

U3UR8LD18 : what if i dont want to have that 0  
U3UR8LD18 : i just want to have size of my disk  
U5NMSURQA : don't `print DISK\_SPACE`  
U3UR8LD18 : i can do if else on the basis of DISK\_SPACE and this 0 wont interrupt me ????  
U5NMSURQA : ... I don't understand.  
U1BP42MRS : Why bother writing to a file and then having python read it and email? Unless you are using that file for other usages, it's probably easiest to just call the subprocess from python and email based on your condition.  
U3UR8LD18 : Now i am not printing. I wonder why it's still printing 80"""

```
import os
DISK_SPACE = os.system("df -Th | grep /dev/sda6 | awk '{print $6}' | cut -d'%' -f1")
```

```
if DISK_SPACE > 90:
    print "warning"
else:
    print "everything is fine"
eccentric@debian:~/Downloads$ python test.py
80
everything is fine
"""
```

U5NMSURQA : because os.system just spawns a bunch of processes with current context  
U5NMSURQA : `df` prints to the terminal  
U3UR8LD18 : actually i just want to save that output to DISK\_SIZE variable  
U1BP42MRS : please format your code, use backticks on both sides instead of quotes: ` `` ` <@U3UR8LD18>  
U3UR8LD18 : backtics says syntax error  
U1BP42MRS : In slack, my friend  
U3UR8LD18 : oh...grimacing:  
U3UR8LD18 : thanks  
U1BP42MRS : The short of it is this: `os.system` does not capture the stdout from the subprocess - you need to use the `subprocess` module: <<https://docs.python.org/3/library/subprocess.html#popen-constructor>>  
U5NMSURQA : ``import subprocess

```
output = subprocess.check_output('df -h | fgrep udev', shell=True)
print(repr(output))
``
```

U219WLJNN : If I have an unknown # of properties that come from a JSON document that I want to attach to a Python class and then use that new class for every record moving forward, what's the best way to do that? I was trying to grasp python class factories and was struggling a bit with it  
U5NMSURQA : wow, do you really want exactly that?  
U5NMSURQA : that's crazy!  
U5NMSURQA : ``import json

```
class A:
    def __init__(self, json_data):
        self.__dict__.update(json.loads(json_data))
```

```
text = '{"a": 7, "b": 10, "flag": true}'
a = A(text)
print(vars(a))
``
```

U3UR8LD18 : thanks  
U5NMSURQA : <@U219WLJNN> If it's crazy but works, it's not crazy :wink:  
U5NMSURQA : (but actually it IS)  
U219WLJNN : Well, the properties come from json source 1, and the attributes come from json source 2  
U219WLJNN : so I want to just build out my template from source 1 and then loop through the thousands of records in source 2 with the template as my container.