

Forum:[Hack Forums](#)

Thread: [Python script to create python scripts \(menus\)](#)

Mr. Robot - March 7, 2023, 2:15 am

BIG UPDATE~!

You can now install this using

Code `pip install menuscrypt`

Code `menuscryptor` for help documents

Code `menuscrypt -h`

<https://pypi.org/project/menuscrypt/>

I created this, was going to make it much more complicated but I decided to go with the route of creating something easily accessible and editable for when you need a quick menu.

This should be really easy to break down for even non-coders due to the way I made it work. While you might not at first be able to break down what the starting script does, you will definitely be able to break down the menu script you create using it.

<https://github.com/AlMadeScripts/createmenus>

It can now read arguments! If you put in

Code `python3 create.py -n "name of your script goes here" -o "5"` This would make a file called `name_of_your_script_goes_here.py`

It would then ask you to name your 5 options.

It will also ask you for any commands you normally execute via command line, and assign those to the menu options. Even something like executing other python scripts. (`python3 myotherscript.py`)

You can also just run it like

Code `python3 create.py` This will let you choose the name and options in here rather than via arguments.

En-fucking-joy. This was actually a load of fun to make.

Example code of how your new script might look after creating it.

This is the script it made called `scanner_script.py` from the commands you see in the image above.

Code `#!/usr/bin/env python`

```
####==Import Modules Go Here==####
```

```
import os
```

```
####==Setting your variables to blank values==####
```

```
target = ""
```

```
target2 = ""
```

```
####==Separator Function For Pretty Menu==####
```

```
def print_separator():  
    print("=" * 50)
```

```
####==Your Viewable Menu==####
```

```
def menu():  
    clear_screen()  
    print('Scanner Script!')  
    variable_counts = {}  
    print('Your script has some variables $target, $target2')  
    print_separator()  
    print('1. Nmap')  
    print('2. Ping')  
    variable_counts = {}  
    print(f'set_target | current variable target = {target}')  
    print(f'set_target2 | current variable target2 = {target2}')  
    print('(exit) Exit')
```

```
####==Making Clear Screen Work On Windows And Linux==####
```

```
def clear_screen():  
    os.system('cls' if os.name == 'nt' else 'clear')
```

```
####==When You Choose An Option It Runs One Of These==####
```

```
def nmap():  
    print('You selected nmap')  
    nmap = f"nmap -sV -sC {target}"  
    try:  
        os.system(nmap)  
    except KeyboardInterrupt:  
        print('nmap interrupted by user')  
    input('Press enter to return to menu...')
```

```
def ping():  
    print('You selected ping')  
    ping = f"ping {target2}"  
    try:  
        os.system(ping)  
    except KeyboardInterrupt:  
        print('ping interrupted by user')  
    input('Press enter to return to menu...')
```

```
def set_target():  
    global target  
    target = input('Enter a new target:')  
    input('Press enter to return to menu...')
```

```
def set_target2():  
    global target2  
    target2 = input('Enter a new target:')  
    input('Press enter to return to menu...')
```

```
def exit_menu():  
    print('Goodbye!')
```

exit()

###===Enter Your Choice And Then It Will Activate That Option===###

```
if __name__ == '__main__':  
    while True:  
        menu()  
        choice = input('Enter your choice: ')  
        if choice == '1':  
            nmap()  
        if choice == '2':  
            ping()  
        if choice == 'set_target':  
            set_target()  
        if choice == 'set_target2':  
            set_target2()  
        elif choice.lower() == 'exit':  
            exit_menu()  
        else:  
            print('Invalid choice. Try again.')
```

Stanley - March 7, 2023, 3:37 pm

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP (Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,
Stanley

Mr. Robot - March 7, 2023, 5:03 pm

I made a bunch of changes to it and am updating the thread with those changes.

Even an extremely novice coder should be able to understand how to modify the menu script it creates now.

DashPHP - March 7, 2023, 8:04 pm

Cool thread OP, nice job!

Mr. Robot - March 8, 2023, 4:01 pm

(Mar 7, 2023 09:04 PM)DashPHP Wrote: [\[CITE\]](#) Cool thread OP, nice job!
Updated with the new version I worked on today.

Omniscient - March 8, 2023, 5:28 pm

This seems like a useful script. I don't yet use Python, probably never will. But looks really useful.

