Python basics

In [2]:
```
"keerthi"
"bharath"
```

Out[2]:
```
'bharath'
```

In [3]:
```
print("keerthi")
print("bharth")
```

```
keerthi
bharth
```

In [4]:
```
type('ke')
```

Out[4]:
```
str
```

In [5]:
```
print(type('ke'))
```

```
<class 'str'>
```

Data types of variables

In [10]:
```
a=2
print(type(a))
type(a)
```

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

Out[10]:
```
int
```

In [21]:
```
print(type("k"))
type("k")
```

```
<class 'str'>
```

Out[21]:
```
str
```

In [15]:
```
print(type(1.00))
type(1.00)
```

```
<class 'float'>
```

Out[15]:
```
float
```

Type conversion

In [35]:
```
a=10
b=str(a)
c=12.0
print(type(a))
print(type(b))
f="hi"
g=int(f)
print(type(g))
h=float(f)
print(type(h))
```

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

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-35-2aba9f96cdc3> in <module>
      7 #g=int(f)
      8 #print(type(g))
----> 9 h=float(f)
     10 print(type(h))
     11

ValueError: could not convert string to float: 'hi'
```

In [42]:
```
#n1%11
n2=10
n3=n2**12
type(n3)
len(str(n3))
word =10 **75
len(str(word))
print(type(str(word)))
```

```
<class 'str'>
```

python version checking

In [28]:
```
import platform
print(platform.python_version())
print(platform.sys.version)
```

```
3.7.3
3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
```

In [29]:
```
import sys
print(sys.version)
```

```
3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
```

In [53]:
```
#hi
''''print(1234)
print("IIIT")'''
print(10)
```

```
10
```

In [74]:
```
print(12,14,sep="hello",end='hi ')
print("hi")
```

```
12hello14hi hi
```

In [72]:
```
help(print)
```

```
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

In [73]:
```
help(platform)
```

```
Help on module platform:

NAME
    platform

DESCRIPTION
    This module tries to retrieve as much platform-identifying data as
    possible. It makes this information available via function APIs.

    If called from the command line, it prints the platform
    information concatenated as single string to stdout. The output
    format is useable as part of a filename.

CLASSES
    builtins.tuple(builtins.object)
        uname_result

    class uname_result(builtins.tuple)
     |  uname_result(system, node, release, version, machine, processor)
     |
     |  uname_result(system, node, release, version, machine, processor)
     |
     |  Method resolution order:
     |      uname_result
     |      builtins.tuple
     |      builtins.object
     |
     |  Methods defined here:
     |
     |  __getnewargs__(self)
     |      Return self as a plain tuple.  Used by copy and pickle.
     |
     |  __repr__(self)
     |      Return a nicely formatted representation string
     |
     |  _asdict(self)
     |      Return a new OrderedDict which maps field names to their values.
     |
     |  _replace(_self, **kwds)
     |      Return a new uname_result object replacing specified fields with new
values
     |
     |  ----------------------------------------------------------------------
     |  Class methods defined here:
     |
     |  _make(iterable) from builtins.type
     |      Make a new uname_result object from a sequence or iterable
     |
     |  ----------------------------------------------------------------------
     |  Static methods defined here:
     |
     |  __new__(_cls, system, node, release, version, machine, processor)
     |      Create new instance of uname_result(system, node, release, version,
machine, processor)
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors defined here:
     |
     |  system
     |      Alias for field number 0
     |
     |  node
     |      Alias for field number 1
     |
```

BOOLEAN

In [75]:
```
print(bool(0))
print(bool(1))
```

```
False
True
```

In [87]:
```
print(True+True)
print(True+False)
print(False+True)
print(False+False)
```

```
2
1
1
0
```

In [93]:
```
print(3//2)
```

```
1
```

In [94]:
```
print(3/2)
```

```
1.5
```

In [95]:
```
print(3.0//2)
```

```
1.0
```

In [96]:
```
print(3*" hello")
```

```
 hello hello hello
```

In [98]:
```
print(3.0//2.0)
```

```
1.0
```

In [99]:
```
int(34.5)
```

Out[99]:
```
34
```

In [100]:
```
float(22.0)
```

Out[100]:
```
22.0
```

In [101]:
```
str(56)
```

Out[101]:
```
'56'
```

In [103]:
```
bool('')
```

Out[103]:
```
False
```

In [104]:
```
4%2
```

Out[104]:
```
0
```

In [106]:
```
4%2
```

Out[106]:
```
0
```

Operators

In [111]:
```
print(2+3)
print(3-2)
print(3*2)
print(14.0/2)
print(14%2)
print(15.5//2)
```

```
5
1
6
7.0
0
7.0
```

In [115]:
```
print("hi"-'go')##we can't apply airthematic operators on strings except +
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-115-e6e2e89bd991> in <module>
----> 1 print("hi"-'go')##we can't apply airthematic operators on strings except
+

TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

TASK ON OPERATOR

In [118]:
```
orange=3;
apples=9;
c=orange*6+apples*10
print(c//10)#Total multiples of 10
```

