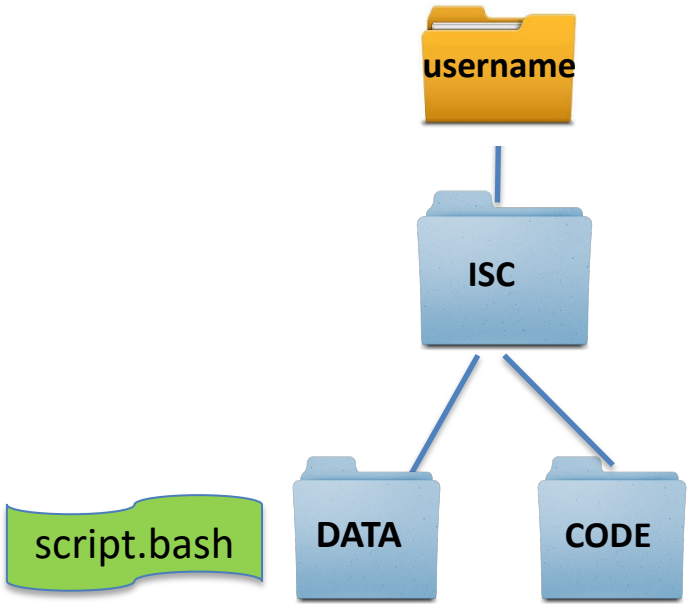
In-class

Before starting, download the zip file ISC.zip and A3-Errors.bash

Run, execute a script

```
. scriptname.bash
source scriptname.bash
bash scriptname.bash
```



Run the script script.bash. Use relative path

- `ISC/script.bash`

Metacharacters: shell expansion

$ expands the value of a variable

```
v1=Hello
echo $v1
```

starts a comment line

~ shortcut to your home directory

```
cd ~
```

; new line of code

```
touch file ; mkdir dir    # two lines of code
```

```
$((arithmetic expression))
```

```
a=10
b=2
c=$((a*b))
echo $((a/b))
```

Arithmetic operators – integer numbers

- `+` addition
- `-` subtraction
- `*` multiplication
- `/` division (rounded down)
- `%` modulo (remainder)
- `**` exponentiation

Metacharacters: wildcard expansion

Wildcards for pattern matching – work on existing files, directories, pathnames

*	zero or more characters – undefined number of characters
?	any single character
[xy]	one character within the brackets
[!xyz]	one character except the ones contained in the brackets

Start from DATA directory

```
ls –l ????  # list in long format files that contain 4 characters

ls *.txt #list all the files ending with .txt

cat [!ab]??? #View the contents of all the files that do not start with a or b and contain 4 characters

rm *[ab] #remove all the files that end with either a or b
```

Metacharacters: brace expansion

Braces expansion: useful for generating multiple files and directories, or for matching existing files and directories

```
generate a sequence of numbers or letters: {start..end}
  {0..12} expanded is 0 1 2 3 4 5 6 7 8 9 10 11 12
  {a..g}  expanded is a b c d e f g

generate a specific list of items {item1,item2,item3}
 {aa,1,cc,3} expanded is  aa 1 cc 3

prefix or suffix string a{0..3}b expanded in a0b a1b a2b a3b

nested   {a,b{1..3},c} expanded is a b1 b2 b3 c
```

We are in DATA directory

```
touch file{1..10}.dat  #make file1.dat to file10.dat

rm file{1,3,9}.txt     #remove file1.dat, file3.dat, and file9.dat

cp file{2,4,6}.dat ../ #copy file2.dat, file4.dat, and file6.dat one
directory back
```

Remove special meaning of Metacharacters

```
Remove special meaning of metacharacters:
‘ inside single quotes, all special characters lose their special
meaning

“ inside double quotes, many metacharacters lose their special
meaning, except $  \ `

\ remove the special meaning of a single metacharacter
```

Make

```
a=10
```

Try these

echo \$a

echo \#\$a

echo \#$a

Variable assignment: space and use of quotes

Rules for variable assignment

- variable name never starts with a number
- No whitespaces in variable definition

```
1v=Hello
v1 = Hello
s1=Hello Unix
```

```
v1=Hello
echo $v1

s1=Hello"  "Unix
s1=Hello'   'Unix
s1=Hello\ Unix
echo $s1

echo Hello Unix #you can echo whitespaces
```

Configuring the vi editor (optional)

You can can setup preferences for vi – like colors, etc. by editing a configuration file called .vimrc
.vimrc must be in your home dirctory

- Make a hidden file called .vimrc within your home directory
- Write in the file

```
syntax on
colorscheme desert
```

If you do not write colorscheme desert it will use the default colors.

Save and quit the script .vimrc

Open a script with vi and you should see colors.

Activity - run a script and fix errors

Download from the Canvas A3-Errors.bash

Put this script in your home directory

Open two terminals.

- Use one terminal to run the script
- Use the second terminal to fix the errors with vi. Use `:w` (so you don't quit vi, and see the code)

Run the script, and fix the errors until the output reads:

```
Hello
I got the # 3
```

Do not remove/add any line of code – fix only the errors

Activity: find me!

