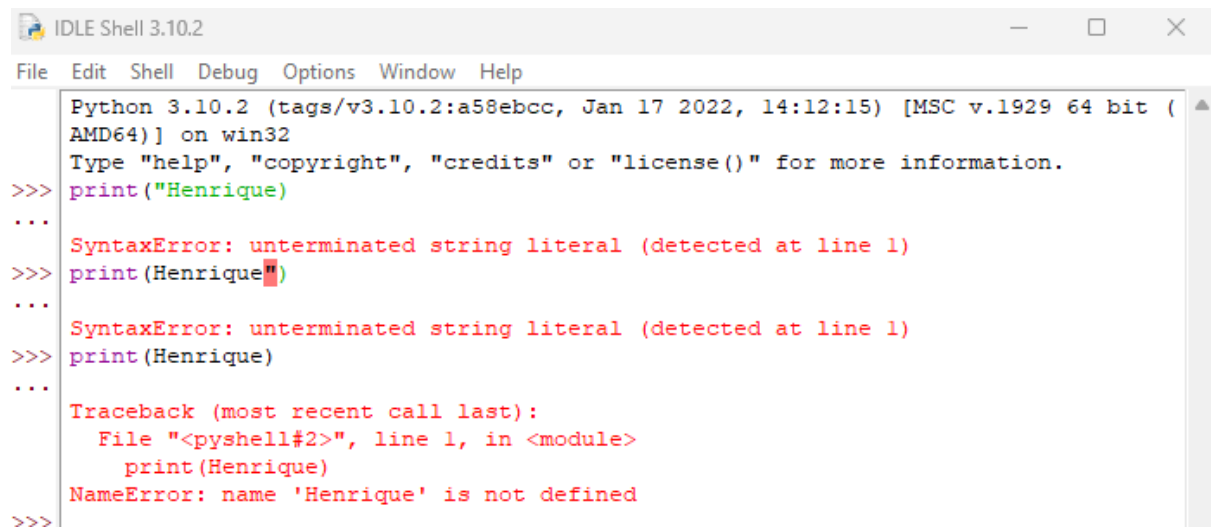


## Part 1: LEARN FROM YOUR MISTAKES!

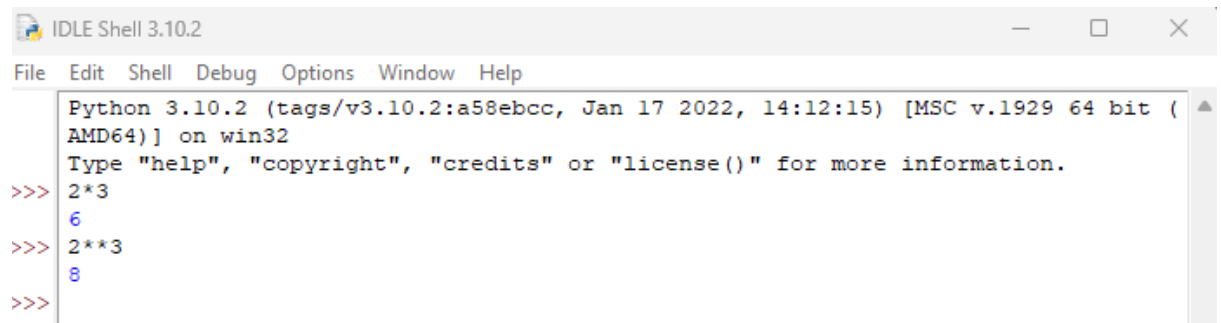
- a. If you are trying to print your name, what happens if you leave out one of the quotation marks or both, and why?

