In [1]:
```
# Constants
>>> print(123) # Numeric Constant
>>> print(98.6)
>>> print('Hello world') # String constants
```

```
123
98.6
Hello world
```

In [7]:
```
# Variables
x = 12.2
print(x)
y = 14
print(y)
```

```
12.2
14
```

In [5]:
```
# You can change the contents of a variable in a later statement
x = 12.2
print(x)
y = 14
print(y)
x = 100
print(x)
```

```
12.2
14
100
```

In [11]:
```
# What is this bit of code doing?
x1q3z9ocd = 35.0 # Variable 1
x1q3z9afd = 12.50 # Variable 2
x1q3p9afd = x1q3z9ocd * x1q3z9afd # Variable 3 = (Variable 1) * (Variable 2)
print(x1q3p9afd) # print(Variable 3)
```

```
437.5
```

In [17]:
```
# Assignment Statements
x = 0.6
x = 3.9 * x * ( 1 - x )
print(x)
x = 3.9 * x * ( 1 - x ) # The Value of x has changed
print(x)
```

```
0.9359999999999999
0.2336256000000002
```

In [38]:
```
# Numeric Expressions
>>> x = 2
>>> x = x + 2 # Addition
>>> print(x)

>>> yy = 440 * 12 # Multiplycation
>>> print(yy)
```

```
>>> zz = yy / 1000 # Division
>>> print(zz)

>>> jj = 23
>>> kk = jj % 5 # Remainder
>>> print(kk)

>>> print(4 ** 3) # Power
```

```
4
5280
5.28
3
64
```

In [52]:
```
# Operator Precedence
x = 1 + 2 * 3 - 4 / 5 ** 6
# = 1 + 2 * 3 - 4 / 3125
# = 1 + 6 - 4 / 3125
# = 1 + 6 - 4 / 3125
# = 1 + 6 - 0.00128
# = 7 - 0.00128
# = 6.99...
print(x)

# Parenthesis
# Power
# Multiplication, Division, and Remainder
# Addition & Subraction
# Left to Righ

x = 1 + 2 ** 3 / 4 * 5
print(x)
```

```
6.999744
11.0
```

In [54]:
```
>>> ddd = 1 + 4 # Addition because the type is Numbers
>>> print(ddd)

>>> eee = 'hello ' + 'there' # Concatenation because the type is String
>>> print(eee)
```

```
5
hello there
```

In [72]:
```
# We can ask Python what type something is by using the type() function.
eee  = 'hello ' + 'there'
type(eee)
```

Out[72]: str

In [68]:
```
type('hello')
```

Out[68]: str

In [70]:
```
type(1)
```

Out[70]: int

In [80]:
```
# Types of Numbers
xx = 1
type (xx)
```

Out[80]: int

In [82]:
```
temp = 98.6
type(temp)
```

Out[82]: float

In [86]:
```
# When you put an integer and floating point in an expression, the integer is impli
# You can Explicitly control this with the built-in functions int() and float()
print(float(99) + 100)
```

```
199.0
```

In [96]:
```
i = 42
type(i)
```

Out[96]: int

In [94]:
```
f = float(i)
print(f)
type(f)
```

```
42.0
```

Out[94]: float

In [98]:
```
# Integer division produces a floating point result
print(10 / 2)
print(9 / 2)
print(99 / 100)
print(10.0 / 2.0)
print(99.0 / 100.0)
```

```
5.0
4.5
0.99
5.0
0.99
```

In [120…
```
# String Conversions
sval = '123'
type(sval)
```

Out[120… str

In [112…
```
print(sval + 1) # Error
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[112], line 1
----> 1 print(sval + 1)

TypeError: can only concatenate str (not "int") to str
```

In [116…
```
ival = int(sval)
type(ival)
```

Out[116… int

In [118…
```
print(ival + 1)
```

```
124
```

In [122…
```
# You will get an error if the string does not contain numeric characters
nsv = 'hello bob'
niv = int(nsv) # Error
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[122], line 3
      1 # You will get an error if the string does not contain numeric characters
      2 nsv = 'hello bob'
----> 3 niv = int(nsv)

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

In [124…
```
# We can instruct Python to pause and read data from the user using the input() fun
# The input() function returns a string
nam = input('Who are you? ')
print('Welcome', nam)
```

```
Welcome Yusra
```

In [126…
```
# Converting User Input
inp = input('Europe floor?')
usf = int(inp) + 1
print('US floor', usf)
```

```
US floor 6
```

In [134…
```
hr = input('Enter Hours: ')
rt = input('Enter Rate: ')
pay = float(hr) * float(rt)
print("Pay: ", pay)
```

```
Pay:  96.25
```

In [ ]: