

Forum:[Hack Forums](#)

Thread: [Thank you for helping me.\[Posibillity dice\]](#)

arkoydas - July 15, 2017, 4:03 pm

I would like to thank you all for helping me learning python. Here is something I made which asks the user what roll dice wants to take and it prints him the possibility of it.

PHP Code

```
<?php
import itertools, random
def Count_times():
    Choice_reversed=Choice[::-1]
    for i in All_dice_rolls:
        if i==(1,1):
            n=0
        if i==Choice or i==Choice_reversed:
            n+=1
    return n
Dice_rolls=[1,2,3,4,5,6]
All_dice_rolls=list(itertools.product(Dice_rolls,range(1,7)))
while 'Hackforums'=='Hackforums':
    try:
        Choice=((int(input("First dice: ")),(int(input("Second dice: "))))))
        break
    except:
        print("Ooops something went wrong.Please try again..!")
print("{}{}".format(round(Count_times()/len(All_dice_rolls)*100,2),"%"))
This is my progress of 4 weeks almost.I know more things than those ;P
Do you have any suggestions to make it better?
```

Davian - July 15, 2017, 4:52 pm

I'm tired asf so i didn't read this thoroughly but please use elif after an if statement thanks.

GameFreak - July 15, 2017, 4:59 pm

U have an except that takes all errors, don't do this. It's bad practice.

Also little comments help a lot because a few months from now you may not remember what stuff does.

Good luck!

lockhead - July 16, 2017, 2:02 am

A few things I noticed:

- 1) Like Bonum said, never catch all exceptions.
- 2) Work on your code readability. This might look nice and concise to you know but for anybody else this is cancer
 - ° Stick with the PEP naming conventions. This maybe sounds stupid but in fact makes your code much nicer

looking and a lot easier to read.

variable_name: all lower case, concatenated multiple words with an underscore

functionName: first word lower case, additional words are capitalized

ClassName: capitalized words

° Pass the variables you're using to the function as parameter, that way you don't have to go looking for

them god knows where. This also becomes more important when getting into OOP where variable scope

becomes more of an issue.

° Your last line is impossible to read. Formatting it like this makes it much more readable.

PHP Code

```
<?php
probability = round(count_times()/len(all_dice_rolls)*100,2)
print("{}%".format(probability))
```

3) Most importantly: this doesn't work.

countTimes() always returns 1 or 2 as int

len(all_dice_rolls) = 36 as int

$2 / 36 = 0$

convert one of the numbers to a float before dividing them

4)

PHP Code

```
<?php
for i in All_dice_rolls:
    if i==(1,1):
        n=0
```

I'm guessing this is your way of setting n = 0 at the first iteration? Why not just do

PHP Code

```
<?php
n = 0
for i in all_dice_rolls:
    if i == choice or i == choice_reversed:
        n += 1
```

Again, it's just easier to read and understand. If you don't believe me, come back in a few months/weeks and look at this code :P

arkoydas - July 16, 2017, 1:39 pm

(Jul 16, 2017 02:02 AM)lockhead Wrote: [\[CITE\]](#) A few things I noticed:

1) Like Bonum said, never catch all exceptions.

2) Work on your code readability. This might look nice and concise to you know but for anybody else this is cancer

° Stick with the PEP naming conventions. This maybe sounds stupid but in fact makes your code much nicer

looking and a lot easier to read.

variable_name: all lower case, concatenated multiple words with an underscore

functionName: first word lower case, additional words are capitalized

ClassName: capitalized words

° Pass the variables you're using to the function as parameter, that way you don't have to go looking for

them god knows where. This also becomes more important when getting into OOP where variable scope

becomes more of an issue.

- Your last line is impossible to read. Formatting it like this makes it much more readable.

PHP Code

```
<?php
probability = round(Count_times()/len(All_dice_rolls)*100,2)
print("{}%".format(probability))
```

3) Most importantly: this doesn't work.

countTimes() always returns 1 or 2 as int

len(all_dice_rolls) = 36 as int

$2 / 36 = 0$

convert one of the numbers to a float before dividing them

4)

PHP Code

```
<?php
for i in All_dice_rolls:
    if i==(1,1):
        n=0
```

I'm guessing this is your way of setting $n = 0$ at the first iteration? Why not just do

PHP Code

```
<?php
n = 0
for i in all_dice_rolls:
    if i == choice or i == choice_reversed:
        n += 1
```

Again, it's just easier to read and understand. If you don't believe me, come back in a few months/weeks and look at this code :P You wrong that $2 / 36 = 0$. You mean $2//36=0$

Maybe you are not using python 3.6 version.

I know that my code is not readable but that's how I am thinking... ;/

I made it in 10 mins so I tried to put what I learned

Ensky - July 17, 2017, 3:48 am

Are you Pakistani, Indian or Bengali?

I can't see a problem in the logic, but your code is severely flawed. I ask if you're one of the above as I've noticed a lot of programmers from those countries neglect white space and don't really try to write readable code.

I highly suggest you follow the PEP8 rules (PyCharm is a great IDE for Python which will highlight the section with a yellow squiggly line underneath, when you hover over it, it'll tell you which PEP8 rule you should be using.)

PEP8 Style Guide: <https://www.python.org/dev/peps/pep-0008/>

Honestly, it's pretty good for a new programmer, but get into good habits with styling while you're still fresh to Python. Your code is ugly and hard to read.

You can always post questions here, or PM me if you like.

Also, you wasn't using the random module, so you could remove it. Not sure why you had while

'Hackforums'=='Hackforums'. It's useless and always returns True, so just use True in its place.

Instead of filling the array out yourself with integers, you can use list comprehensions, such as: `i for i in range(6)`. That's going to populate the array with integers just as a for loop would.

Code import itertools

```
def count_times():
    choice_reversed = choice[::-1]
    for i in all_dice_rolls:
        if i == (1, 1):
            n = 0
        if i == choice or i == choice_reversed:
            n += 1
    return n

dice_rolls = [i for i in range(6)]
all_dice_rolls = list(itertools.product(dice_rolls, range(1, 7)))

while True:
    try:
        choice = int(input('First dice: ')), int(input('Second dice: '))
        break
    except ValueError as e:
        print(e)

print('{} {}'.format(round(count_times() / len(all_dice_rolls) * 100, 2), '%'))
```