Mr. Robot - March 8, 2023, 6:27 pm

(Mar 8, 2023 06:29 PM)Lowkey Wrote: [\[CITE\]](#) We're entering parts of the matrix we didn't think were possible for at least another 12 years.
I have now published it so anyone call install it with
Code pip install menuscrypt

SIM-5330817 - March 9, 2023, 6:09 pm

This is sweet, thanks for sharing.

Mr. Robot - March 9, 2023, 10:55 pm

Update!

It can now accept variables.

Variables are defined by the dollar sign example:

nmap -sV -sC \$target

ping \$website

If you added

nmap and ping as menu options - and defined their command to be those above - you would receive 2 new menu entries in your final file. These will allow you to set those variables whenever you choose.

SIM-5367118 - May 24, 2023, 1:20 pm

This is a useful tool, thank you for sharing! I'm going to give it a try later.

Connor kid - May 30, 2023, 4:48 pm

(Mar 7, 2023 03:15 AM)Mr. Robot Wrote: [\[CITE\]](#) BIG UPDATE~!

You can now install this using

Code pip install menuscrypt

Code menuscryptor for help documents

Code menuscrypt -h

<https://pypi.org/project/menuscrypt/>

I created this, was going to make it much more complicated but I decided to go with the route of creating something easily accessible and editable for when you need a quick menu.

This should be really easy to break down for even non-coders due to the way I made it work. While you might not at first be able to break down what the starting script does, you will definitely be able to break down the menu script you create using it.

<https://github.com/AlMadeScripts/createmenus>

It can now read arguments! If you put in

Code python3 create.py -n "name of your script goes here" -o "5" This would make a file called name_of_your_script_goes_here.py

It would then ask you to name your 5 options.

It will also ask you for any commands you normally execute via command line, and assign those to the menu options. Even something like executing other python scripts. (python3 myotherscript.py)

You can also just run it like

Code python3 create.py This will let you choose the name and options in here rather than via arguments.

En-fucking-joy. This was actually a load of fun to make.

Example code of how your new script might look after creating it.

This is the script it made called scanner_script.py from the commands you see in the image above.

Code `#!/usr/bin/env python`

```
####==Import Modules Go Here==####
import os

####==Setting your variables to blank values==####
target = ""
target2 = ""

####==Separator Function For Pretty Menu==####
def print_separator():
    print("=" * 50)

####==Your Viewable Menu==####
def menu():
    clear_screen()
    print('Scanner Script!')
    variable_counts = {}
    print('Your script has some variables $target, $target2')
    print_separator()
    print('1. Nmap')
    print('2. Ping')
    variable_counts = {}
    print(f'set_target | current variable target = {target}')
    print(f'set_target2 | current variable target2 = {target2}')
    print('(exit) Exit')

####==Making Clear Screen Work On Windows And Linux==####
def clear_screen():
    os.system('cls' if os.name == 'nt' else 'clear')

####==When You Choose An Option It Runs One Of These==####
def nmap():
    print('You selected nmap')
    nmap = f'nmap -sV -sC {target}'
    try:
        os.system(nmap)
    except KeyboardInterrupt:
        print('nmap interrupted by user')
    input('Press enter to return to menu...')

def ping():
    print('You selected ping')
    ping = f'ping {target2}'
    try:
        os.system(ping)
    except KeyboardInterrupt:
        print('ping interrupted by user')
    input('Press enter to return to menu...')
```

```
def set_target():
    global target
    target = input('Enter a new target:')
    input('Press enter to return to menu...')
```

```
def set_target2():
    global target2
    target2 = input('Enter a new target:')
    input('Press enter to return to menu...')
```

```
def exit_menu():
    print('Goodbye!')
    exit()
```

###===Enter Your Choice And Then It Will Activate That Option===###

```
if __name__ == '__main__':
    while True:
        menu()
        choice = input('Enter your choice: ')
        if choice == '1':
            nmap()
        if choice == '2':
            ping()
        if choice == 'set_target':
            set_target()
        if choice == 'set_target2':
            set_target2()
        elif choice.lower() == 'exit':
            exit_menu()
        else:
            print('Invalid choice. Try again.')
Would love some more information on this !!
```

This looks awesome I need to learn more !