

## Methods for passing information to a bash script with use cases

A limited number of arguments that are heterogeneous (e.g. file name and arguments to Unix functions)

```
# usage: bash script.sh input.csv , 1,2
cat $1 | cut -d $2 -f $3 | sort
```

A set of file names to be used in a for loop

```
# usage: bash script.sh *.csv
for file in "$@"
do
    cat $file | head -n 5 | sort -u
done
```

You want your script to be orthogonal and take information from stdin and return it to stdout

```
# usage: cat input.csv | bash script.sh
# usage: cat *.csv | sort | tail -n 100 | bash script.sh
cut -d , -f 1,2 | sort
```

You want to loop through a set of files, but also want to pass additional arguments for use with Unix functions in the script

a) using ' to delay evaluation of wildcards

```
# usage: bash script.sh `*.csv` 5
for file in $1
do
    cat $file | head -n $2 | sort -u
done
```

b) using \$(cat) to put the set of file names from stdin into a script variable

```
# usage: find . -name "*.csv" | bash script.sh 5
list=$(cat)
for file in $list
do
    cat $file | head -n $1 | sort -u
done
```