



Recap



```
print("an example input")
```

```
type("an example input")
```

```
اسم تابع print("ان ورودی example input")
```

Recap



Variable Type	Description	Example Assignment
<code>int</code> (Integer)	Whole numbers, positive or negative, without decimals	<code>x = 10</code>
<code>float</code> (Floating Point)	Numbers with decimal points	<code>y = 3.14</code>
<code>str</code> (String)	A sequence of characters	<code>name = "Ali"</code>
<code>bool</code> (Boolean)	Logical value: <code>True</code> or <code>False</code>	<code>is_active = True</code>

Recap



عملگرهای حسابی

Arithmetic Operators

Operators	Meaning	Example	Result
+	Addition	$4 + 2$	6
-	Subtraction	$4 - 2$	2
*	Multiplication	$4 * 2$	8
/	Division	$4 / 2$	2
%	Modulus operator to get remainder in integer division	$5 \% 2$	1
**	Exponent	$5 ** 2 = 5^2$	25
//	Integer Division/ Floor Division	$5 // 2$ $-5 // 2$	2 -3

```
print("salam")  
print(3 + 0.14)
```

output



salam

3.14

What about input ?



اسم تابع (ورودی "an example input")

input()

بدون ورودی

```
>>> input()
my text :)
'my text :)'
```



نشان دادن سر کلاس

```
main.py +
1 print("salam")
2 input()
Ln: 2, Col: 8
Run Share
salam
wrt324rw ←
```

متغیر

```
>>> name = input()
```

Masoud

مقدار دریافتی

متن تایپ شده

print(name) ?

```
>>> name = input()
Ali
>>> print(name)
Ali
```

تایپ شده توسط کاربر
(ورودی برنامه) ←

خروجی برنامه ←


```
اسم تابع print("an example input"ورودی)
```

```
input("enter your name: ")
```

```
input("enter your name: ")
```

```
>>> name = input("enter your name: ")
enter your name: Masoud
>>> print(name)
Masoud
```



نشان دادن سر کلاس

```
name = input("enter your name: ")  
print("Hi ", name)
```



main.py



```
1 name = input("enter your name: ")
2 print("Hi ",name)
3 |
```

Ln: 3, Col: 1



Run



Share



Command Line Arguments



enter your name:



Masoud



Hi Masoud

```
>>> name = input("enter your name: ")  
enter your name: Masoud  
>>> type(name)
```



```
>>> name = input("enter your name: ")
enter your name: Masoud
>>> type(name)
<class 'str'>
```

برنامه ای بنویسید که در آن یک عدد از کاربر گرفته شود و دو برابر آن عدد به کاربر نمایش داده شود



```
number = input("enter your number: ")  
print( 2 * number )
```



suggestion

main.py



```
1 number = input("enter your number: ")
2 print( 2 * number )
3
4 |
```

Ln: 4, Col: 1



Run



Share



Command Line Argument



enter your number:



45



4545


```
>>> 2 * "He"
```

```
'HeHe'
```

```
>>> 2 * "He "
```

```
'He He '
```





main.py


+

```
1 number = input("enter your number: ")
2 print( 2 * number )
3
4 |
```


Ln: 4, Col: 1

 Run


 Share

 \$


Command Line Argument



enter your number:



45



4545

```
>>> 2 * 45
90
```

```
>>> 2 * "45"
'4545'
```

"45"

convert



45



`int("45")`

`int("45")`

		<code>>>> "45"</code>
str	→	<code>'45'</code>
		<code>>>> int("45")</code>
int	→	<code>45</code>

Type Conversion

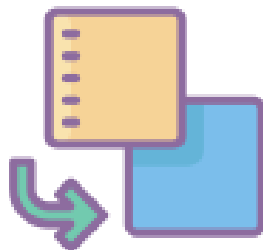
```
>>> number = "45"  
>>> type(number)  
<class 'str'>  
>>> number = int(number)  
>>> type(number)  
<class 'int'>
```

برنامه ای بنویسید که در آن یک عدد از کاربر گرفته شود و دو برابر آن عدد به کاربر نمایش داده شود

