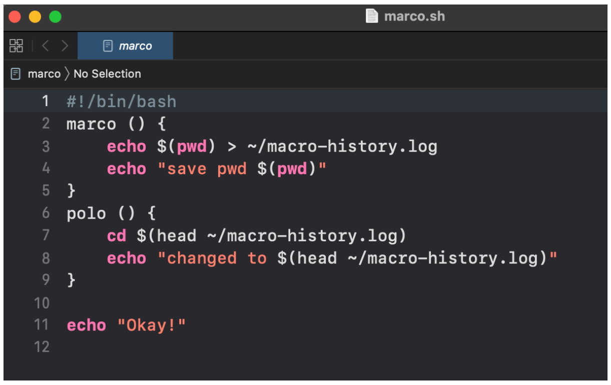
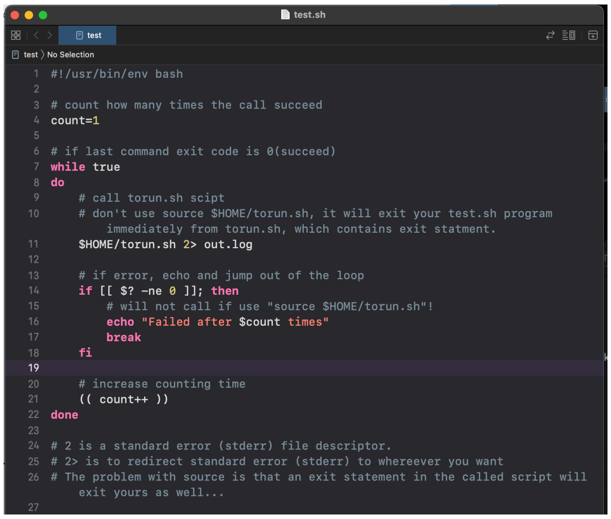
1.
ls -a -l -t(文件以最近访问顺序排序) -h(文件打印以人类可以理解的格式输出) —color=auto(彩色)



```
#!/bin/bash
marco () {
    echo $(pwd) > ~/macro-history.log
    echo "save pwd $(pwd)"
}
polo () {
    cd $(head ~/macro-history.log)
    echo "changed to $(head ~/macro-history.log)"
}

echo "Okay!"

```

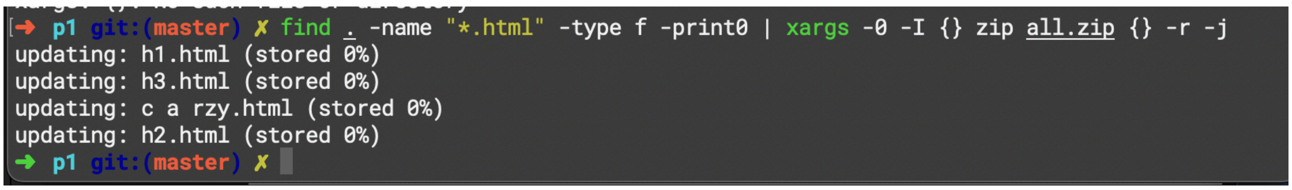
2.

3



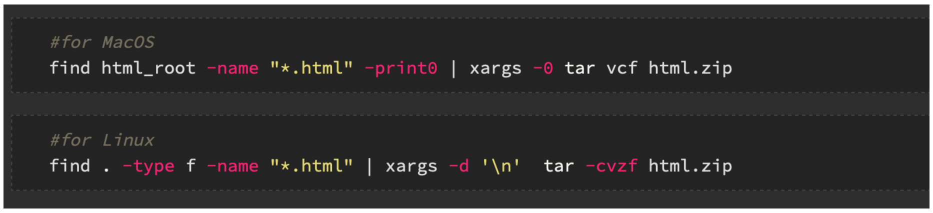
```
#!/usr/bin/env bash

# count how many times the call succeed
count=1

# if last command exit code is 0(succeed)
while true
do
    # call torun.sh scipt
    # don't use source $HOME/torun.sh, it will exit your test.sh program immediately from torun.sh, which contains exit statment.
    $HOME/torun.sh 2> out.log

    # if error, echo and jump out of the loop
    if [[ $? -ne 0 ]]; then
        # will not call if use "source $HOME/torun.sh"!
        echo "Failed after $count times"
        break
    fi

    # increase counting time
    (( count++ ))
done

# 2 is a standard error (stderr) file descriptor.
# 2> is to redirect standard error (stderr) to whereever you want
# The problem with source is that an exit statement in the called script will exit yours as well...

```

4



```
➜ p1 git:(master) ✗ find . -name "*.html" -type f -print0 | xargs -0 -I {} zip all.zip {} -r -j
updating: h1.html (stored 0%)
updating: h3.html (stored 0%)
updating: c a rzy.html (stored 0%)
updating: h2.html (stored 0%)
➜ p1 git:(master) ✗
```

-print0 on find: print on on line.
-type f on file: gets all files From the current directory down, **-type d**: would be all directories onlly.
-0 on xargs: passes that list to your program using a null delimiter instead of the default space delimiter in order to handle file names with spaces., **take care of space in between file names.**
-j on zip: don't want file structure in my zip file
-I Replace occurrences of replace-str in the initial-arguments with names read from standard input. Also, unquoted blanks do not terminate input items; instead the separator is the newline character.



```
#for MacOS
find html_root -name "*.html" -print0 | xargs -0 tar vcf html.zip

#for Linux
find . -type f -name "*.html" | xargs -d '\n'  tar -cvzf html.zip
```

5

5. (进阶) 编写一个命令或脚本递归的查找文件夹中最近使用的文件。更通用的做法，用时间列出文件吗？ `find . -type f -print0 | xargs -0 ls -lt | head -1`

当文件数量较多时，上面的解答会得出错误结果，解决办法是增加 -mmin 条件，先将最近修改的文件进行初步筛选再交给ls进行排序显示 **find . -type f -mmin -60 -print0 | xargs -0 ls -lt | head -10**