

Python Tutorial

```
In [103]: import sys
import keyword
import operator
from datetime import datetime
import os
```

Keywords

Keywords are the reserved words in Python and can't be used as an identifier

```
In [3]: print(keyword.kwlist) # List all Python Keywords

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'cl
ass', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'fr
om', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

```
In [4]: len(keyword.kwlist) # Python contains 35 keywords
```

```
Out[4]: 35
```

Identifiers

An identifier is a name given to entities like class, functions, variables, etc. It helps to differentiate one entity from another.

```
In [13]: lvar = 10 # Identifier can't start with a digit

File "<ipython-input-13-37e58aaf2d3b>", line 1
    lvar = 10 # Identifier can't start with a digit
        ^
SyntaxError: invalid syntax
```

```
In [14]: val2@ = 35 # Identifier can't use special symbols

File "<ipython-input-14-cfbf60736601>", line 1
    val2@ = 35 # Identifier can't use special symbols
        ^
SyntaxError: invalid syntax
```

```
In [15]: import = 125 # Keywords can't be used as identifiers

File "<ipython-input-15-f7061d4fc9ba>", line 1
      import = 125 # Keywords can't be used as identifiers
            ^
SyntaxError: invalid syntax
```

```
In [16]: """
Correct way of defining an identifier
(Identifiers can be a combination of letters in lowercase (a to z) or uppercase
"""

val2 = 10
```

```
In [17]: val_ = 99
```

Comments in Python

Comments can be used to explain the code for more readability.

```
In [18]: # Single line comment
val1 = 10
```

```
In [19]: # Multiple
# line
# comment
val1 = 10
```

```
In [20]: '''
Multiple
line
comment
'''
val1 = 10
```

```
In [21]: """
Multiple
line
comment
"""
val1 = 10
```

Statements

Instructions that a Python interpreter can execute.

```
In [94]: p = 20 #Creates an integer object with value 20 and assigns the variable p to p
q = 20 # Create new reference q which will point to value 20. p & q will be pointing to the same location
r = q # variable r will also point to the same location where p & q are pointing
p, type(p), hex(id(p)) # Variable P is pointing to memory location '0x7fff6d71a3f0'
```

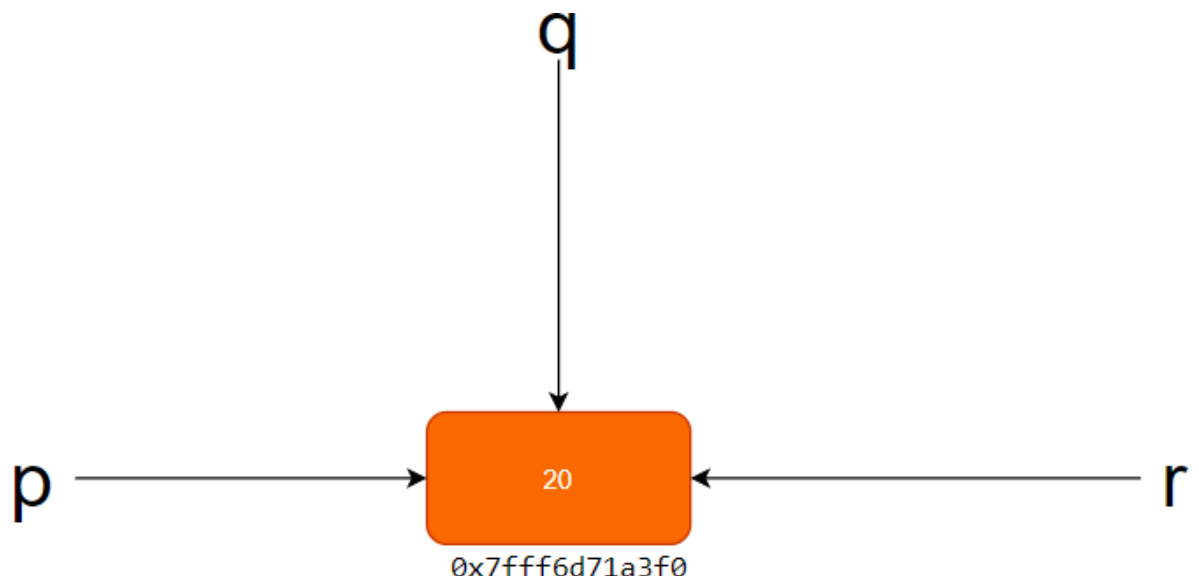
```
Out[94]: (20, int, '0x7fff6d71a3f0')
```

