

1 Yes

```
condition = True  
  
if condition:  
    x = 1  
else:  
    x = 0  
  
print(x)
```

better

```
condition = False  
  
x = 1 if condition else 0  
  
print(x)
```

2 Yes

```
num1 = 100000000000  
num2 = 1000000000  
  
total = num1 + num2  
  
print(total)
```

just to make it easy
to read numbers

```
2 num1 = 10_000_000_000  
3 num2 = 100_000_000  
4  
5 total = num1 + num2  
6  
7 print(total)  
8
```

better

```
2 num1 = 10_000_000_000  
3 num2 = 100_000_000  
4  
5 total = num1 + num2  
6  
7 print(f'{total:,}')  
8
```

total
→ better

much better

10100000000

10,100,000,000

Yes

```
f = open('test.txt', 'r')  
file_contents = f.read()  
f.close()  
words = file_contents.split(' ')  
word_count = len(words)  
print(word_count)
```

better

```
with open('test.txt', 'r') as f:  
    file_contents = f.read()  
  
words = file_contents.split(' ')  
word_count = len(words)  
print(word_count)
```

Context manager is meant to manage
resources for me, so I do not have
to manage them by myself

for example: Here we do not have to close
the file manually, because the context
manager will do that for me
automatically

yes

```
2 names = ['Corey', 'Chris', 'Dave', 'Travis']
3
4 index = 0
5 for name in names:
6     print(index, name)
7     index += 1
8
```

```
0 Corey
1 Chris
2 Dave
3 Travis
```

better

```
2 names = ['Corey', 'Chris', 'Dave', 'Travis']
3
4 for index, name in enumerate(names, start=1):
5     print(index, name)
6
```

```
1 Corey
2 Chris
3 Dave
4 Travis
```

yes

```
2 names = ['Peter Parker', 'Clark Kent', 'Wade Wilson', 'Bruce Wayne']
3 heroes = ['Spiderman', 'Superman', 'Deadpool', 'Batman']
4
5 for index, name in enumerate(names):
6     hero = heroes[index]
7     print(f'{name} is actually {hero}')
8
```

```
Peter Parker is actually Spiderman
Clark Kent is actually Superman
Wade Wilson is actually Deadpool
Bruce Wayne is actually Batman
```

Same but with 3 arrays

```
1
2 names = ['Peter Parker', 'Clark Kent', 'Wade Wilson', 'Bruce Wayne']
3 heroes = ['Spiderman', 'Superman', 'Deadpool', 'Batman']
4 universes = ['Marvel', 'DC', 'Marvel', 'DC']
5
6 for name, hero, universe in zip(names, heroes, universes):
7     print(f'{name} is actually {hero} from {universe}')
8
```

output

```
Peter Parker is actually Spiderman from Marvel
Clark Kent is actually Superman from DC
Wade Wilson is actually Deadpool from Marvel
Bruce Wayne is actually Batman from DC
```

better

```
2 names = ['Peter Parker', 'Clark Kent', 'Wade Wilson', 'Bruce Wayne']
3 heroes = ['Spiderman', 'Superman', 'Deadpool', 'Batman']
4
5 for name, hero in zip(names, heroes):
6     print(f'{name} is actually {hero}')
7
```

```
Peter Parker is actually Spiderman
Clark Kent is actually Superman
Wade Wilson is actually Deadpool
Bruce Wayne is actually Batman
```

→ with zip you can put inside it
as much arrays as you want.

more stuff

```
2 names = ['Peter Parker', 'Clark Kent', 'Wade Wilson', 'Bruce Wayne']
3 heroes = ['Spiderman', 'Superman', 'Deadpool', 'Batman']
4 universes = ['Marvel', 'DC', 'Marvel', 'DC']
5
6 for value in zip(names, heroes, universes):
7     print(value)
8
```

→ will put values in a form
of tuples

output .

```
('Peter Parker', 'Spiderman', 'Marvel')
('Clark Kent', 'Superman', 'DC')
('Wade Wilson', 'Deadpool', 'Marvel')
('Bruce Wayne', 'Batman', 'DC')
```

unpacking

means ignore this variable
(we do not want to use it!)

```
2 # Unpacking
3 a, _ = (1, 2)
4
5 print(a)
6 # print(b)
7
```

output

1

problem

```
2 # Unpacking
3 a, b, c = (1, 2, 3, 4, 5)
4
5 print(a)
6 # print(b)
7
```

Traceback (most recent call last):
 File "/Users/coreyschafer/Google Drive/YouTube/Six.py", line 3, in <module>
 a, b, c = (1, 2, 3, 4, 5)
ValueError: too many values to unpack (expected 3)

Solution

```
2 # Unpacking
3 a, b, *c = (1, 2, 3, 4, 5)
4
5 print(a)
6 print(b)
7 print(c)
8
```

1
2
[3, 4, 5]

values from last to first

```
2 # Unpacking
3 a, b, *_ = (1, 2, 3, 4, 5)
4
5 print(a)
6 print(b)
7 # print(c)
8
```

only takes value c

will take always last value because of *

```
2 # Unpacking
3 a, b, *c, d = (1, 2, 3, 4, 5, 6, 7)
4
5 print(a)
6 print(b)
7 print(c)
8 print(d)
9
```

