INTRODUCTION TO PYTHON BASICS

In [27]:

```
# Taking input from the user
num = int(input("Enter a number:"))
print("The number entered by the user: ", num)
type(num)
```

```
The number entered by the user:  15
```

Out[27]:

```
int
```

In [2]:

```
a = 5
type(a)
```

Out[2]:

```
int
```

In [3]:

```
type(float(a))
```

Out[3]:

```
float
```

In [4]:

```
b = 5.5
type(b)
```

Out[4]:

```
float
```

In [5]:

```
type(int(b))
```

Out[5]:

```
int
```

Arithmetic Operators

In [6]:

```
print(a+b)
print(a-b)
print(a*b)
print(a/b)
print(a**b)     # a to the power of b
print(a%b)      # remainder of a/b
```

```
10.5
-0.5
27.5
0.9090909090909091
6987.712429686842
5.0
```

Comparison Operators

In [10]:

```
print(a<b)
print(a>b)
print(a<=b)
print(a>=b)
print(a == b)   # double equality operator
print(a!=b)     # not equal to operator
print(a is not b)   # not equal to operator
```

```
True
False
True
False
False
True
True
```

Logical and Identity Operators

In [21]:

```
# and opeartion
print(True and True)
print(True and False)
print(False and False)
print("\n")

# or opeartion
print(True or True)
print(True or False)
print(False or False)
print("\n")

# not opeartion
print(not True)
print(not False)
```

```
True
False
False


True
True
False


False
True
```

Note:- Preference of Logical Operators:

- not
- and
- or

In [22]:

```
# is, is not are identity operators
print(a is b)       # similar to ==
print(a is not b)   # similar to !=
```

```
False
True
```

In [39]:

```
# in, not in are membership operators
print(5 in [1,2,3,4,5])
print(10 not in [1,2,3,4,5])
```

```
True
True
```

Basic Conversions

In [36]:

```
x = 97
print(chr(x))       # converts integers to characater
#print(ord(x))       # converts single characters to integers(ASCII CODE)
print(oct(x))       # converts integer to octal
print(bin(x))       # converts integer to binary
```

```
a
0o141
0b1100001
```

Decision Making

- if
- elif
- nested if

In [40]:

```
if a<b:
    print("A is less than b")
```

```
A is less than b
```

In [42]:

```
a = 20
if a<b:
    print("A is less than b")
else:
    print("A is greater than b")
```

```
A is greater than b
```

In [43]:

```
if a<b and a<num:
    print("A is smallest")
elif b<a and b<num:
    print("B is smallest")
else:
    print("num is smallest")
```

```
B is smallest
```

Loops/ Iterations

- for
- while
- nested loop

In [45]:

```
count = 1
while(count <=  10):
    print(count)
    count = count + 1
```

```
1
2
3
4
5
6
7
8
9
10
```

In [46]:

```
for i in range(1,11,1):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [54]:

```
import math
print("e to the power of 100.12: ",math.exp(100.12))
print("log of 3 to the base 2: ",math.log(3,2))
print("10 power 10: ",math.pow(10,10))
print("Square root:", math.sqrt(100))
```

```
e to the power of 100.12:  3.0308436140742566e+43
log of 3 to the base 2:  1.5849625007211563
10 power 10:  10000000000.0
Square root: 10.0
```

STRINGS

In [63]:

```
var1 = "Hello World"
var2 = "Python Programming"
print(var1)
print(var2)
print("\n")
print(var1[0])      # accessing first character of var1
print(var1[0:5])    # accessing first word of var1 using slicing operator(:)
print(var2[7:])     # accessing second word of var2 using slicing operator(:)
print(var2[-1])     # acessing last character of var2
```

```
Hello World
Python Programming


H
Hello
Programming
g
```

In [67]:

```
print(var1 + " to" + var2)      # +     -> concatenation
print(var1 * 2)                 # *     -> repetition
print(var1[0])                  #[]     -> slice
print(var1[1:5])                #[:]    -> slice range
```

```
Hello World toPython Programming
Hello WorldHello World
H
ello
```

In [79]:

```
str = "name"
print(str.capitalize())     # capitalizes the 1st letter of string
print(str.endswith("me"))
print(str.isalpha())
print("12345".isalpha())
print(str.isdigit())
print("12345".isnumeric())
print("+".isalnum())
print(str.upper())          # capitalizes entire string
```

```
Name
True
True
False
False
True
False
NAME
```

LISTS

In [81]:

```
list1 = ['physics', 'chemistry', 'mathematics', 'computer']
print(list1)
type(list1)
```

```
['physics', 'chemistry', 'mathematics', 'computer']
```

Out[81]:

```
list
```

In [106]:

```
print(list1[0])    # accessing the first element
print(list1[-1])   # accessing the last element
print(list1[:2])   # accessing the first two elements from the list

list1[1] = 'Chemistry'
print("Updated list: ", list1)

del list1[1]       # to delete a list element from the list
print("After deleting: ",list1)

a = ['p','c',"m",'c']
```

```
physics
computer
['physics', 'computer']
Updated list:  ['physics', 'Chemistry']
After deleting:  ['physics']
```

In [118]:

