

1-Validate an email,print 1 if email is valid, 0 if not valid

[ <https://unix.stackexchange.com/questions/194912/what-is-the-email-matching-regex-in-basic-regex-for-grep> ]

```
echo "mahmoudfierr@gmail.com" | grep -c -E "^[a-zA-Z0-9]+\@[a-zA-Z0-9]+\.[a-z]{2,}"
```

OR

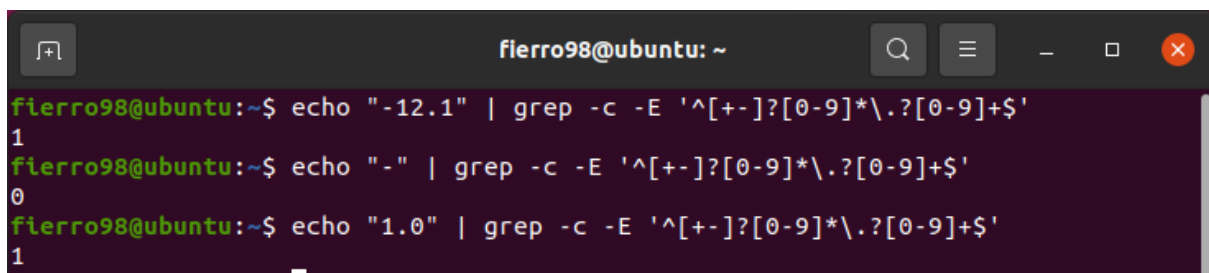
```
echo "mahmoudfierr@gmail.com" | grep -c '^[a-zA-Z0-9]\+[a-zA-Z0-9]\+\.[a-z]\{2,\}'
```



```
fierro98@ubuntu: ~  
fierro98@ubuntu:~$ echo "mahmoudfierr@gmail.com" | grep -c '^[a-zA-Z0-9]\+[a-zA-Z0-9]\+\.[a-z]\{2,\}'  
1  
fierro98@ubuntu:~$ echo "mahmoudfierr" | grep -c '^[a-zA-Z0-9]\+[a-zA-Z0-9]\+\.[a-z]\{2,\}'  
0
```

2-Validate a floating point number (negative and poisitive)

```
echo "-12.1" | grep -c -E "^[+-]?[0-9]*\.[0-9]+$"
```

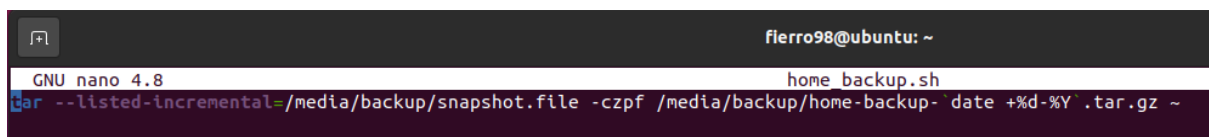


```
fierro98@ubuntu: ~  
fierro98@ubuntu:~$ echo "-12.1" | grep -c -E "^[+-]?[0-9]*\.[0-9]+$"  
1  
fierro98@ubuntu:~$ echo "-" | grep -c -E "^[+-]?[0-9]*\.[0-9]+$"  
0  
fierro98@ubuntu:~$ echo "1.0" | grep -c -E "^[+-]?[0-9]*\.[0-9]+$"  
1
```

3-Backup your home directory in compressed archive in a directory /backup every day The archive name must be home-backup-<day>-<year>.tar.gz

```
touch home_backup.sh
```

```
nano home_backup.sh
```



```
fierro98@ubuntu: ~  
GNU nano 4.8 home_backup.sh  
tar --listed-incremental=/media/backup/snapshot.file -czpf /media/backup/home-backup-`date +%d-%Y`.tar.gz ~
```

```
tar --listed-incremental=/media/backup/snapshot.file -czpf /media/backup/home-backup-`date +%d-%Y`.tar.gz ~
```

```
chmod +754 home_backup.sh
```

```
sudo crontab -u $USER -e
```

```
fierro98@ubuntu: ~
GNU nano 4.8 /tmp/crontab.SakvtT/crontab Modified
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 3 * * * /home/$USER/home_backup.sh
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^_ Replace ^U Paste Text ^T To Spell

0 3 * * * /home/$USER/home_backup.sh
```

```
fierro98@ubuntu:~$ sudo crontab -u fierro98 -e
crontab: installing new crontab
```

#### 4-Save the system load, and memory used in log file (syslog) every min

```
while true; do (echo "%CPU %MEM ARGS $(date)" && ps -e -o pcpu,pmem,args --sort=pcpu | cut -d"
" -f1-5) >> syslog.log; sleep 60; done
```

OR

[ <https://crontab.tech/> ]

```
touch new.sh
```

```
nano new.sh
```

```
GNU nano 4.8
(uptime | cut -d, -f3-5) >> syslog.log;
```

```
(uptime | cut -d, -f3-5) >> syslog.log;
```

```
fierro98@ubuntu: ~
GNU nano 4.8 /tmp/crontab.yLY4zx/cr
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
#0 3 * * * /home/$USER/home_backup.sh
* * * * * /home/$USER/new.sh
```

\*\*\*\*\* /home/\$USER/new.sh

```
fierro98@ubuntu:~$ sudo crontab -u $USER -e
[sudo] password for fierro98:
crontab: installing new crontab
```

Note: Load per process

```
echo "%CPU %MEM ARGS $(date)" && ps -e -o pcpu,pmem,args --sort=pcpu | cut -d" " -f1-5 >>
syslog.log;
```