TypeError: unsupported operand type(s) for %: 'NoneType' and 'int'

2

★

1

```python
def prime(x):
    if (x == 0 or x % 2 == 0):
        return 0
    elif (x == 1):
        return 1
    else:
        for y in range(x-1,0,-1):
            if (x % y == 0):
                return 0
            else:
                pass
        if (y == 1):
            return 1

for x in range(1,20):
    if (prime(x)):
        print ("x:%d, prime YES") % (x)
    else:
        print ("x:%d, prime NO") % (x)
```

I'm starting experimenting Python and I can't understand what's wrong with my code... I'm getting:

> ... print ("x:%d, prime YES") % (x)
> TypeError: unsupported operand type(s) for %: 'NoneType' and 'int'

`python`

edited May 23 '17 at 11:18   asked Feb 27 '14 at 14:04

Mel   peperunas
3,945   9   26   32   **218**   1   3   12

> We need some more information to help you properly. Are you getting an errors? If so, what line is causing the error? Have you tried anything to attempt to rectify it, and if so, what did you try? – snollygolly Feb 27 '14 at 14:07

> there is a case where the `prime` function returns nothing at all (i.e. will return `None`). And `None` cannot be formatted as an integer. – isedev Feb 27 '14 at 14:08

> The error is in the title BTW.. – peperunas Feb 27 '14 at 14:13

> But he never tries to output the result of isprime as an integer, so that's no problem. – RemcoGerlich Feb 27 '14 at 14:13

1 Answer

12

Wait -- I've found it. You are using Python 3! In which `print` is a function. And therefore,

```python
print ("x:%d, prime YES") % (x)
```

actually means

```
(print ("x:%d, prime YES")) % (x)
```

And since `print` returns `None`, that gives you the error you are getting.

Also, beware -- `(x)` is not a tuple containing 1 element, it's simply the value `x`. Use `(x,)` for the tuple.

So just move the parens and add a comma:

```
print("x:%d, prime YES" % (x,))
```

answered Feb 27 '14 at 14:12

RemcoGerlich
20.9k   3   38   58

+1 You are correct :) – thefourtheye Feb 27 '14 at 14:13 ✎

That solved my problem! Thanks! Two ither questions: you said that now print is a function... What was "print" in python 2.x? And, is "y" remembered after my for loop in the prime function? Thanks!!! – peperunas Feb 27 '14 at 14:24

`print` used to be a statement, used with parens. Yes after that for loop, `y` will have the last value it had in the loop. – RemcoGerlich Feb 27 '14 at 14:30

If you use Python 3k, how about `print("X:{}, prime YES".format(x))` ? – nodakai Feb 27 '14 at 23:11