```
In [95]: q, type(q), hex(id(q))
```

```
Out[95]: (20, int, '0x7fff6d71a3f0')
```

```
In [96]: r, type(r), hex(id(r))
```

```
Out[96]: (20, int, '0x7fff6d71a3f0')
```



```
In [146]: p = 20
p = p + 10 # Variable Overwriting
p
```

```
Out[146]: 30
```

Variable Assignment

```
In [100]: intvar = 10 # Integer variable
floatvar = 2.57 # Float Variable
strvar = "Python Language" # String variable

print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Python Language
```

Multiple Assignments

```
In [102]: intvar , floatvar , strvar = 10,2.57,"Python Language" # Using commas to separat
print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Python Language
```

```
In [105]: p1 = p2 = p3 = p4 = 44 # ALL variables pointing to same value
print(p1,p2,p3,p4)
```

```
44 44 44 44
```

Data Types

Numeric

```
In [135]: val1 = 10 # Integer data type
print(val1)
print(type(val1)) # type of object
print(sys.getsizeof(val1)) # size of integer object in bytes
print(val1, " is Integer?", isinstance(val1, int)) # val1 is an instance of int
```

```
10
<class 'int'>
28
10 is Integer? True
```

```
In [126]: val2 = 92.78 # Float data type
print(val2)
print(type(val2)) # type of object
print(sys.getsizeof(val2)) # size of float object in bytes
print(val2, " is float?", isinstance(val2, float)) # Val2 is an instance of floa
```

```
92.78
<class 'float'>
24
92.78 is float? True
```

```
In [136]: val3 = 25 + 10j # Complex data type
print(val3)
print(type(val3)) # type of object
print(sys.getsizeof(val3)) # size of float object in bytes
print(val3, " is complex?", isinstance(val3, complex)) # val3 is an instance of
```

```
(25+10j)
<class 'complex'>
32
(25+10j) is complex? True
```

```
In [119]: sys.getsizeof(int()) # size of integer object in bytes
```

```
Out[119]: 24
```

```
In [120]: sys.getsizeof(float()) # size of float object in bytes
```

```
Out[120]: 24
```

```
In [138]: sys.getsizeof(complex()) # size of complex object in bytes
```

```
Out[138]: 32
```

Boolean

Boolean data type can have only two possible values **true** or **false**.

```
In [139]: bool1 = True
```

```
In [140]: bool2 = False
```

```
In [143]: print(type(bool1))
```

```
<class 'bool'>
```

```
In [144]: print(type(bool2))
```

```
<class 'bool'>
```

```
In [148]: isinstance(bool1, bool)
```

```
Out[148]: True
```

```
In [235]: bool(0)
```

```
Out[235]: False
```

```
In [236]: bool(1)
```

```
Out[236]: True
```

```
In [237]: bool(None)
```

```
Out[237]: False
```

```
In [238]: bool (False)
```

```
Out[238]: False
```

Strings

String Creation

In [193]: `str1 = "HELLO PYTHON"`

```
print(str1)
```

HELLO PYTHON

In [194]: `mystr = 'Hello World' # Define string using single quotes`

```
print(mystr)
```

Hello World

In [195]: `mystr = "Hello World" # Define string using double quotes`

```
print(mystr)
```

Hello World

In [196]: `mystr = '''Hello
World''' # Define string using triple quotes`

```
print(mystr)
```

Hello

World

In [197]: `mystr = """Hello
World""" # Define string using triple quotes`

