

[C1114-OO-EX3][TahrehGholami-401114037180030]

Max Function

```
big = max ('Hello world')
```

```
print(big)
```

تابع ماکزیمم یک تابع است مقداری ورودی می گیرد و یک خروجی تولید می کند.
در خط اول ماکسیمم ارزش حروف تعیین شده در پرانتز یعنی (Hello world) را برابر
big قرار می دهیم یعنی بین حروف کدوم در زبان کامپیوتر ارزش بیشتری دارد آن را
پرینت کن که w با حروف کوچک را پرینت می کند.

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left lists files under 'PYTHON', including 'Max function.py'. The main editor window displays the code for 'Max function.py':

```
1 big = max('Hello world')
2 print(big)
3
```

The Output panel at the bottom shows the execution results:

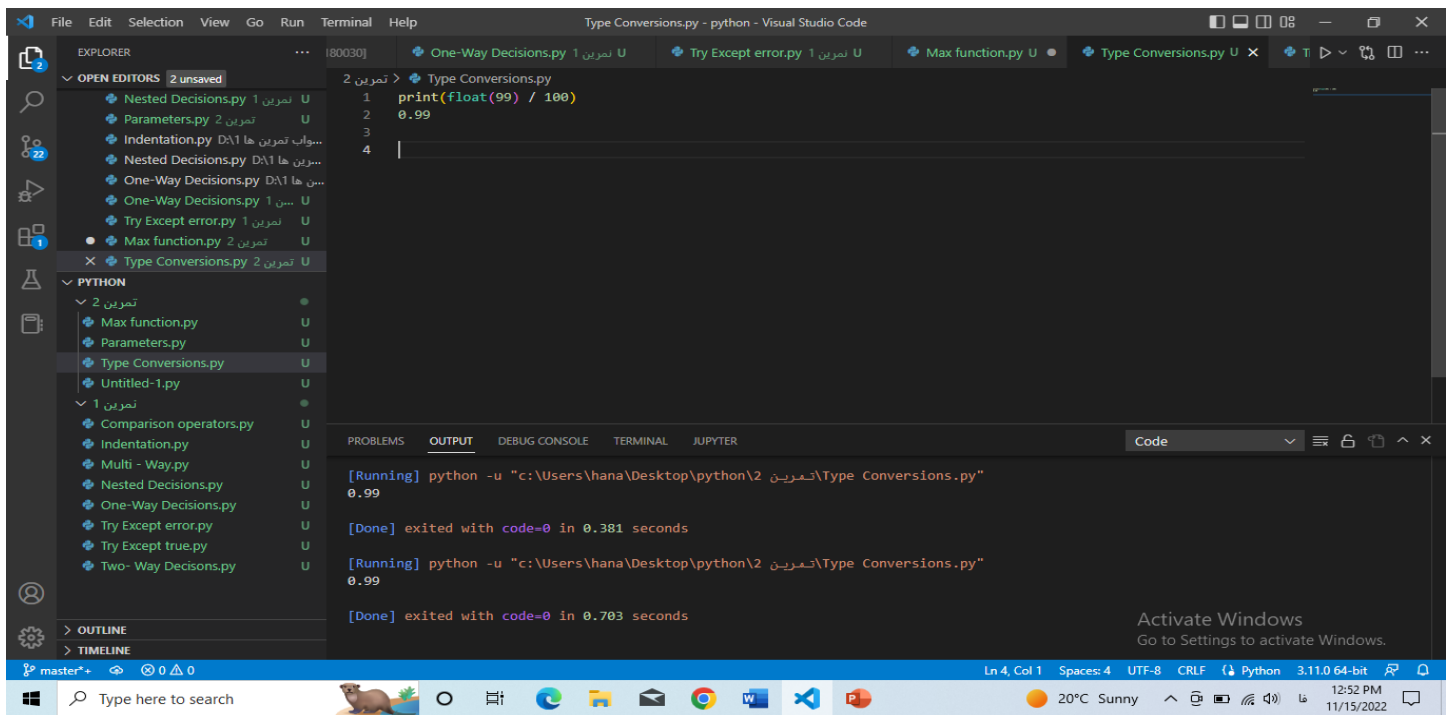
```
[Done] exited with code=0 in 1.609 seconds
[Running] python -u "c:\Users\hana\Desktop\python\2 تمرین\Max function.py"
w
[Done] exited with code=0 in 3.191 seconds
```

```
print (float (99) / 100)
```

```
0.99
```

