

Forum:[Hack Forums](#)

Thread: [Waiting script with turns -> my #4 script in python.](#)

Malentor - July 10, 2017, 11:42 am

first of all thank you for the tips on my previous threads, I have took these tips and used them. A user told me to start using try/except , and another user told me to start using ELIF instead making too many if statements.

Here is what I made with these tips:

PHP Code

```
<?php
import os
```

```
clients_waiting = []
```

```
def options():
```

```
    try:
```

```
        while True:
```

```
            select_option = int(input("""1 = ADD NEW CLIENT TO THE WAITING LIST
2 = SHOW CLIENTS ON THE WAITING LIST
3 = MAKE A CALL TO THE NEXT CLIENT IN THE LIST
INSERT YOUR OPTION: """))
```

```
            if select_option == 1:
```

```
                add_client()
```

```
            elif select_option == 2:
```

```
                show_clients_waiting()
```

```
            elif select_option == 3:
```

```
                make_call()
```

```
        except:
```

```
            if select_option != 1 or 2 or 3:
                print("INVALID")
                options()
```

```
def add_client():
    add_client = raw_input("Client name to add: ")
    clients_waiting.append(add_client)
    print("Added")
    os.system('clear')
    print("Added")
```

```
def show_clients_waiting():
    os.system('clear')
    for a, b in enumerate(clients_waiting):
        a = str(a)
        print(a + "-> name: " + b)
```

```
def make_call():
    os.system('clear')
    who_is_first = min(clients_waiting)
    print(who_is_first)
    clients_waiting.remove(who_is_first)
    print(who_is_first)
```

options()

I feel happy that only had to use google only once for search how to clear the console after every command , the rest was on my mind.

cya next!!

Crypto Elliot - July 10, 2017, 1:55 pm

Looks good! There isn't a problem using google or anything. Most people I know are on stack overflow the whole day. Just wondering, what are you learning python for?

Anxuein - July 10, 2017, 1:56 pm

You should be using Google. I don't mean that in a bad way, a LOT of programmers use Google still and you can find good improvements while doing so.

Charger - July 10, 2017, 2:01 pm

Quit centering your code.

Malentor - July 10, 2017, 2:05 pm

(Jul 10, 2017 01:55 PM)Kryometric Wrote: [\[CITE\]](#) Looks good! There isn't a problem using google or anything. Most people I know are on stack overflow the whole day. Just wondering, what are you learning python for?

My goal for the end of this year is learn as much coding as I can. I already know the basics of PHP and Python, I choose python just because I am little bit familiar with PHP and is almost same syntax.

vorticalbox - July 10, 2017, 2:35 pm

(Jul 10, 2017 02:05 PM)Malentor Wrote: [\[CITE\]](#) My goal for the end of this year is learn as much coding as I can. I already know the basics of PHP and Python, I choose python just because I am

little bit familiar with PHP and is almost same syntax.

No its really not close to php, for one php doesn't use white space for syntax and php requires lines to end with ;

If are different too

```
If($var == 'something')
{
//stuff
}
```

Where as python

```
If var == 'something':
//stuff
```

Malentor - July 10, 2017, 2:43 pm

(Jul 10, 2017 02:35 PM)vorticalbox Wrote: [\[CITE\]](#) No its really not close to php, for one php doesn't use white space for syntax and php requires lines to end with ;

If are different too

```
If($var == 'something')
{
//stuff
}
```

Where as python

```
If var == 'something':
//stuff
```

its almost same to me, I'm sorry if you don't think the same. Also in PHP,Python you don't have to state the type of variable, it auto-detect itself, unlike another languages like java or C++ Thats another reason why I think why python is little bit same than PHP and a very easy language to start with.

Malicious Intent - July 10, 2017, 2:51 pm

In Pytho indentation is very important and this code will not run if you copy and run it. As Charger said .. please dont center the code.. any code... good luck with your learning tho. Python is a good language to know and very useful for quickness. Also .. python might detect the variable type but it doesnt mean you dont need to know or declare the types ...you will understand once you dig deeper into it.