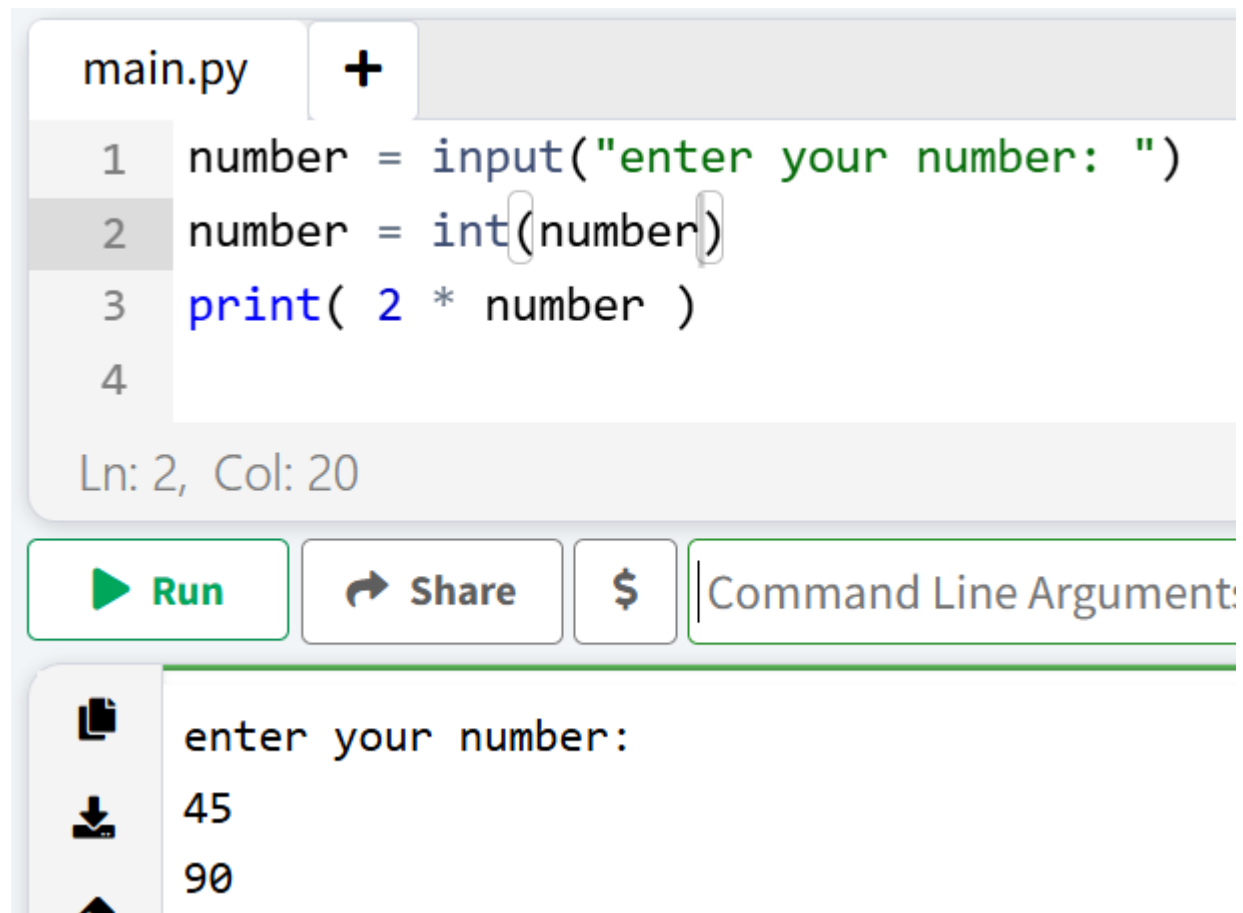
```
number = input("enter your number: ")  
print( 2 * number )
```



What change ?



برنامه ای بنویسید که در آن یک عدد از کاربر گرفته شود و دو برابر آن عدد به کاربر نمایش داده شود



The image shows a Python IDE interface. At the top, there's a tab labeled 'main.py' with a '+' icon to its right. Below the tab, the code is displayed with line numbers 1 through 4 on the left. The code is:
1 number = input("enter your number: ")
2 number = int(number)
3 print(2 * number)
4
Below the code, it says 'Ln: 2, Col: 20'. At the bottom of the editor, there are three buttons: 'Run' (with a green play icon), 'Share' (with a right arrow icon), and '\$' (with a dollar sign icon). To the right of these buttons is a text input field labeled 'Command Line Arguments:'. Below the editor, there's a console area with a copy icon on the left. The console shows the output of the program: 'enter your number:' followed by '45' and '90' on separate lines.

```
main.py +  
1 number = input("enter your number: ")  
2 number = int(number)  
3 print( 2 * number )  
4  
Ln: 2, Col: 20  
  
Run Share $ Command Line Arguments:  
  
enter your number:  
45  
90
```

```
>>> number = "65"
```

```
>>> int(number)
```

```
65
```

```
>>> number = "salam"
```

```
>>> int(number)
```

