

دست گرمی برای پروژه عملی 1

زمان آن رسیده است که شما تمام مهارت های خود را برای شروع ساخت پروژه های قابل استفاده به کار ببندید! قبل از اینکه به پروژه عملی خود بپردازید، ما از طریق تمرینات دست گرمی، شما را با چند ایده کلیدی که در پروژه عملی و پروژه های بزرگتر استفاده می کنیم، آشنا می کنیم، به ویژه:

- دریافت ورودی کاربر
- ایجاد توابع که متغیرها را بر اساس ورودی کاربر ویرایش می کنند
- تولید خروجی
- پیوستن ورودی های کاربر و جریان منطق

تابع نمایش اطلاعات

ایجاد یک تابع که لیست را برای کاربر نشان می دهد

In [1]:

```
1 def display_list(mylist):  
2     print(mylist)
```

In [2]:

```
1 lst = [0,1,2,3,4,5,6,7,8,9,10]  
2 display_list(lst)
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

پذیرش ورودی کاربر

*ایجاد تابعی که ورودی را از کاربر دریافت کرده و نتیجه را با نوع داده اعداد صحیح برمی گرداند. هنگام استفاده از تابع `input()`، مواظب باشید، اجرای دوباره سلول بدون ارائه مقدار ورودی، باعث می شود پایتون در حالت انتظار برای مقدار اولیه در اجرای اول گیر کند. اگر سلول گیر کردید، با توجه به `ln` کنار سلول، هستید، در این صورت، به راحتی `kernel` را راه اندازی مجدد کنید و هر سلول لازم را دوباره اجرا کنید.

In [3]:

```
1 input('Please enter a value: ')
```

Please enter a value: 2

Out[3]:

'2'

In [4]:

```
1 result = input('Please enter a value: ')
```

Please enter a value: 2

In [5]:

```
1 result
```

Out[5]:

'2'

In [6]:

```
1 type(result)
```

Out[6]:

str

In [7]:

```
1 int(result)
```

Out[7]:

2

In [8]:

```
1 result = int(input('Please enter a value: '))
```

Please enter a value: 7

In [9]:

```
1 type(result)
```

Out[9]:

int

In [10]:

```
1 # Example of an error!
2 result = int(input('Please enter a value: '))
```

Please enter a value: two

```
-----
-
ValueError                                Traceback (most recent call las
t)
Cell In[10], line 2
      1 # Example of an error!
----> 2 result = int(input('Please enter a value: '))

ValueError: invalid literal for int() with base 10: 'two'
```

تابع نهایی برای اینکار:

In [11]:

```
1 def user_choice():
2     '''
3     User inputs a number (0-10) and we return this in integer form.
4     No parameter is passed when calling this function.
5     '''
6     choice = input('Please input a number (0-10): ')
7
8     return int(choice)
```

In [12]:

```
1 user_choice()
```

Please input a number (0-10): 5

Out[12]:

5

In [13]:

```
1 result = user_choice()
```

Please input a number (0-10): 7

In [14]:

```
1 type(result)
```

Out[14]:

int

In [15]:

```
1 result
```

Out[15]:

7

تأیید ورودی کاربر

قبل از تلاش برای تبدیل، بررسی کنید که ورودی معتبر است.

ما از یک روش ساده استفاده خواهیم کرد.

هنگامی که پیشرفت می کنید، می توانید شروع به نگاه کردن به روش های دیگر برای انجام این کار کنید (این روش ها در درس های بعدی دوره، بیشتر معنا خواهند داشت، بنابراین در حال حاضر نگران آنها نباشید).

- [بلاگ های متعدد \(https://www.google.com/search?q=python+check+if+input+is+number\)](https://www.google.com/search?q=python+check+if+input+is+number)
- [پست 1 در استک اورفلو \(https://stackoverflow.com/questions/5424716/how-to-check-if-string-input-is-a-number\)](https://stackoverflow.com/questions/5424716/how-to-check-if-string-input-is-a-number)
- [پست 2 در استک اورفلو \(https://stackoverflow.com/questions/1265665/how-can-i-check-if-a-string-represents-an-int-without-using-try-except\)](https://stackoverflow.com/questions/1265665/how-can-i-check-if-a-string-represents-an-int-without-using-try-except)

In [20]:

