

Bash

grep, sed, awk, and Functions

grep

- `grep` is a standard *NIX tool to search files for patterns and print matching lines
- There are numerous options for both how matching is done, and what the output of `grep` is
- Options that effect matching
 - `-P`: Use a Perl style regular expression
 - `-e PATTERN`: Specify a pattern after each `-e` (allows multiple patterns)
 - `-f FILE`: Read patterns in from FILE, one per line
 - `-i`: Ignore case
 - `-v`: Invert matches, only print lines that do not match the pattern

```
In [ ]: grep "todo" /usr/src/linux-headers-4.13.0-25/include/soc/tegra/bpmp-abi.h
```

```
In [ ]: grep -R "todo" /usr/src/linux-headers-4.13.0-25/include
```

```
In [ ]: grep -Ri "todo" /usr/src/linux-headers-4.13.0-25/include
```

```
In [ ]: grep -P "^import" /usr/local/lib/python3.6/dist-packages/jupyter_core/*.py
```

```
In [ ]: more src/shell/read_ps_example.sh
```

```
In [ ]: grep -vP "^#[^\!]" src/shell/read_ps_example.sh
```

grep Output Options

- -o: Only output the part of the line that matches
- -color=COLOR_OPTION: suppress or enforce highlighting of matches in line
- -l: Print only the file names where a match has been found
- -L: Print only the file names where no match has been found
- -h: Don't print the file name for each match
- -n: Print the line number the match was found on
- -c: Print the number of matches found in each file

```
In [ ]: grep -Ri "todo" /usr/src/linux-headers-4.13.0-25/include
```



```
In [ ]: grep -Ri --color=always "todo" /usr/src/linux-headers-4.13.0-25/include | head
```

```
In [ ]: grep -Rin --color=always "todo" /usr/src/linux-headers-4.13.0-25/include | head
```

```
In [ ]: grep -Ril "todo" /usr/src/linux-headers-4.13.0-25/include | head
```

```
In [ ]: grep -Pc "^import" /usr/local/lib/python3.6/dist-packages/jupyter_core/*.py
```

```
In [ ]: grep -Ph "^import" /usr/local/lib/python3.6/dist-packages/jupyter_core/*.py | head
```

Printing More Context per Match

- It is often useful, especially in debugging, to print some lines around each match
- `grep` has three flags that control this, each taking a numerical argument
 - `-A NUM`: Print the NUM lines after each match
 - `-B NUM`: Print the NUM lines before each match
 - `-C NUM`: Print the NUM lines before and after each match

```
In [ ]: grep -A4 "todo" /usr/src/linux-headers-4.13.0-25/include/soc/tegra/bpmp-abi.h
```

```
In [ ]: grep -m2 -P --color=never "huge\thuge\tADJ\t" ~/Research/Data/UD_English-EWT/en-ud
-dev.conllu
```



```
In [ ]: grep -m2 -P -C5 --color=never "huge\thuge\tADJ\t" ~/Research/Data/UD_English-EWT/e
n-ud-dev.conllu
```

Grep Practice

- Use `grep` (or `grep` piped to another `grep`, etc.) to return a list of all scripting languages used by files in a directory
 - Look for the shebang line and go from there

`sed` and `awk`

- `sed` and `awk` are both extremely popular tools for text manipulation from the command line
 - `sed` is short for Stream EDitor
 - `awk` is named after the last names of the 3 creators
- Both `sed` and `awk` are full-fledged scripting languages in their own right