or function.

https://ryanstutorials.net/bash-scripting-tutorial/bash-functions.php

Functions


return_hack.sh

```bash
#!/bin/bash
# Setting a return value to a function

lines_in_file () {
   cat $1 | wc -l
}

num_lines=$( lines_in_file $1 )

echo The file $1 has $num_lines lines in it.
```

https://ryanstutorials.net/bash-scripting-tutorial/bash-functions.php

Thanks!

// | ?