The find command is used to search and locate files and directories within the file system.
It will begin looking in the starting directory you specify (**pathname**) and proceed to search through all accessible subdirectories

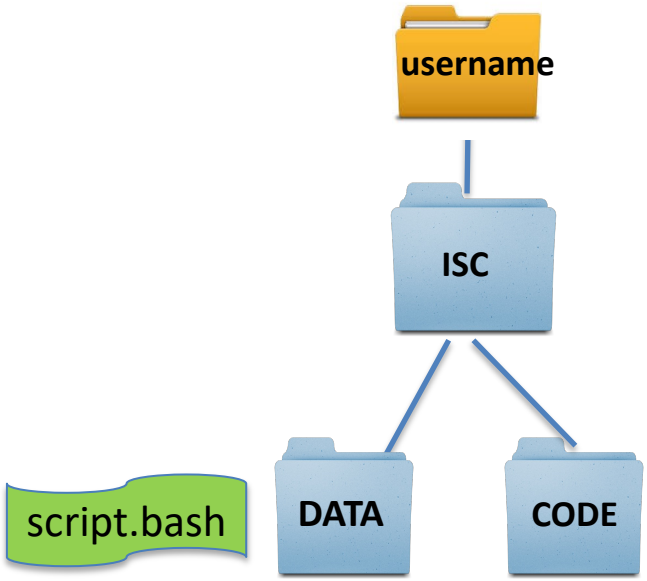
```
find pathname options argument

options: -name match argument
         -type f  find file
         -type d  find directory
```

```
find /Users/mprocop2/Desktop –type f –name file1 #find the location
of file1 staring the search from Desktop, and use absolute path
```

In a script called **A3-find.bash** do the following
You are in home directory:

1. Find the location of the file script.bash. Start the search from ISC directory and use absolute path

2. Find the location of directory CODE. Start the search from ISC directory, and use relative path

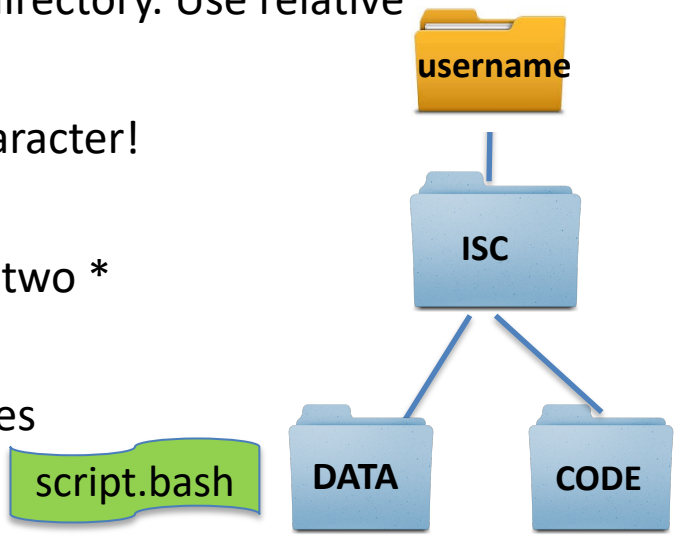


Activity – Practice wildcards and braces

Make a script called **A3-meta.bash** and in it do the following. Write the class header and follow Q&A

0. Start from directory DATA, which contains files

1. Run the script scrip10.bash that is within the CODE directory. Use relative path
2. List all files starting with #. Remember # is a metacharacter!
3. List all files with a number 1 in the name – hint: use two *
4. List all files that have only 4 characters in their names
5. View the contents of all files ending with .dat
6. Make directories dirA, dirC, dirAC within your current directory
7. Make files fileA1.dat, fileA2.dat and fileA3.dat within your current directory



Activity - Practice wildcards and braces

In **A3-meta.bash**

8. Create files
   #thingA.txt
   #thing1.txt
   #thing2.txt

9. Copy all files ending with .dat into dirA

10. Remove directories dirA, dirC, dirAC

11. Optional - Make 1000 files, file1..file1000
12. Optional- Now remove files made in point 11
13. Optional- Go to home directory
14. Optional - Remove ISC

When you copy and move files and directories, only the first argument can contain metacharacters. Do not write metachacters in the second argument, because metachracters carry ambiguous names in the destination folder.

Caution! Before deleting files by using *, ? and other metacharacters ALWAYS perform the ls command first to make sure that you don't delete files that you still need!

Activity - Practice wildcards and braces

In **A3-meta.bash**

15. Optional - Make these directories with one line of code:

2011-01	2011-08	2012-03	2012-10	2013-05	2013-12	2014-07	2015-02	2015-09
2011-02	2011-09	2012-04	2012-11	2013-06	2014-01	2014-08	2015-03	2015-10
2011-03	2011-10	2012-05	2012-12	2013-07	2014-02	2014-09	2015-04	2015-11
2011-04	2011-11	2012-06	2013-01	2013-08	2014-03	2014-10	2015-05	2015-12
2011-05	2011-12	2012-07	2013-02	2013-09	2014-04	2014-11	2015-06	
2011-06	2012-01	2012-08	2013-03	2013-10	2014-05	2014-12	2015-07	
2011-07	2012-02	2012-09	2013-04	2013-11	2014-06	2015-01	2015-08	

Optional –make and run a script

- Make a script called **script-copy.bash** that does the following - follow Q&A format and add your class header

  a. Change into your home directory - is this step necessary? See points b and c
     Write with a comment line if this step is necessary or not

  b. Copy **A3-Errors.bash** into directory CODE by using absolute pathnames

  c. List the content of the CODE directory by using absolute pathname

- Run the script script-copy.bash

Optional

Write a script called **mycal.bash** that will print the current calendar to screen in this format:

The month should be the current month, and so it is different from the output below.

```
**********************
**********************
 September 2022
Su Mo Tu We Th Fr Sa
              1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
**********************
**********************
```

Submit to Gradescope A3

- **A3-Errors.bash 1 point – graded for accuracy**
- **A3-find.bash. 1 point – graded for completion**
- **A3-meta.bash 1-10 mandatory 1 point – graded for completion**