Malentor - July 10, 2017, 2:57 pm

(Jul 10, 2017 02:01 PM)Charger Wrote: [\[CITE\]](#) Quit centering your code.
i just edited the thread, thanks for heads up

Malicious Intent - July 10, 2017, 3:06 pm

(Jul 10, 2017 02:57 PM)Malentor Wrote: [\[CITE\]](#) i just edited the thread, thanks for heads up
Also just noticed .. you dont need to convert the input to int.. its int by default. Another thing is that the exception handling will not work correctly the way it is.. option() is also not needed in exception .. there is few things wrong here but keep trying buddy.

Malentor - July 10, 2017, 3:21 pm

(Jul 10, 2017 03:06 PM)Malicious Intent Wrote: [\[CITE\]](#) Also just noticed .. you dont need to convert the input to int.. its int by default.

in this line?

```
select_option = int(input(
```

I thought i always had to set int for numbers and raw_input for strings

alright thanks :)

Malicious Intent - July 10, 2017, 3:24 pm

(Jul 10, 2017 03:21 PM)Malentor Wrote: [\[CITE\]](#) in this line?

```
select_option = int(input(
```

I thought i always had to set int for numbers and raw_input for strings

alright thanks :)

In python 3+ there is no raw_input .. input can be str or int but its int by default. in python 2.7 you can use raw_input for strings or integers but might have to convert somewhere down the line if you need to make operation on int...

Malentor - July 10, 2017, 6:42 pm

(Jul 10, 2017 03:24 PM)Malicious Intent Wrote: [\[CITE\]](#) In python 3+ there is no raw_input .. input can be str or int but its int by default. in python 2.7 you can use raw_input for strings or integers but might have to convert somewhere down the line if you need to make operation on int...

good to know, thanks

Malicious Intent - July 11, 2017, 1:16 am

Code def add_client():

```
    add_client = raw_input("Client name to add: ")
```

```
    clients_waiting.append(add_client)
```

```
    print("Added")
```

```
    os.system('clear')
```

```
    print("Added")
```

Also there is no need to print "Added" twice here as it will get cleared by the os.system anyway.

Davian - July 12, 2017, 6:26 pm

Just a small mistake I thought is worth mentioning,

PHP Code

```
<?php
```

```
if select_option != 1 or 2 or 3:
```

```
    print("INVALID")
```

```
    options()
```

This block will always execute because you're basically checking if 2 or 3 are True, which they are.

Correct way:

PHP Code

```
<?php
```

```
if select_option != 1 and select_option != 2 and select_option != 3:
```

```
    print("INVALID")
```

```
    options()
```

or even better:

PHP Code

```
<?php
```

```
if select_option not in (1,2,3):
    print("INVALID")
    options()
```

Keep learning dude you're doing great.

lockhead - July 13, 2017, 2:34 am

Keep on practicing, hard work is the road to success!
Here are a few things I noticed you could improve:

1) try/except or exception handling:

You're not using it correctly at all. I suggest you read up on it [here](#) or [here](#)

Basically try/except blocks are supposed to do exception (or error) handling. You're using them for program flow control, which should be done by if/elif/else statements.

Example:

PHP Code

```
<?php
```

```
try:
```

```
    text = input('Enter something --> ')
```

```
except EOFError, e:
```

```
    print(e)
```

```
    print('Why did you do an EOF on me?')
```

```
except KeyboardInterrupt, e:
```

```
    print('You cancelled the operation.')
```

```
except Exception, e:
```

```
    print('generall error occured.. idk what happened here')
```

Notice in this example that specific exceptions are caught (EOFError, KeyboardInterrupt) so that you can respond to the problem at hand and try to fix it. It is no good to just do

PHP Code

```
<?php
```

```
try:
```

```
    someFunction()
```

```
except:
```

```
    print('some error')
```

that doesn't help you at all since you don't know what went wrong and you therefore can't fix it. But please read the links, I'm sure it explains better what exceptions are, how you can raise them, create your own exceptions, how to handle them etc..

2) Don't need to do `int(input('enter stuff'))` since `input()` returns a int. A better way to make sure the user enters a number, not a string is to use exception handling.

