

19th - 20th

Keywords

```
In [2]: import sys
import keyword
import operator
from datetime import datetime
import os
```

```
In [3]: print(keyword.kwlist)
```

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

Identifiers

```
In [4]: 1var = 10
```

```
Cell In[4], line 1
```

```
1var = 10
```

```
^
```

```
SyntaxError: invalid decimal literal
```

```
In [5]: val2@ = 35
```

```
Cell In[5], line 1
```

```
val2@ = 35
```

```
^
```

```
SyntaxError: invalid syntax
```

```
In [6]: import = 125
```

```
Cell In[6], line 1
```

```
import = 125
```

```
^
```

```
SyntaxError: invalid syntax
```

Comments in Python

```
In [7]: # Single line comment
val1 = 10
```

```
In [8]: # Multiple
# line
# comment
val1 = 10
```

```
In [9]: '''
Multiple
line
comment
'''
val1 = 10
```

Statements

```
In [10]: p = 20
q = 20
r = q
p , type(p), hex(id(p)) # Variable P is pointing to memory location '0x7fff6d71a
```

```
Out[10]: (20, int, '0x7fff5d7f2c18')
```

```
In [11]: q , type(q), hex(id(q))
```

```
Out[11]: (20, int, '0x7fff5d7f2c18')
```

```
In [12]: r , type(r), hex(id(r))
```

```
Out[12]: (20, int, '0x7fff5d7f2c18')
```

```
In [13]: p = 20
p = p + 10
p
```

```
Out[13]: 30
```

Variable Assignment

```
In [14]: intvar = 10
floatvar = 2.57
strvar = "Python Language"
print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Python Language
```

Multiple Assignments

```
In [15]: intvar , floatvar , strvar = 10,2.57,"Python Language" # Using commas to separat
print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Python Language
```

```
In [16]: p1 = p2 = p3 = p4 = 44 # All variables pointing to same value
         print(p1,p2,p3,p4)

44 44 44 44
```

Data Types

Numeric

```
In [17]: val1 = 10
         print(val1)
         print(type(val1))
         print(sys.getsizeof(val1))
         print(val1, " is Integer?", isinstance(val1, int))

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

```
In [19]: val2 = 90.35
         print(val2)
         print(type(val2))
         print(sys.getsizeof(val2))
         print(val2, " is float?", isinstance(val2, float))

90.35
<class 'float'>
24
90.35 is float? True
```

```
In [20]: val3 = 25 + 10j
         print(val3)
         print(type(val3))
         print(sys.getsizeof(val3))
         print(val3, " is complex?", isinstance(val3, complex))

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

```
In [21]: sys.getsizeof(int())
```

```
Out[21]: 28
```

```
In [22]: sys.getsizeof(float())
```

```
Out[22]: 24
```

```
In [23]: sys.getsizeof(complex())
```

Out[23]: 32

Boolean

```
In [24]: bool1 = True
```

```
In [25]: bool2 = False
```

```
In [26]: print(type(bool1))
```

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

```
In [27]: print(type(bool2))
```

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

```
In [28]: isinstance(bool1, bool)
```

Out[28]: True

```
In [29]: bool(0)
```

Out[29]: False

```
In [30]: bool(1)
```

Out[30]: True

```
In [31]: bool(None)
```

Out[31]: False

```
In [32]: bool (False)
```

Out[32]: False

Strings

```
In [33]: str1 = "HELLO PYTHON"
print(str1)
```

```
HELLO PYTHON
```

```
In [34]: mystr = 'Hello World' # Define string using single quotes
print(mystr)
```

```
Hello World
```

```
In [35]: mystr = "Hello World" # Define string using double quotes
print(mystr)
```

```
Hello World
```

```
In [36]: mystr = '''Hello
World ''' # Define string using triple quotes
print(mystr)
```

Hello
World

```
In [37]: mystr = """Hello
World""" # Define string using triple quotes
print(mystr)
```

Hello
World

```
In [38]: mystr = ('Happy '
'Monday '
'Everyone')
print(mystr)
```

Happy Monday Everyone

```
In [39]: mystr2 = 'Woohoo '
mystr2 = mystr2*5
mystr2
```

Out[39]: 'Woohoo Woohoo Woohoo Woohoo Woohoo '

```
In [40]: len(mystr2)
```

Out[40]: 35

```
In [41]: str1
```

Out[41]: 'HELLO PYTHON'

```
In [42]: str1[0]
```

Out[42]: 'H'

```
In [43]: str1[len(str1)-1]
```

Out[43]: 'N'

```
In [44]: str1[-1]
```

Out[44]: 'N'

```
In [45]: str1[6]
```

Out[45]: 'P'

String Slicing

```
In [46]: str1[0:5]
```

Out[46]: 'HELLO'

```
In [47]: str1[6:12]
```

```
Out[47]: 'PYTHON'
```

```
In [48]: str1[-4:]
```

```
Out[48]: 'THON'
```

```
In [49]: str1[-5:]
```

```
Out[49]: 'YTHON'
```

```
In [51]: str1[:5]
```

```
Out[51]: 'HELLO'
```

Update & Delete String

```
In [52]: str1
```

```
Out[52]: 'HELLO PYTHON'
```

```
In [53]: str1[0:5] = 'HOLAA'
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[53], line 1  
----> 1 str1[0:5] = 'HOLAA'  
  
TypeError: 'str' object does not support item assignment
```

```
In [54]: del str1 # Delete a string  
         print(str1)
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[54], line 2  
      1 del str1 # Delete a string  
----> 2 print(str1)  
  
NameError: name 'str1' is not defined
```

String concatenation

```
In [55]: s1 = "Hello"  
         s2 = "Asif"  
         s3 = s1 + s2  
         print(s3)
```

```
HelloAsif
```

```
In [ ]:
```