Python toturial

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

Keywords

```
In [21]: print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'clas
s', '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']
```

```
In [23]: len(keyword.kwlist)
```

```
Out[23]: 35
```

Identifiers

```
In [56]: 1var=10
```

```
  Cell In[56], line 1
    1var=10
    ^
SyntaxError: invalid decimal literal
```

```
In [58]: val2@ = 20
```

```
  Cell In[58], line 1
    val2@ = 20
          ^
SyntaxError: invalid syntax
```

```
In [60]: import = 100
```

```
  Cell In[60], line 1
    import = 100
           ^
SyntaxError: invalid syntax
```

```
In [64]: val2=10
         val2
```

```
Out[64]: 10
```

```
In [68]: val_ = 55
         val_
```

```
Out[68]:  55
```

```
In [70]:  VAL1 = 15
          VAL1
```

```
Out[70]:  15
```

Comments in python

```
In [ ]:  val1 = 10

         val1
```

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

```
In [1]:  '''
         Multiple
         line
         comment
         '''
         val1=10
```

```
In [3]:  """
         Multiple
         line
         comment
         """
         val1 = 10
```

```
In [87]:  p = 10
          q = 10
          r = q
          p, type(p), hex(id(p))
```

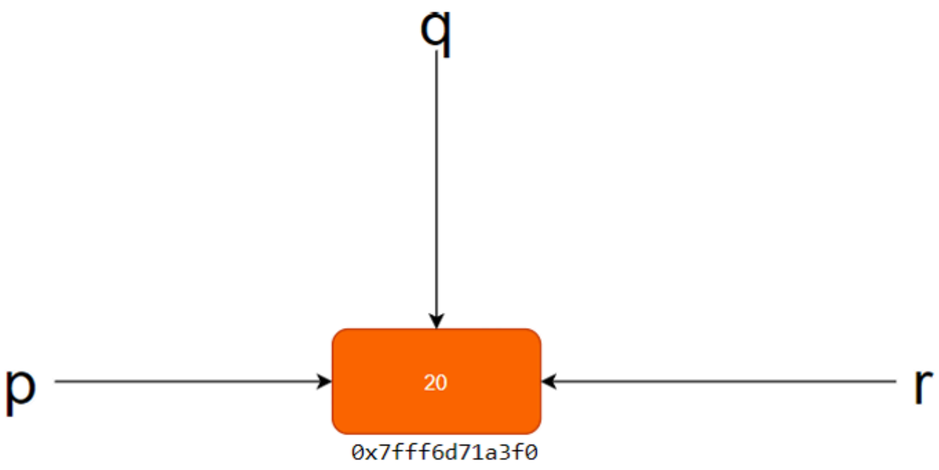
```
Out[87]:  (10, int, '0x7ffed3312ad8')
```

```
In [89]:  q, type(q), hex(id(q))
```

```
Out[89]:  (10, int, '0x7ffed3312ad8')
```

```
In [91]:  r, type(r), hex(id(r))
```

```
Out[91]:  (10, int, '0x7ffed3312ad8')
```



In [7]:
```
p = 20
p = p+10
p
```

Out[7]:
```
30
```

variable assignment

In [12]:
```
intvar = 20
floatvar = 2.50
strvar = "python processor"
```

In [14]:
```
print(intvar)
print(floatvar)
print(strvar)
```

```
20
2.5
python processor
```

Multiple assignments

In [17]:
```
intvar, floatvar, strvar = 15,1.52,"python user"
print(intvar)
print(floatvar)
print(strvar)
```

```
15
1.52
python user
```

In [19]:
```
p1=p2=p3=p4 =99
```

In [21]:
```
print(p1,p2,p3,p4)
```

```
99 99 99 99
```

Datatypes

#Numerics

In [36]:

```
import sys
```

In [38]:

```
val1 = 20
print(val1)
print(type(val1))
print(sys.getsizeof(val1))
print(val1,"is Integer?", isinstance (val1, int))
```

```
20
<class 'int'>
28
20 is Integer? True
```

In [40]:

```
val2 = 15.25
print(val2)
print(type(val2))
print(sys.getsizeof(val2))
print(val1,"is float?", isinstance (val2, float))
```

```
15.25
<class 'float'>
24
20 is float? True
```

In [48]:

```
val3 = 2+5j
print(val3)
print(type(val3))
print(sys.getsizeof(val3))
print(val3,"is compex?", isinstance (val3, complex))
```

```
(2+5j)
<class 'complex'>
32
(2+5j) is compex? True
```