In the case of leaving out one of the quotation marks, Python throws a `SyntaxError` message. In the case of leaving out both, Python understands the input as a variable. Because this possible variable is not declared, Python throws a `NameError` message.



```
IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Henrique)
...
SyntaxError: unterminated string literal (detected at line 1)
>>> print(Henrique)
...
SyntaxError: unterminated string literal (detected at line 1)
>>> print(Henrique)
...
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    print(Henrique)
NameError: name 'Henrique' is not defined
>>>
```

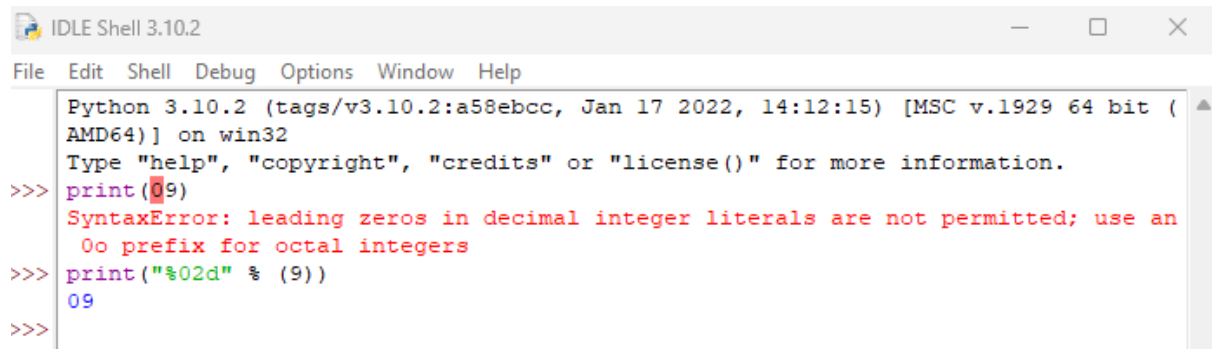
- b. What is the difference between `*` and `**` operators in Python? Explain with the help of an example.  
`*` operator is responsible for multiplications, and `**` operator is responsible for exponentiation.



```
IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2*3
6
>>> 2**3
8
>>>
```

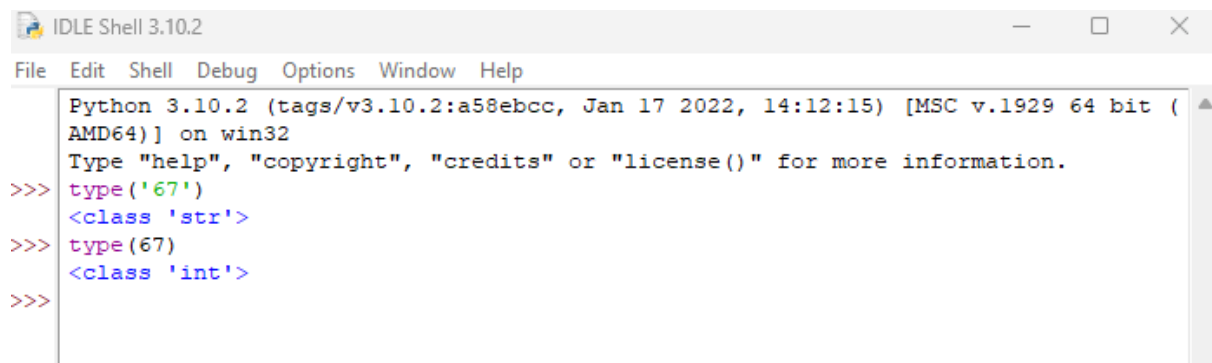
c. In Python, is it possible to display an integer like 09? Justify your answer.

No, because it has a leading zero. If you need to display the value “09” you should pass it as a string.



```
IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print(09)
SyntaxError: leading zeros in decimal integer literals are not permitted; use an
0o prefix for octal integers
>>> print("%02d" % (9))
09
>>>
```

d. Run the commands `type('67')` and `type(67)`. What is the difference in the output and why?  
`type('67')` shows that the input is a string and `type(67)` show that the input is an integer. That happens because, although both are numbers, '67' is interpreted by Python as an array of characters, and 67 as a number.



```
IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> type('67')
<class 'str'>
>>> type(67)
<class 'int'>
>>>
```