Type Conversions

این دستور به معنای تبدیل نوع است به این معنی که وقتی یک عدد صحیح و ممیز را در
یک عبارت قرار می دهیم، عدد صحیح به طور ضمنی به یک ممیز تبدیل می شود
با توابع float() , int () می توانیم اینکار را انجام دهیم



def greet(lang):

if lang == 'es':

return 'Hola'

elif lang == 'fr':

return 'Bonjour'

elif lang == 'Fa':

return 'salam'

else:

return 'Hello'

print(greet('en'),'Glenn')

print(greet('es'),'Sally')

print(greet('fr'),'Michael')

print(greet('Fa'),'sara')

Parametrs

این یک "دسته" است که به کد موجود در تابع اجازه می دهد تا به آرگومان های فراخوانی یک تابع خاص دسترسی پیدا کند.

در این کد دستور در خط اول احوال پرسى به زبان گفته شده در واقع هر زبان را به greet مربوط کردیم.

اگر زبان انگلیسی برگردان 'Hola'

اگر زبان فرانسه برگردان 'Bonjour'

اگر زبان فارسی برگردان 'salam'

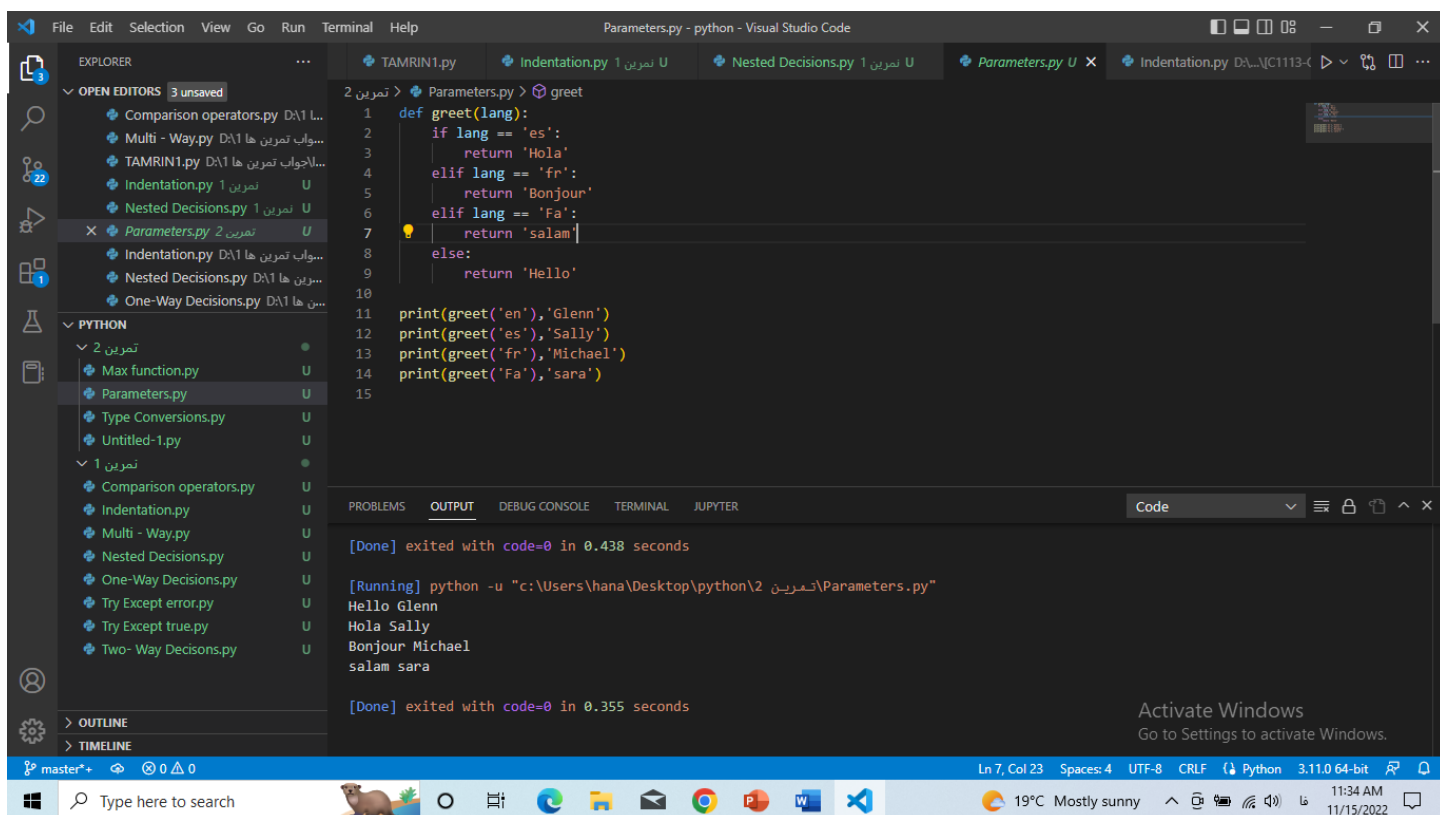
در غیر اینصورت : برگردان 'Hello'