```
print(mystr)
```

Hello

World

In [198]: `mystr = ('Happy '
 'Monday '
 'Everyone')
print(mystr)`

Happy Monday Everyone

In [199]: `mystr2 = 'Woohoo '
mystr2 = mystr2*5
mystr2`

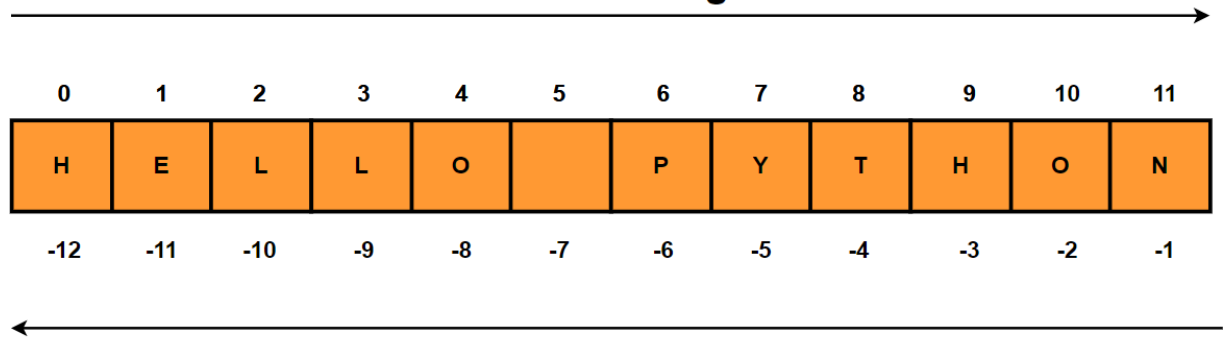
Out[199]: 'Woohoo Woohoo Woohoo Woohoo Woohoo '

In [200]: `len(mystr2) # Length of string`

Out[200]: 35

String Indexing

Forward Indexing



Backward Indexing

In [201]: `str1`

Out[201]: 'HELLO PYTHON'

In [202]: `str1[0]` *# First character in string "str1"*

Out[202]: 'H'

In [203]: `str1[len(str1)-1]` *# Last character in string using len function*

Out[203]: 'N'

In [204]: `str1[-1]` *# Last character in string*

Out[204]: 'N'

In [205]: `str1[6]` *#Fetch 7th element of the string*

Out[205]: 'P'

In [206]: `str1[5]`

Out[206]: ' '

String Slicing

In [207]: `str1[0:5]` *# String slicing - Fetch all characters from 0 to 5 index location exc*

Out[207]: 'HELLO'

In [208]: `str1[6:12]` *# String slicing - Retrieve all characters between 6 - 12 index loc e*

Out[208]: 'PYTHON'

In [209]: `str1[-4:]` *# Retrieve last four characters of the string*

Out[209]: 'THON'


```
In [210]: str1[-6:] # Retrieve last six characters of the string
```

```
Out[210]: 'PYTHON'
```

```
In [211]: str1[:4] # Retrieve first four characters of the string
```

```
Out[211]: 'HELL'
```

```
In [212]: str1[:6] # Retrieve first six characters of the string
```

```
Out[212]: 'HELLO '
```

Update & Delete String

```
In [213]: str1
```

```
Out[213]: 'HELLO PYTHON'
```

```
In [214]: #Strings are immutable which means elements of a string cannot be changed once t  
str1[0:5] = 'HOLAA'
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-214-ea670ff3ec72> in <module>  
      1 #Strings are immutable which means elements of a string cannot be chang  
ed once they have been assigned.  
----> 2 str1[0:5] = 'HOLAA'  
  
TypeError: 'str' object does not support item assignment
```

```
In [215]: del str1 # Delete a string  
print(srt1)
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-215-7fcc0cc83dcc> in <module>  
      1 del str1 # Delete a string  
----> 2 print(srt1)  
  
NameError: name 'srt1' is not defined
```

String concatenation

```
In [216]: # String concatenation  
s1 = "Hello"  
s2 = "Asif"  
s3 = s1 + s2  
print(s3)
```

```
HelloAsif
```