```
print("Length of the list", len(list1))
print("Maximum element from the list: ", max([1,2.5, 50,10]))
print("Minimum element from the list: ", min([1,2.5, 50,10]))
a.reverse()
print(a)
```

```
Length of the list 1
Maximum element from the list:  50
Minimum element from the list:  1
['c', 'c', 'm', 'p']
```

DICTIONARY

In [122]:

```
dict1 = {"Name": ["Mithun", "Myil", "Sunaina", "Varshini"], "Gender": ["M","M","F","F"]}
print(dict1['Name'])
print(dict1['Gender'])
print(dict1['Name'][0])     # accessing the first element of Name key
```

```
['Mithun', 'Myil', 'Sunaina', 'Varshini']
['M', 'M', 'F', 'F']
```

Out[122]:

```
'Mithun'
```

In [126]:

```
print(len(dict1['Name']))
print(dict1.keys())
print(dict1.values())
```

```
4
dict_keys(['Name', 'Gender'])
dict_values([['Mithun', 'Myil', 'Sunaina', 'Varshini'], ['M', 'M', 'F',
'F']])
```

Date and Time

In [146]:

```
import time
print(time.time())      # returns the number of ticks/seconds since Jan 1,1970 00:00, which
print(time.localtime()) # returns the structure format of epoch
print(time.asctime(time.localtime()))    # returns the local time from the struct format
print(time.process_time())  # Returns the current processor time used in benchmarking tools
print(time.asctime())
time.sleep(5)    # suspends execution for 5 seconds
print(time.asctime())
```

```
1625412444.2684996
time.struct_time(tm_year=2021, tm_mon=7, tm_mday=4, tm_hour=20, tm_min=57, t
m_sec=24, tm_wday=6, tm_yday=185, tm_isdst=0)
Sun Jul  4 20:57:24 2021
9.875
Sun Jul  4 20:57:24 2021
Sun Jul  4 20:57:29 2021
```

In [147]:

```
import calendar
print(calendar.calendar(2021))
```

```
                                  2021

      January                   February                   March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3       1  2  3  4  5  6  7       1  2  3  4  5  6  7
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       8  9 10 11 12 13 14
11 12 13 14 15 16 17      15 16 17 18 19 20 21      15 16 17 18 19 20 21
18 19 20 21 22 23 24      22 23 24 25 26 27 28      22 23 24 25 26 27 28
25 26 27 28 29 30 31                                29 30 31

       April                      May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
          1  2  3  4                      1  2          1  2  3  4  5  6
 5  6  7  8  9 10 11       3  4  5  6  7  8  9       7  8  9 10 11 12 13
12 13 14 15 16 17 18      10 11 12 13 14 15 16      14 15 16 17 18 19 20
19 20 21 22 23 24 25      17 18 19 20 21 22 23      21 22 23 24 25 26 27
26 27 28 29 30            24 25 26 27 28 29 30      28 29 30
                          31

        July                     August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
          1  2  3  4                         1             1  2  3  4  5
 5  6  7  8  9 10 11       2  3  4  5  6  7  8       6  7  8  9 10 11 12
12 13 14 15 16 17 18       9 10 11 12 13 14 15      13 14 15 16 17 18 19
19 20 21 22 23 24 25      16 17 18 19 20 21 22      20 21 22 23 24 25 26
26 27 28 29 30 31         23 24 25 26 27 28 29      27 28 29 30
                          30 31

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3       1  2  3  4  5  6  7             1  2  3  4  5
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       6  7  8  9 10 11 12
11 12 13 14 15 16 17      15 16 17 18 19 20 21      13 14 15 16 17 18 19
18 19 20 21 22 23 24      22 23 24 25 26 27 28      20 21 22 23 24 25 26
25 26 27 28 29 30 31      29 30                     27 28 29 30 31
```

In [151]:

```
print(calendar.isleap(2021))
print(calendar.isleap(2020))
print("\n")
print(calendar.month(2021,7))   # prints calendar of a specific month
```

```
False
True


     July 2021
Mo Tu We Th Fr Sa Su
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

Functions

In [156]:

```
def fib(num):                  # A function to print the fibonacci numbers
    result = []
    a, b = 0, 1
    while b<num:
        result.append(b)
        a,b = b, a+b
    return result
fib(20)
```

Out[156]:

```
[1, 1, 2, 3, 5, 8, 13]
```

The Anonymous Functions

* They are nore declared in the standard manner using *def* keyword

* *lambda* keyword is used to create small anonymous function

* They can take any number of arguments but returns only one value

* They cannot contain suites/ multiple expressions

In [157]:

```
sum = lambda arg1, arg2 :(arg1+arg2)
sum(10,20)
```

Out[157]:

```
30
```

In [159]:

```
import math
content = dir(math)     # returns the sorted lists of strings present in the math module
print(content)
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'ac
osh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',
'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial',
'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose',
'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'l
og2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sq
rt', 'tan', 'tanh', 'tau', 'trunc']
```