پرینت کن سلام گلن به زبان انگلیسی

پرینت کن سلام سالی به زبان اسپانیایی

پرینت کن سلام میشل به زبان فرانسه

پرینت کن سلام سارا به زبان فارسی



Buliting Function

ما در این دستورات یک تابع جدید را با استفاده از کلمه کلیدی `def` و بدنبال آن پارامترهای اختیاری در پرانتز ایجاد می کنیم.

در خط اول `x` را مساوی 5 قرار می دهیم

خط بعدی از برنامه خواستیم `('Hello')` را پرینت کند.

خط بعدی دستوری نوشته شده که به علت اینکه از `def` استفاده شده دو خط بعدی آن اجرا نخواهد شد.

خط بعدی از برنامه خواستیم `('Yo')` را پرینت کند.

خط بعدی `x` را با عدد 2 جمع می کنیم $7=5+2$

پرینت کن `x` جدید را که عدد 7 چاپ می شود.

```

x = 5

print('Hello')

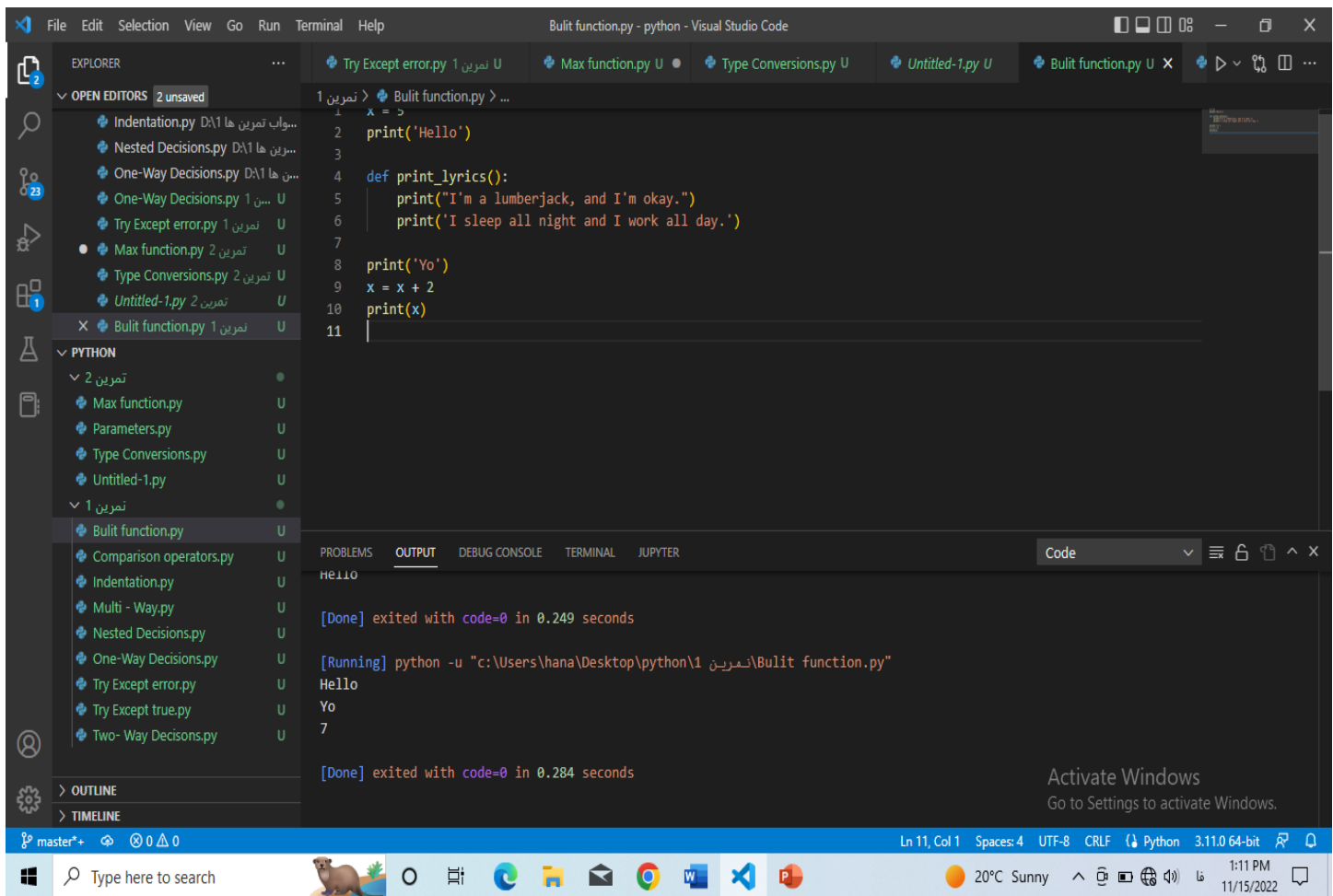
def print_lyrics():
    print ("I'm a lumberjack, and I'm okay.")
    print ('I sleep all night and I work all day.')

print('Yo')

x = x + 2

print(x)

```



String Conversion

Sval = '123'

Type (sval)

<class 'str'>

Print (sval + 1)

در این دستور کد می توانیم از `Int()`, `Float()` برای تبدیل بین رشته ها و اعداد صحیح استفاده کنیم.