1
2
[3, 4, 5, 6]
7

```
class Person():
    pass
```

```
person = Person()
```

first_key = 'first' make this a variable
first_val = 'Corey' ← to this value
|
print(person.first) ← to be able to do this

So How?

we will use setattr(), getattr()

example with setattr()

```
2 class Person():
3     pass
4
5
6 person = Person()
7
8 first_key = 'first'
9 first_val = 'Corey'
10 setattr(person, first_key, first_val)
11
12 print(person.first)
13
14
15
Corey
```

setattr(classobj, variable, value)

example with getattr()

```
2 class Person():
3     pass
4
5
6 person = Person()
7
8 first_key = 'first'
9 first_val = 'Corey'
10
11 setattr(person, first_key, first_val)
12
13 first = getattr(person, first_key)
14
15 print(first)
16
17
18 # person_info = {'first': 'Corey', 'last': 'Schafer'}
```

Corey

```

2 class Person():
3     pass
4
5
6 person = Person()
7
8 person_info = {'first': 'Corey', 'last': 'Schafer'}
9
10 for key, value in person_info.items():
11     setattr(person, key, value)
12
13 print(person.first)
14 print(person.last)
15

```

Corey
Schafer

for loop use getattr()

```

for key in person_info.keys():
    print(getattr(person, key))

```

5 Common Python mistakes & How to Fix them:

①

```

1
2 nums = [11, 30, 44, 54]
3
4 for num in nums:
5     square = num ** 2
6     print(square)
7

```

Tab ↲
4 spaces ↲
we got
an error ↲

I in this case you need to
change your settings &
enable translate tabs to spaces.

```

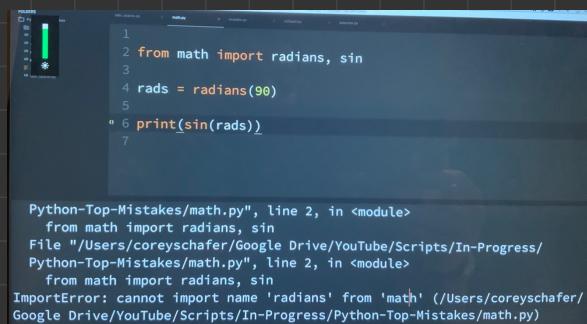
File "/Users/coreyschafer/Google Drive/YouTube/Scripts/In-Progress/
Python-Top-Mistakes/tabs_spaces.py", line 6
    print(square)
          ^

```

IndentationError: unindent does not match any outer indentation level

②

Do not name your module same as like imported module, because
our module have higher priority than imported modules.



```

1
2 from math import radians, sin
3
4 rads = radians(90)
5
6 print(sin(rads))
7

```

```

Python-Top-Mistakes/math.py", line 2, in <module>
    from math import radians, sin
File "/Users/coreyschafer/Google Drive/YouTube/Scripts/In-Progress/
Python-Top-Mistakes/math.py", line 2, in <module>
    from math import radians, sin
ImportError: cannot import name 'radians' from 'math' (/Users/coreyschafer/
Google Drive/YouTube/Scripts/In-Progress/Python-Top-Mistakes/math.py)

```

Don't name a variable
same as the name of
function in python

TypeError: not callable

```

1
2 from math import radians, sin
3
4 rads = radians(90)
5
6 print(sin(radians))
7
8 print(rads)
9

```

```

Traceback (most recent call last):
File "/Users/coreyschafer/Google Drive/YouTube/Scripts/In-Progress/
Python-Top-Mistakes/math.py", line 8, in <module>
    print(radians)
TypeError: 'float' object is not callable

```

```
1 def add_employee(emp, emp_list=[]):
2     emp_list.append(emp)
3     print(emp_list)
4
5
6
7 emps = ['John', 'Jane']
8
9 add_employee('Corey')    { we want -
10 add_employee('John')    } list every-
11
12 ['Corey']                want we
13 ['Corey', 'John']
```

we want a new
list everytime
but we got this

Another example:

```
6 def display_time(time=datetime.now()):  
7     print(time.strftime('%B %d, %Y %H:%M:%S'))  
8  
9  
10 display_time()  
11 time.sleep(1)  
12 display_time()  
13 time.sleep(1)  
14 display_time()  
15
```

April 16, 2019 11:44:27 } ↗ it is not updating
April 16, 2019 11:44:27 the time.
April 16, 2019 11:44:27

s not updating
the time.

because it executing the default

argument one time not every time the code executed.

```
 2 def add_employee(emp, emp_list=None):
 3     if emp_list is None:
 4         emp_list = []
 5     emp_list.append(emp)
 6     print(emp_list)           ^
 7
 8
 9 emps = ['John', 'Jane']
10
11 add_employee('Corey', emps)
12 add_employee('John')

['John', 'Jane', 'Corey']
['John']
['Jane']
```

```
6 def display_time(time=None):
7     if time is None:
8         time = datetime.now()
9     print(time.strftime('%B %d, %Y %H:%M:%S'))
10
11
12 display_time()
13 time.sleep(1)
14 display_time()
15 time.sleep(1)
16 display_time()
17
```