```
1 some_input = '10'
```

In [21]:

```
1 # Lot's of .is methods availble on string
2 some_input.isdigit()
```

Out[21]:

True

حال تابع را براساس مطالب فوق اصلاح نمایید تا ورودی مناسب را دریافت و پردازش کند

In [22]:

```
1 def user_choice():
2
3     # This original choice value can be anything that isn't an integer
4     choice = 'wrong'
5
6     # While the choice is not a digit, keep asking for input.
7     while choice.isdigit() == False:
8
9         # we shouldn't convert here, otherwise we get an error on a wrong input
10        choice = input("Choose a number: ")
11
12    # We can convert once the while loop above has confirmed we have a digit.
13    return int(choice)
```

In [23]:

```
1 user_choice()
```

Choose a number: hello

Choose a number: two

Choose a number: 5

Out[23]:

5

حال بیا یک پیام خطا را در حلقه while اضافه کنیم.

In [24]:

```
1 def user_choice():
2
3     # This original choice value can be anything that isn't an integer
4     choice = 'wrong'
5
6     # While the choice is not a digit, keep asking for input.
7     while choice.isdigit() == False:
8
9         # we shouldn't convert here, otherwise we get an error on a wrong input
10        choice = input("Choose a number: ")
11
12        # Error Message check:
13        if choice.isdigit() == False:
14            print('Sorry, but you did not enter an integer number. Please try again!')
15
16    # We can convert once the while loop above has confirmed we have a digit.
17    return int(choice)
```

In [26]:

```
1 user_choice()
```

```
Choose a number: asd
Sorry, but you did not enter an integer number. Please try again!
Choose a number: asda
Sorry, but you did not enter an integer number. Please try again!
Choose a number: asdsa
Sorry, but you did not enter an integer number. Please try again!
Choose a number: ads
Sorry, but you did not enter an integer number. Please try again!
Choose a number: asdafsadsf
Sorry, but you did not enter an integer number. Please try again!
Choose a number: 5
```

Out[26]:

5

حالا بیا ببینیم بررسی کنیم چگونه خروجی را "پاک" کنیم، به این ترتیب ما تاریخچه اظهارات "انتخاب یک شماره" را نمی بینیم.

توجه: کاربران Jupyter Notebook از روش IPython نشان داده شده در اینجا استفاده خواهند کرد. سایر کاربران IDE (PyCharm، VS و غیره)

In [27]:

```
1 from IPython.display import clear_output
2 clear_output()
```

In [28]:

```
1 def user_choice():
2
3     # This original choice value can be anything that isn't an integer
4     choice = 'wrong'
5
6     # While the choice is not a digit, keep asking for input.
7     while choice.isdigit() == False:
8
9         # we shouldn't convert here, otherwise we get an error on a wrong input
10        choice = input("Choose a number: ")
11
12        # Error Message check:
13        if choice.isdigit() == False:
14
15            clear_output()
16
17            print('Sorry, but you did not enter an integer number. Please try again!')
18
19        # We can convert once the while loop above has confirmed we have a digit.
20        return int(choice)
```

In [29]:

```
1 user_choice()
```

Sorry, but you did not enter an integer number. Please try again!
Choose a number: 5

Out[29]:

5

بررسی صحت ورودی برای اطمینان از انتخاب یکی از گزینه های قابل قبول

In [30]:

```
1 result = 'wrong value'  
2 acceptable_values = ['0', '1', '2']
```

In [31]:

```
1 result in acceptable_values
```

Out[31]:

False

In [32]:

```
1 result not in acceptable_values
```

Out[32]:

True

In [33]:

```
1 from IPython.display import clear_output  
2 clear_output()
```

In [34]:

```
1 def user_choice():
2
3     # This original choice value can be anything that isn't an integer
4     choice = 'wrong'
5
6     # While the choice is not a digit, keep asking for input.
7     while choice not in ['0', '1', '2']:
8
9         # we shouldn't convert here, otherwise we get an error on a wrong input
10        choice = input("Choose one of these numbers (0, 1, 2): ")
11
12        # Error Message check:
13        if choice not in ['0', '1', '2']:
14
15            clear_output()
16
17            print('Sorry, but you did not enter an integer number. Please try again!')
18
19        # We can convert once the while loop above has confirmed we have a digit.
20        return int(choice)
```

In [35]:

