

# Bioinformatics Lessons Schedule

NOTE: It's already changed 3 times, so it will continue to change as we go forward.

Date	Subject
10-22	Terminus
10-29	Server Basics
11-5	Server Basics, continued
11-12	Server Basics, continued
11-19	Server Basics, continued
11-26	No lesson, week of Thanksgiving
12-03	Basic Git
12-10	How to run software on the server
12-17	Start running through RRBS?
12-24	Christmas break
12-31	Christmas break
01-07	Resume lessons

# How to Use the Server

Input and Output (I/O), cont.

# Input and Output Definitions

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



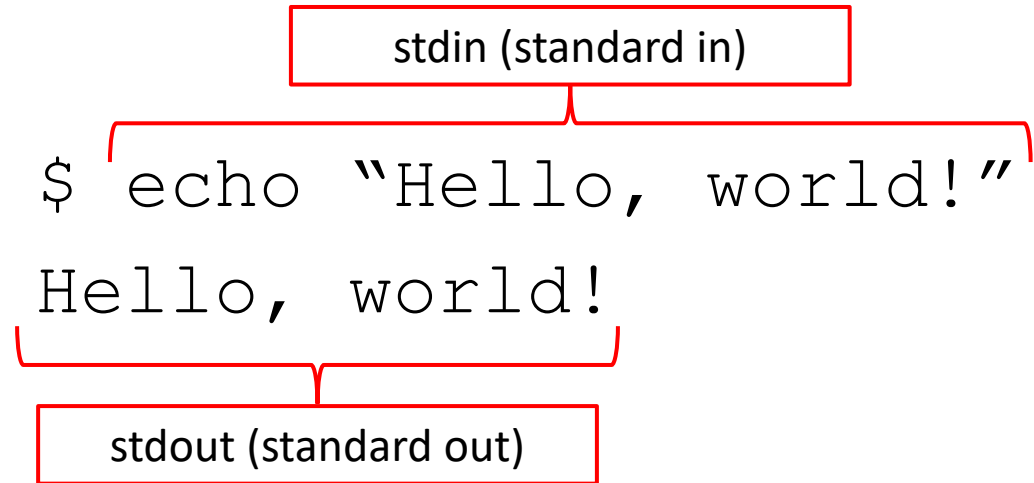
stdin (standard in)

```
$ echo "Hello, world!"
```

The diagram illustrates the concept of standard input (stdin). A red box labeled "stdin (standard in)" is positioned above the command "echo" in the terminal command "\$ echo 'Hello, world!'". A red bracket connects the box to the command, indicating that the command is receiving input from stdin.

# 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



# 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

stdin (standard in)

```
$ echo "Hello, world!"  
Hello, world!
```

stdout (standard out)

```
$ ech "Hello, world!"  
bash: ech: command not  
found...
```

stderr (standard error)

# I/O Operators

```
$ echo "Hello, world!" > hello.txt
```

# I/O Operators

- `>` = Redirect stdout to a file



# I/O Operators

```
$ echo "Hello, world!" > hello.txt  
$ cat hello.txt  
Hello, world!
```

- `>` = Redirect stdout to a file

# I/O Operators

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

```
$ echo "Hello, world!" > hello.txt
$ cat hello.txt
Hello, world!
$ echo "This is how to use a Linux
server." >> hello.txt
```

# I/O Operators

- `>` = Redirect stdout to a file
- `>>` = append
- `<` = redirect file to a command;  
this exists, but people don't  
really use it

```
$ echo "Hello, world!" > hello.txt
$ cat hello.txt
Hello, world!
$ echo "This is how to use a Linux
server." >> hello.txt
$ cat < hello.txt
Hello, world!
This is how to use a Linux server.
```

# I/O Operators

- `>` = Redirect stdout to a file
- `>>` = append
- `<` = redirect file to a command;  
this exists, but people don't  
really use it
- `1>` = redirect stdout

```
$ echo "Hello, world!" > hello.txt
$ cat hello.txt
Hello, world!
$ echo "This is how to use a Linux
server." >> hello.txt
$ cat < hello.txt
Hello, world!
This is how to use a Linux server.
```

# I/O Operators

- `>` = Redirect stdout to a file
- `>>` = append
- `<` = redirect file to a command;  
this exists, but people don't  
really use it
- `1>` = redirect stdout
- `2>` = redirect stderr

```
$ echo "Hello, world!" > hello.txt
$ cat hello.txt
Hello, world!
$ echo "This is how to use a Linux
server." >> hello.txt
$ cat < hello.txt
Hello, world!
This is how to use a Linux server.
```

# I/O Operators

- `>` = Redirect stdout to a file
- `>>` = append
- `<` = redirect file to a command;  
this exists, but people don't  
really use it
- `1>` = redirect stdout
- `2>` = redirect stderr
- `&>` = redirect stdout and  
stderr

```
$ echo "Hello, world!" > hello.txt
$ cat hello.txt
Hello, world!
$ echo "This is how to use a Linux
server." >> hello.txt
$ cat < hello.txt
Hello, world!
This is how to use a Linux server.
```

# I/O Operators

- `>` = Redirect stdout to a file
- `>>` = append
- `<` - redirect file to a command; this exists, but people don't really use it
- `1>` = redirect stdout
- `2>` = redirect stderr
- `&>` = redirect stdout and stderr
- `|` = pipe (inspired `%>%` in R)

```
$ echo "Hello, world!" > hello.txt
$ cat hello.txt
Hello, world!
$ echo "This is how to use a Linux
server." >> hello.txt
$ cat < hello.txt
Hello, world!
This is how to use a Linux server.
$ cat wrong_alphabet.txt | sort |
uniq | 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
```

PRACTICE

BREAK