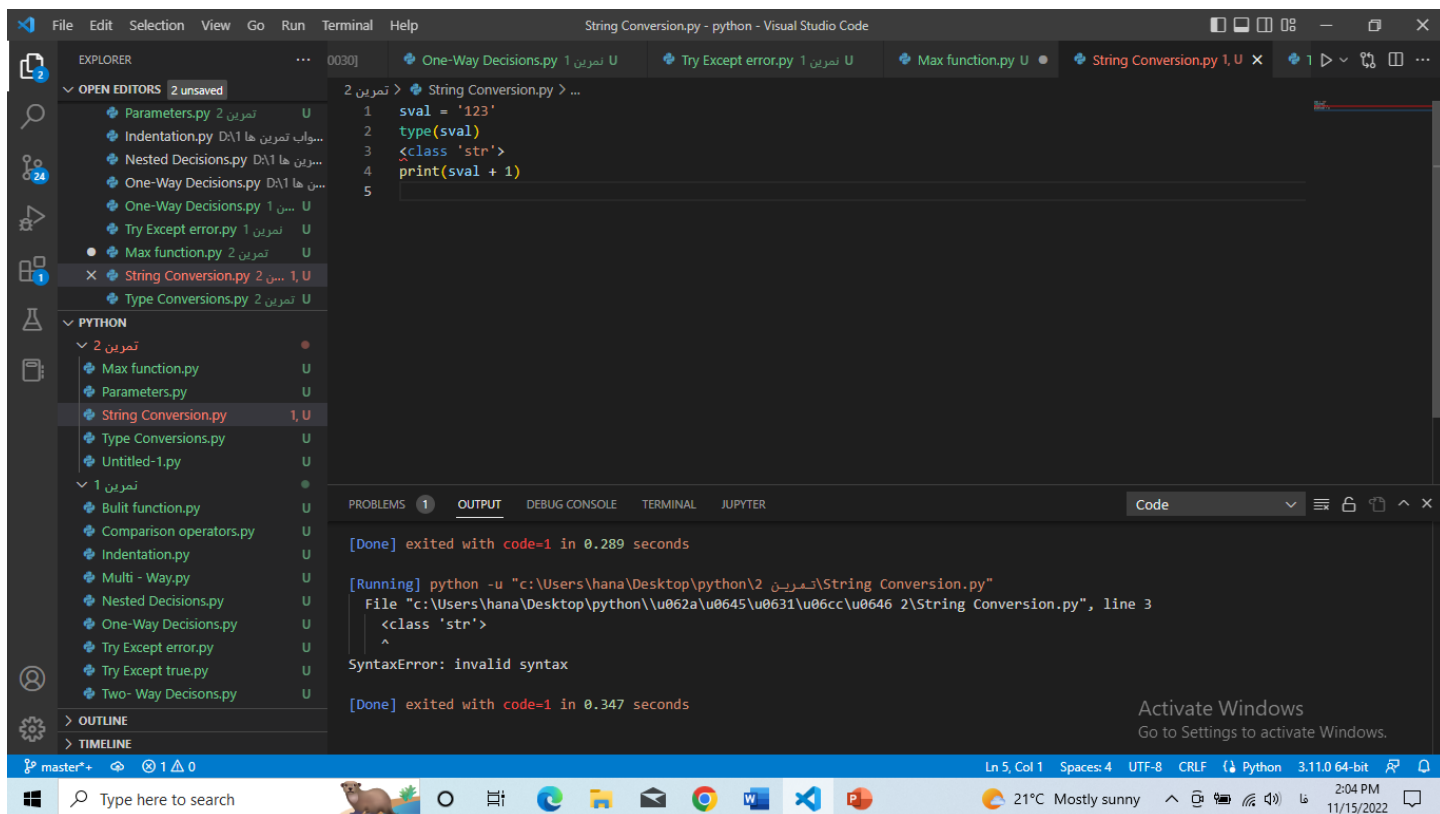
اگر دستور دارای کاراکترهای عددی نباشد، خطا نشان می دهد.

در دستور روبرو مقدار عددی "123" را برابر Sval قرار می دهیم

نوع آن را تایین می کنیم از کلاس استرینگ

در آخر می خواهیم که Sval + 1 جمع کرده و پرینت کند.

که بدلیل عدم تطابق عدد و حروف خطا خروجی می دهد.



def my_function(fname, lname):

print(fname + " " + lname)

my_function("Emil", "Refsnes")

Arguments

آرگومان مقداری که هنگام فراخوانی تابع به عنوان ورودی آن را به تابع می‌فرستیم

از آرگومان‌ها استفاده می‌کنیم تا بتوانیم وقتی تابع را در زمان‌های مختلف فراخوانی می‌کنیم، کارهای مختلفی را انجام دهیم. آرگومان‌ها را بعد از نام تابع در پرانتز قرار می‌دهیم.

در کد روبرو دستور می‌دهیم که بجای **fname** کلمه **email** و بجای **lname** کلمه **Refsnes** را پرینت کند.

```
def my_function(fname, lname):
    print(fname + " " + lname)
my_function("Emil", "Refsnes")
```

Terminal Output:

```
[Done] exited with code=1 in 0.334 seconds

[Running] python -u "c:\Users\hana\Desktop\python\2 تمرین\Arguments.py"

[Done] exited with code=0 in 0.537 seconds

[Running] python -u "c:\Users\hana\Desktop\python\2 تمرین\Arguments.py"
Emil Refsnes

[Done] exited with code=0 in 0.242 seconds
```

big = max('Hello world')

print (big)