```
1 user_choice()
```

Sorry, but you did not enter an integer number. Please try again!
Choose one of these numbers (0, 1, 2): 2

Out[35]:

2

مثال انعطاف پذیرتر

In [36]:

```
1 def user_choice():
2
3     choice = 'wrong'
4
5     within_range = False
6
7     while choice.isdigit() == False or within_range == False:
8         choice = input("Choose a number (0-10): ")
9
10        if choice.isdigit() == False:
11            print('Sorry that is not a digit!')
12
13        if choice.isdigit() == True:
14            if int(choice) in range(0, 11):
15                within_range = True
16            else:
17                within_range = False
18
19    return int(choice)
```

In [37]:

```
1 user_choice()
```

```
Choose a number (0-10): asd
Sorry that is not a digit!
Choose a number (0-10): 15
Choose a number (0-10): 5
```

Out[37]:

5

تعامل ساده کاربر

در نهایت بیا باید تمام این ایده ها را ترکیب کنیم تا یک بازی کوچک ایجاد کنیم که در آن کاربر می تواند یک "موقعیت" را در یک لیست موجود انتخاب کرده و آن را با یک مقدار از انتخاب خود جایگزین کند.

In [38]:

```
1 game_list = [0,1,2]
```

In [39]:

```
1 def display_game(game_list):
2     print("Here is the current list")
3     print(game_list)
```

In [40]:

```
1 display_game(game_list)
```

Here is the current list
[0, 1, 2]

In [41]:

```
1 def position_choice():
2
3     # This original choice value can be anything that isn't an integer
4     choice = 'wrong'
5
6     # While the choice is not a digit, keep asking for input.
7     while choice not in ['0','1','2']:
8
9         # we shouldn't convert here, otherwise we get an error on a wrong input
10        choice = input("Pick a position to replace (0,1,2): ")
11
12        if choice not in ['0','1','2']:
13            # THIS CLEARS THE CURRENT OUTPUT BELOW THE CELL
14            clear_output()
15
16            print("Sorry, but you did not choose a valid position (0,1,2)")
17
18
19        # Optionally you can clear everything after running the function
20        # clear_output()
21
22        # We can convert once the while loop above has confirmed we have a digit.
23        return int(choice)
```

In [42]:

```
1 def replacement_choice(game_list, position):
2     user_placement = input("Type a string to palce at the position: ")
3
4     game_list[position] = user_placement
5
6     return game_list
```

In [43]:

```
1 def gameon_choice():
2
3     # This original choice value can be anything that isn't a Y or N
4     choice = 'wrong'
5
6     # While the choice is not a digit, keep asking for input.
7     while choice not in ['Y','N']:
8
9         # we shouldn't convert here, otherwise we get an error on a wrong input
10        choice = input("Would you like to keep playing? Y or N ")
11
12
13        if choice not in ['Y','N']:
14            # THIS CLEARS THE CURRENT OUTPUT BELOW THE CELL
15            clear_output()
16
17            print("Sorry, I didn't understand. Please make sure to choose Y or N.")
18
19
20    # Optionally you can clear everything after running the function
21    # clear_output()
22
23    if choice == "Y":
24        # Game is still on
25        return True
26    else:
27        # Game is over
28        return False
```

کنار هم قراردادن منطق بازی

In [45]:

```
1
2 game_on = True
3
4 game_list = [0, 1, 2]
5
6 while game_on:
7     clear_output()
8     display_game(game_list)
9
10    position = position_choice()
11
12    game_list = replacement_choice(game_list, position)
13
14    clear_output()
15    display_game(game_list)
16
17    game_on = gameon_choice()
18
19
20
21
22
23 print('Goodbye')
```

Here is the current list

['hi', 'hello', 2]

Would you like to keep playing? Y or N N

Goodbye

عالی، کار بزرگی انجام دادید! شما در حال حاضر درکی از ترکیب توابع و منطق حلقه برای ساخت یک بازی ساده دارید. این در پروژه مرحله بعدی گسترش خواهد یافت!