PHP Code

```
<?php
```

```
try:
```

```
    select_option = input('enter number')
```

```
except NameError, e:
```

```
    print(e)
```

```
    print('enter a number, fool')
```

```
    #this error would not be raised if the user enters a string like '34' or 'hello'
```

3) You can force a newline by printing `"\n"` like

PHP Code

```
<?php
```

```
print("1) menu item one \n2) menu item two \n3) ...")
```

4) Your functions should perform a specific operation, then return. What you did is make a function called options, which then goes on to call other functions which have nothing to do with that function! This can make code very confusing and impossible to read. Try to keep a clear structure. This is probably the most important (and difficult) point in generally being a good coder.

PHP Code

```
<?php
def main():
    while True:
        selected_option = options()
        if selected_option == 1:
            add_client()
        elif selected_option == 2:
            show_clients_waiting()
        elif selected_option == 3:
            make_call()
        elif selected_option == 4:
            exit()
        else:
            print('wrong number, try again..')
```

main()

[php]

5) Use str.format() instead of

[php]

```
for a , b in enumerate(clients_waiting):
    a = str(a)
    print(a + "-> name: " + b)
```

do

PHP Code

```
<?php
for a , b in enumerate(clients_waiting):
    print("{} -> name: {}".format(a, b))
```

It's much quicker and considered the 'correct' way of formatting strings in python. Read more [here](#)

Anyway don't let that discourage you, keep up the good work :)

Malentor - July 14, 2017, 10:27 am

(Jul 13, 2017 02:34 AM)lockhead Wrote: [\[CITE\]](#) Keep on practicing, hard work is the road to success! Here are a few things I noticed you could improve:

1) try/except or exception handling:

You're not using it correctly at all. I suggest you read up on it [here](#) or [here](#)

Basically try/except blocks are supposed to do exception (or error) handling. You're using them for program flow control, which should be done by if/elif/else statements.

Example:

PHP Code

```
<?php
try:
    text = input('Enter something --> ')
except EOFError, e:
    print(e)
    print('Why did you do an EOF on me?')
```

```
except KeyboardInterrupt, e:
    print('You cancelled the operation.')
```

```
except Exception, e:
    print('generall error occured.. idk what happened here')
```

Notice in this example that specific exceptions are caught (EOFError, KeyboardInterrupt) so that you can respond to the problem at hand and try to fix it. It is no good to just do

PHP Code

```
<?php
try:
    someFunction()
```

```
except:
    print('some error')
```

that doesn't help you at all since you don't know what went wrong and you therefore can't fix it. But please read the links, I'm sure it explains better what exceptions are, how you can raise them, create your own exceptions, how to handle them etc..

2) Don't need to do `int(input('enter stuff'))` since `input()` returns a int. A better way to make sure the user enters a number, not a string is to use exception handling.

PHP Code

```
<?php
try:
    select_option = input('enter number')
except NameError, e:
    print(e)
    print('enter a number, fool')
    #this error would not be raised if the user enters a string like '34' or 'hello'
```

3) You can force a newline by printing `"\n"` like

PHP Code

```
<?php
print("1) menu item one \n2) menu item two \n3) ...")
```

4) Your functions should perform a specific operation, then return. What you did is make a function called `options`, which then goes on to call other functions wich have nothing to do with that function! This can make code very confusing and impossible to read. Try to keep a clear structure. This is probably the most important (and difficult) point in generally being a good coder.

PHP Code

```
<?php
def main():
    while True:
        selected_option = options()
        if selected_option == 1:
            add_client()
        elif selected_option == 2:
            show_clients_waiting()
        elif selected_option == 3:
            make_call()
        elif selected_option == 4:
            exit()
        else:
            print('wrong number, try again..')
```

```
main()
```

[php]

5) Use str.format() instead of

[php]

```
for a , b in enumerate(clients_waiting):  
    a = str(a)  
    print(a + "-> name: " + b)
```

do

PHP Code

```
<?php
```

```
for a , b in enumerate(clients_waiting):  
    print("{} -> name: {}".format(a, b))
```

It's much quicker and considered the 'correct' way of formatting strings in python. Read more [here](#)

Anyway don't let that discourage you, keep up the good work :)

Took note of all these tips, thank you!!!