در این دستور ('Hello world') به عنوان یک Argument تعریف شده و خواسته شده بین این کلمات بزرگترین حرف از نظر ارزش اسکی آن به زبان 0 و 1 را مشخص کند. که سیستم حرف w را پرینت می کند.

```
big = max('Hello world')
print (big)
```

Terminal Output:

```
File: c:\Users\hana\Desktop\python\1002a\10045\10031\1006C\10040 2\Argu.py, line 3, in <module>
|
| w
|
NameError: name 'w' is not defined

[Done] exited with code=1 in 0.485 seconds

[Running] python -u "c:\Users\hana\Desktop\python\2 تمرین\Argu.py"
w

[Done] exited with code=0 in 0.452 seconds
```

```
def greet():
```

```
    return "Hello"
```

```
print(greet(), "Glenn")
```

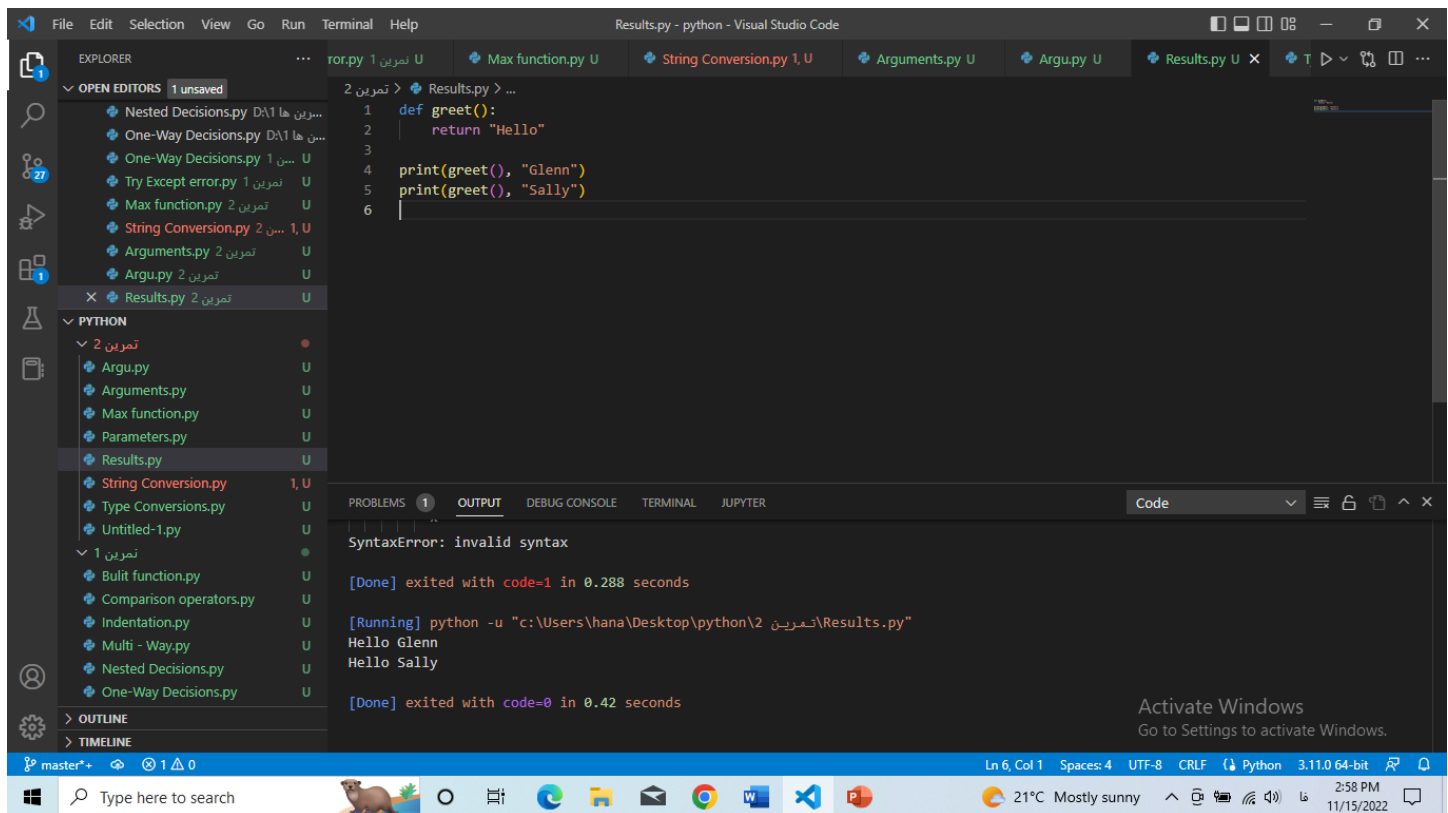
```
print(greet(), "Sally")
```

Results

در این دستور کلمه "Hello" را به `greet()` مربوط می کنیم

در دستور پرینت می خواهیم با توجه به دستور مشخص شده آنها را به اسامی

"Glenn" و "Sally" پرینت کنیم که در خروجی سلام گلن و سلام سالی پرینت می شود.



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left lists several Python files, including 'Results.py'. The main editor displays the following code in 'Results.py':

```
1 def greet():
2     return "Hello"
3