In [52]:

```
sys.getsizeof(int())
```

Out[52]:

```
28
```

In [54]:

```
sys.getsizeof(float())
```

Out[54]:

```
24
```

In [56]:

```
sys.getsizeof(complex())
```

Out[56]:

```
32
```

Boolean

```
In [59]: bool1 = True
```

```
In [61]: bool2 = False
```

```
In [63]: print(type(bool1))
```

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

```
In [65]: print(type(bool2))
```

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

```
In [67]: isinstance(bool1,bool)
```

```
Out[67]: True
```

```
In [69]: bool(0)
```

```
Out[69]: False
```

```
In [77]: bool(1)
```

```
Out[77]: True
```

```
In [73]: bool(None)
```

```
Out[73]: False
```

```
In [75]: bool(False)
```

```
Out[75]: False
```

Strings

string creations

```
In [168…: str1 = "Hello Python"
          print(str1)
```

```
Hello Python
```

```
In [85]: astr = 'Hello world'
         print(astr)
```

```
Hello world
```

```
In [87]: mystr = "Welcome to my world"
         print(mystr)
```

```
Welcome to my world
```

```
In [89]: mystr = '''Hello
                      Devil'''
         print(mystr)
```

```
Hello
            Devil
```

In [91]:

```
mystr = """White
             Devil"""
print(mystr)
```

```
White
            Devil
```

In [97]:

```
mystr = ('Hello '
         'Eveyone '
         'Happy '
         'Monday ')
print(mystr)
```

```
Hello Eveyone Happy Monday
```

In [103…

```
str2 = 'Wow '
str2 = str2*4
print(str2)
```

```
Wow Wow Wow Wow
```

In [105…

```
len(str2)
```

Out[105…

```
16
```

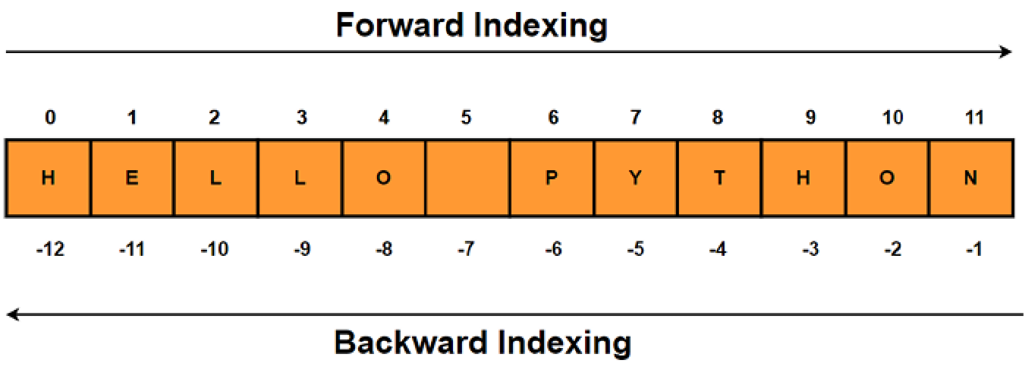
In [107…

```
len(mystr)
```

Out[107…

```
27
```

String Indexing



In [170…

```
str1
```

Out[170…

```
'Hello Python'
```

In [117…

```
str1[0]
```

Out[117…

```
'H'
```

In [121…

```
str1[5]
```

Out[121…

```
' '
```

```
In [123…  str1[len(str1)-1]

Out[123…  'n'

In [125…  str1[-1]

Out[125…  'n'
```

String slicing

```
In [128…  str1[0:5]

Out[128…  'Hello'

In [130…  str1[2:4]

Out[130…  'll'

In [134…  str1[6:12]

Out[134…  'Python'

In [142…  str1[-5:]

Out[142…  'ython'

In [140…  str1[-3:]

Out[140…  'hon'

In [144…  str1[-6:]

Out[144…  'Python'

In [146…  str1[4:]

Out[146…  'o Python'

In [148…  str1[:3]

Out[148…  'Hel'
```

Update and Delete string

```
In [172…  str1

Out[172…  'Hello Python'

In [176…  str1[0:5] = 'Walla'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[176], line 1
----> 1 str1[0:5] = 'Walla'

TypeError: 'str' object does not support item assignment
```

In [180…

```
del str1
print(str1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[180], line 1
----> 1 del str1
      2 print(str1)

NameError: name 'str1' is not defined
```

String concentrartion

In [194…

```
s1 = "Hello "
s2 = "Rohit"
s3 = s1 + s2
print(s3)
```

```
Hello Rohit
```

In [196…

```
s1='Welcome '
s2='to my world'
s3=s1+s2
print(s3)
```

```
Welcome to my world
```

In [ ]: