In [1]:
```
'Hello World'
```

Out[1]:
```
'Hello World'
```

In [3]:
```
import sys
sys.version
```

Out[3]:
```
'3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.1929
64 bit (AMD64)]'
```

In [5]:
```
2+3
```

Out[5]:
```
5
```

In [7]:
```
(3 + 4) - 5
```

Out[7]:
```
2
```

In [9]:
```
#These lines are called variable assignments.
a = 10
b = 5
```

In [23]:
```
#Arithmetic operation (addition).
a + b
```

Out[23]:
```
15
```

In [25]:
```
#Arithmetic operation (substraction).
a - b
```

Out[25]:
```
5
```

In [15]:
```
#Arithmetic operation (multiplication).
a * b
```

Out[15]:
```
50
```

In [17]:
```
#exponentiation or Power to the number
a ** b
```

Out[17]:
```
100000
```

In [19]:
```
#Arithmetic operation (division).
a / b
```

Out[19]:
```
2.0
```

In [21]:
```
#Floor Division or Integer Division (decimal values will be rounded up)
a // b
```

Out[21]:
```
2
```

In [ ]:
```

```

PYTHON IDENTIFIER = PYTHON VARIABLE

In [2]:
```
nit = 3
nit
```

Out[2]: 3

In [4]:
```
#IDENTIFIER DO NOT START WITH A NUMBER
```

In [6]:
```
1nit = 10
1nit
```

```
  Cell In[6], line 1
    1nit = 10
    ^
SyntaxError: invalid decimal literal
```

In [8]:
```
nit1=10
nit1
```

Out[8]: 10

In [10]:
```
#IDENTIFIER DO NOT START WITH A SPECIAL CHARACTER I.E SYMBOL EXCEPT ( "_" ) UNDERS
```

In [12]:
```
nit@ = 63
nit@
```

```
  Cell In[12], line 1
    nit@ = 63
        ^
SyntaxError: invalid syntax
```

In [14]:
```
nit_ = 10
nit_
```

Out[14]: 10

In [16]:
```
_nit = 10
_nit
```

Out[16]: 10

In [18]:
```
import keyword
keyword.kwlist
```

Out[18]:
```
['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']
```

In [20]:
```
len (keyword.kwlist)
```

Out[20]:
```
35
```

In [23]:
```
# key words = reserved words
```

In [ ]:
```
# Identifier is a case sensitive
```

In [25]:
```
a1 = true
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[25], line 1
----> 1 a1 = true

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

In [27]:
```
a1 = True
a1
```

Out[27]:
```
True
```

In [29]:
```
for = 13
for
```

```
  Cell In[29], line 1
    for = 13
        ^
SyntaxError: invalid syntax
```

In [ ]:
```
For = 13
For
```

In [31]:
```
#Python identifier has no limit for character length
```

In [33]:
```
aaaaaaaaaaa = 67
```

In [35]:
```
a
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[35], line 1
----> 1 a

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

In [37]:
```
aaaaaaaaaaa
```

Out[37]: 67

In [39]:
```
True
```

Out[39]: True

In [41]:
```
True + True
```

Out[41]: 2

In [43]:
```
# True value = 1
# False value = 0
```

In [45]:
```
int(True)
```

Out[45]: 1

In [47]:
```
int(False)
```

Out[47]: 0

In [49]:
```
True + False
```

Out[49]: 1

In [51]:
```
False / True
```

Out[51]: 0.0

In [53]:
```
False // True
```

Out[53]:
```
0
```

In [55]:
```
True / False
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[55], line 1
----> 1 True / False

ZeroDivisionError: division by zero
```

In [57]:
```
import numpy as np
```

In [59]:
```
np.zeros(3, dtype = int)
```

Out[59]:
```
array([0, 0, 0])
```

In [61]:
```
np.ones((10,10)) #observe the ones they are float values
```

Out[61]:
```
array([[1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]])
```

In [63]:
```
np.ones((10,10),dtype = int) #observe the ones they are integers
```

Out[63]:
```
array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])
```

In [65]:
```
np.ones((2,3),dtype=int) #there only zeros and ones in numpy
```

Out[65]:
```
array([[1, 1, 1],
       [1, 1, 1]])
```

In [67]:
```
np.arange(2,10)
```

Out[67]:
```
array([2, 3, 4, 5, 6, 7, 8, 9])
```

In [69]:
```
2 + 3
```

Out[69]:
```
5
```

In [71]:
```
int.__add__(2,3)
```

Out[71]: 5

In [73]:
```
int.__sub__(2,3)
```

Out[73]: -1

In [75]:
```
int.__mul__(2,3)
```

Out[75]: 6

In [77]:
```
int.__div__(2,3)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[77], line 1
----> 1 int.__div__(2,3)

AttributeError: type object 'int' has no attribute '__div__'
```

In [ ]:

Python Data Types

In [80]:
```
i = 67
```

In [82]:
```
i
```

Out[82]: 67

In [84]:
```
type(i)
```

Out[84]: int

In [86]:
```
f = 110.67
```

In [88]:
```
f
```

Out[88]: 110.67

In [90]:
```
f1 = 1e0
f1
```

Out[90]: 1.0

In [92]:
```
f2 = 1e1
```

In [94]:
```
f2
```

Out[94]: 10.0

In [96]:
```
f3 = 2e2
f3
```

Out[96]: 200.0

In [98]:
```
f4 = 4m2
f4
```

```
  Cell In[98], line 1
    f4 = 4m2
         ^
SyntaxError: invalid decimal literal
```

In [100]:
```
pi = 3.14
pi
```

Out[100]: 3.14

String

In [103]:
```
s = 'nittech'
s
```

Out[103]: 'nittech'

In [105]:
```
s[:]
```

Out[105]: 'nittech'

In [107]:
```
s[2]
```

Out[107]: 't'

In [109]:
```
print(s[0])
print(s[1])
print(s[2])
print(s[3])
print(s[4])
print(s[5])
print(s[6])
```

```
n
i
t
t
e
c
h
```

In [111]:
```
s1 = "nittechnology"
s1
```

Out[111]: 'nittechnology'

Triple quotes for multi comments or paragraph

In [114]:
```
s2 = '''nittechnology'''
s2
```

Out[114]: 'nittechnology'

In [116]:
```
s3 = 'nittechnology'
s3
```

Out[116]: 'nittechnology'

In [118]:
```
nit = 4
nit
```

Out[118]: 4

In [120]:
```
a = nittech
a
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[120], line 1
----> 1 a = nittech
      2 a

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

Complex

a+bj

In [123]:
```
c = 10 + 20j
c
```

Out[123]: (10+20j)

In [125]:
```
type(c)
```

Out[125]: complex

In [127]:
```
c.real
```

Out[127]: 10.0

In [129]:
```
c.imag
```

Out[129]: 20.0

In [131]:
```
d = 20 + 30j
d
```

Out[131]: (20+30j)

In [133]:
```
(20+30j)
```

Out[133]: (20+30j)

In [135]:
```
print(c)
```

```
print(d)
```

```
(10+20j)
(20+30j)
```

In [137]:
```
e = c + d
e
```

Out[137]: (30+50j)

In [139]:
```
type(e)
```

Out[139]: complex

Bool

true false

In [142]:
```
True
```

Out[142]: True

In [144]:
```
False
```

Out[144]: False

In [146]:
```
int(True)
```

Out[146]: 1

In [148]:
```
int(False)
```

Out[148]: 0

In [150]:
```
True + True
```

Out[150]: 2

In [152]:
```
True + False + True
```

Out[152]: 2

In [154]:
```
True + True * False
```

Out[154]: 1

In [156]:
```
False / True
```

Out[156]: 0.0

In [158]:
```
False // True
```

Out[158]: 0

In [160]:
```
True / False
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[160], line 1
----> 1 True / False

ZeroDivisionError: division by zero
```

In [162]:
```
int(12.3)
```

Out[162]: 12

In [164]:
```
float(12.3)
```

Out[164]: 12.3

In [166]:
```
int(hello)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[166], line 1
----> 1 int(hello)

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

In [168]:
```
int("1")
```

Out[168]: 1

In [170]:
```
i4, f4, c4, s4, b4 = 4, 2.2, 4 + 4j, 'niy', True
```

In [172]:
```
print(type(i4))
```

```
<class 'int'>
```

In [174]:
```
f = 4
type(f)
```

Out[174]: int

In [176]:
```
print (True *2)
```

```
2
```

In [178]:
```
int(12.3)
```

Out[178]: 12

In [180]:
```
i4, f4, c4, s4, b4 = 4, 4.4, 4 + 4j, 'nit', True
```

In [182]:
```
print(type(i4))
```

```
<class 'int'>
```

In [184]:
```
f= 4
type(f)
```

Out[184]: int

In [186]:
```
import numpy as np
```

In [188]:
```
a = np.nan
```

In [190]:
```
type(a)
```

Out[190]: float

In [192]:
```
int(2.3)
```

Out[192]: 2

In [194]:
```
int(2.3, 3.4) #there's decimal values we need another datatype
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[194], line 1
----> 1 int(2.3, 3.4)

TypeError: 'float' object cannot be interpreted as an integer
```

In [196]:
```
int(True)
```

Out[196]: 1

In [198]:
```
int('10')
```

Out[198]: 10

In [200]:
```
int('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[200], line 1
----> 1 int('ten')

ValueError: invalid literal for int() with base 10: 'ten'
```

In [202]:
```
int(1+2j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[202], line 1
----> 1 int(1+2j)

TypeError: int() argument must be a string, a bytes-like object or a real number, n
ot 'complex'
```

Float

float contains decimal values

In [205]:
```
float(2)
```

Out[205]: 2.0

In [207]:
```
float('10')
```

Out[207]: 10.0

In [209]:
```
float(False) #Empty and Zero values are considered to be False
```

Out[209]: 0.0

In [211]:
```
float(10+20j) # float only takes string or int but not complex
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[211], line 1
----> 1 float(10+20j)

TypeError: float() argument must be a string or a real number, not 'complex'
```

Complex

complex is in the format => (a + ij)

In [214]:
```
complex(1)
```

Out[214]: (1+0j)

In [216]:
```
complex(10, 40)
```

Out[216]: (10+40j)

In [218]:
```
complex(2.3)
```

Out[218]: (2.3+0j)

In [220]:
```
complex(2.3, 4.5)
```

Out[220]: (2.3+4.5j)

In [222]:
```
complex(True, False)
```

Out[222]: (1+0j)

In [224]:
```
complex(False, True)
```

Out[224]: 1j

In [226]:
```
complex('10')
```

Out[226]: (10+0j)

In [228]:
```
complex('10', '100') #complex only takes a single string if the first value is str
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[228], line 1
----> 1 complex('10', '100')

TypeError: complex() can't take second arg if first is a string
```

In [230]:
```
complex(10, '100')#heres the same error but vice-versa case
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[230], line 1
----> 1 complex(10, '100')

TypeError: complex() second arg can't be a string
```

In [232]:
```
complex('10')
```

Out[232]: (10+0j)

In [234]:
```
str(1)
```

Out[234]: '1'

In [236]:
```
str(2.3)
```

Out[236]: '2.3'

In [238]:
```
str(True)
```

Out[238]: 'True'

In [240]:
```
str(int(True))
```

Out[240]: '1'

In [242]:
```
str(10+20j)
```

Out[242]: '(10+20j)'

Boolean

boolean => bool => True or False

In [245]:
```
bool(1) # every no zero value is True
```

Out[245]: True

In [247]:
```
bool(2.3)
```

Out[247]: True

In [249]:
```
bool() #Empty and Zero values are False
```

Out[249]: False

In [251]:
```
bool(0)
```

Out[251]: False

In [253]:
```
bool(-1)
```

Out[253]: True

In [255]:
```
bool(10+20j)
```

Out[255]: True

We completedType casting