4 print(greet(), "Glenn")
5 print(greet(), "Sally")
6
```

The Output panel at the bottom shows the execution results:

```
[Done] exited with code=1 in 0.288 seconds
[Running] python -u "c:\Users\hana\Desktop\python\2 تمرین\Results.py"
Hello Glenn
Hello Sally
[Done] exited with code=0 in 0.42 seconds
```

An 'Activate Windows' watermark is visible in the bottom right corner of the screenshot.

```
i = 5
```

```
while i:
```

```
    print(i)
```

```
    #i -= 1 #i = i-1
```

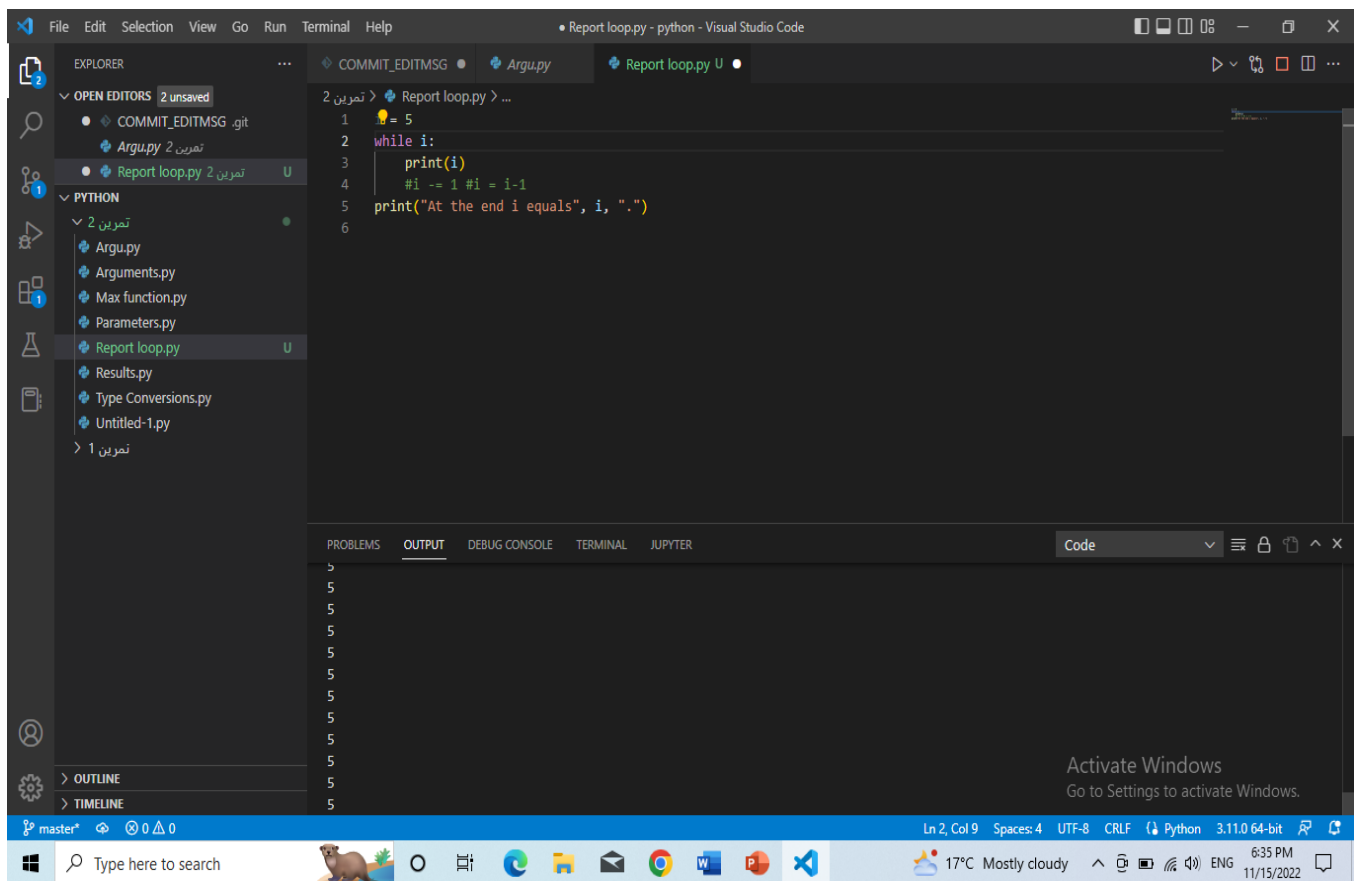
```
print("At the end i equals", i, ".")
```