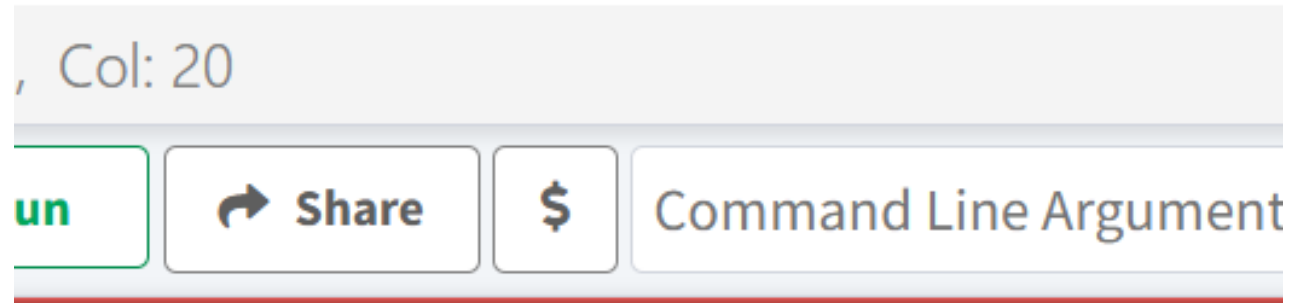



```
>>> number = "salam"  
>>> int(number)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
ValueError: invalid literal for int() with base  
10: 'salam'
```



نشان دادن سر کلاس

```
number = input("enter your number: ")  
number = int(number)  
print( 2 * number )
```



enter your number:

3.14



```
>>> int("3.14")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base
10: '3.14'
```

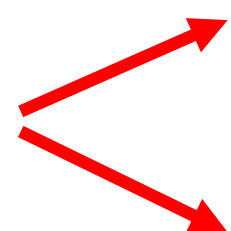
~~`int("3.14")`~~



```
>>> float("3.14")
3.14
```

```
float("3")
```











```
>>> float("3")
3.0
>>> int("3")
3
```

برنامه ای بنویسید که در آن یک عدد از کاربر گرفته شود و دو برابر آن عدد به کاربر نمایش داده شود

```
main.py +
1 number = input("enter your number: ")
2 number = float(number)
3 print( 2 * number )
4 |
```

Ln: 4, Col: 1

 Run  Share  Command Line Arguments

 enter your number:
 2.5
 5.0

```
>>> int(3.14)
```



```
>>> int(3.14)  
3
```



```
>>> int(3.81)
```



```
>>> int(3.81)
3
```

```
>>> "Masoud" + "Khodaverdian"
'MasoudKhodaverdian'
>>> "Masoud" + " " + "Khodaverdian"
'Masoud Khodaverdian'
```

```
>>> age = 25
```

```
>>> "Masoud age is " + age
```



```
>>> age = 25
>>> "Masoud age is " + age
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int")
to str
```

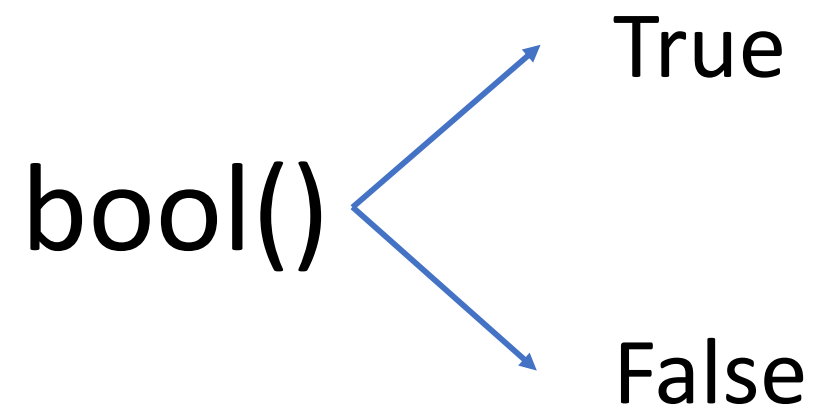
```
>>> str(19)
```

```
'19'
```

```
>>> str(3.14)
```

```
'3.14'
```

- int()
- float()
- str()
- ?




```
>>> bool(1)
```



```
>>> bool(1)
True
```

```
>>> bool(0)  
False
```

```
print(bool(0))    # False
print(bool(1))    # True
print(bool(-5))   # True
print(bool(3.14)) # True
print(bool(0.0))  # False
```

```
>>> bool("dhgerth")
```

```
True
```

```
>>> bool("")
```

```
False
```

```
bool("0")
```



```
>>> bool("0")  
True
```

```
|>>> bool(" ")
```




```
>>> bool(" ")  
True
```