Loop in Coding

در کد دستور روبرو بخشی از کد که مسئول کاهش مقدار متغیر `i` است را کامنت کردیم .

در هر بار اجرای حلقه چیزی از مقدار اولیه متغیر `i` کم نمی شود و همواره برابر با عدد پنج باقی می ماند و در نتیجه شرط حلقه `while` همواره `True` در نظر گرفته می شود

برنامه روبرو هیچ نقطه پایانی نداشته و تا بی نهایت ادامه می یابد



```
for val in "string":
```

```
    if val == "i":
```

```
        continue
```

```
    print(val)
```

```
print("The end")
```

Finishing an Iteration with continue

دستور **continue** تکرار فعلی را پایان می دهد و به بالای حلقه می پرد و تکرار بعدی را شروع می کند.

در دستور روپرو برای کلمه "string"

بجز حرف **i** بقیه حروف را پرینت کن و چون هر بار به **continue** می رسد به بالا می پرد حروف زیر هم نوشته می شوند البته به جز حرف **i**

continue.py - python - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

Report loop.py A COMMIT_EDITMSG exl2.py Exercise2.py U Exercise 2.pdf U continue.py U

OPEN EDITORS 1 unsaved

Report loop.py 2 تمرین A

COMMIT_EDITMSG .git

exl2.py C:\Users\hana\Downloads

Exercise2.py 2 تمرین U

Exercise 2.pdf 2 تمرین U

continue.py 2 تمرین U

PYTHON

2 تمرین

[C1113-OO-EX2][TahrehGhola... U

~\$2 تمرین.docx U

~WRL1060.tmp U

Argu.py

Arguments.py

continue.py U

Exercise 2 - .pdf U

Exercise 2.pdf U

Exercise2.py U

Max function.py

Parameters.py

Report loop.py A

Results.py

Type Conversions.py

Untitled-1.py

2 تمرین.docx U

تمرین 1

Built function.py

OUTLINE

TIMELINE

2 تمرین > continue.py > ...

```
1 for val in "string":
2     if val == "i":
3         continue
4     print(val)
5
6 print("The end")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Code

> [Running] python -u "c:\Users\hana\Desktop\python\2 تمرین\continue.py"

s

t

r

n

g

The end

[Done] exited with code=1 in 218.444 seconds

Activate Windows
Go to Settings to activate Windows.

master+ 0 0 0

Ln 6, Col 17 Spaces: 4 UTF-8 CRLF Python 3.11.0 64-bit

Type here to search

16°C Clear 10:44 PM 11/15/2022