```
10
```

In [119]:
```
help(print
    )
```

```
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

In [134]:
```
7//2
```

Out[134]: 3

In [120]:
```
-7+-5
```

Out[120]: -12

In [130]:
```
-7//2
```

Out[130]: -4

STRINGS

In [136]:
```
print("hi\nhello")
```

```
hi
hello
```

In [137]:
```
print("hi\n hello")
```

```
hi
 hello
```

In [144]:
```
s="hi"
print(s)
s
```

```
hi
```

Out[144]: 'hi'

STRING Slicing

In [145]:
```
s="Hello"
s[0]
```

Out[145]: 'H'

In [146]:
```
print(s[0])
```

```
H
```

In [147]:
```
s[0::2]
```

Out[147]: 'Hlo'

In [148]:
```
f="hello world"
```

In [149]:
```
f[0::2]#  skip 1 letter from stating
```

Out[149]: 'hlowrd'

In [151]:
```
f[0::3]
```

Out[151]: 'hlwl'

In [150]:
```
f[::-1]
```

Out[150]:
```
'dlrow olleh'
```

In [152]:
```
p="WELCOME TO PYHTON"
```

In [153]:
```
p.capitalize()
```

Out[153]:
```
'Welcome to pyhton'
```

In [154]:
```
p.lower()
```

Out[154]:
```
'welcome to pyhton'
```

In [155]:
```
p.upper()
```

Out[155]:
```
'WELCOME TO PYHTON'
```

In [156]:
```
p.swapcase()
```

Out[156]:
```
'welcome to pyhton'
```

In [157]:
```
p.title()
```

Out[157]:
```
'Welcome To Pyhton'
```

In [159]:
```
len(p)
```

Out[159]:
```
17
```

In [160]:
```
dir(p)
```

Out[160]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
'capitalize',
'casefold',
'center',
'count',
'encode',
'endswith',
'expandtabs',
'find',
'format',
'format_map',
'index',
'isalnum',
'isalpha',
'isascii',
'isdecimal',
'isdigit',
'isidentifier',
'islower',
'isnumeric',
'isprintable',
'isspace',
'istitle',
'isupper',
'join',
'ljust',
'lower',
'lstrip',
'maketrans',
'partition',
'replace',
'rfind',
```

In [161]:
```
p.casefold()
```

Out[161]:
```
'welcome to pyhton'
```

In [162]:
```
p
```

Out[162]:
```
'WELCOME TO PYHTON'
```

In [164]:
```
help(p.center)
```

```
Help on built-in function center:

center(width, fillchar=' ', /) method of builtins.str instance
    Return a centered string of length width.

    Padding is done using the specified fill character (default is a space).
```

In [165]:
```
p.center()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-165-515483f95060> in <module>
----> 1 p.center()

TypeError: center() takes at least 1 argument (0 given)
```

In [167]:
```
p.center(0)
```

Out[167]:
```
'WELCOME TO PYHTON'
```

In [168]:
```
p.split()
```

Out[168]:
```
['WELCOME', 'TO', 'PYHTON']
```

In [169]:
```
p.count('w')
```

Out[169]:
```
0
```

In [170]:
```
p.count('W')
```

Out[170]:
```
1
```

In [172]:
```
p.count('O')
```

Out[172]:
```
3
```

In [173]:
```
p.encode()
```

Out[173]:
```
b'WELCOME TO PYHTON'
```

Tasks

In [174]:
```
k="Problem Solving Using Python"
```

In [175]:
```
len(k)#print length of string
```

Out[175]:
```
28
```

In [176]:
```
k.lower()#convert to lower case
```

Out[176]:
```
'problem solving using python'
```

In [178]:
```
k[::-1]#print string in reverse order
```

Out[178]:
```
'nohtyP gnisU gnivloS melborP'
```

In [180]:
```
k.split()#split the string
```

Out[180]:
```
['Problem', 'Solving', 'Using', 'Python']
```

In [203]:
```
k.replace(" ","")
```

Out[203]:
```
'ProblemSolvingUsingPython'
```

In [204]:
```
k.count('s')
```

Out[204]:
```
1
```

In [205]:
```
a="python programming by python platforms"
```

In [207]:
```
a.count("python")
```

Out[207]:
```
2
```

tasks

In [3]:
```
# read input form keyboard
a=input("enter some thing")
b=int(input("enter integer"))
print(a)
print(type(a))
print(type(b))
```

```
enter some thinghi
enter integer4
hi
<class 'str'>
<class 'int'>
```

In [7]:
```
e=int(input("enter e"))
h=int(input("enter h"))
print(e+h)
```

```
enter e3
enter h2
5
```

In [15]:
```
e=int(input("enter e"))
h=(input("enter h"))
print(e*h)
```

```
enter e3
enter hhi
hihihi
```

In [47]:
```
g=input("enter first name")
h=input("enter first name")
print(g+" "+h)
print(h+" "+g)
print(h,g,sep=".")
print(str(12))
```

```
enter first namehi
enter first namehello
hi hello
hello hi
hello.hi
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-47-68614a44c071> in <module>
      4 print(h+" "+g)
      5 print(h,g,sep=".")
----> 6 print(str(12))

TypeError: 'str' object is not callable
```

In [44]:
```
number=56
str="String"
number =str(number)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-44-8ba7ffff4e6d> in <module>
      1 number=56
      2 str="String"
----> 3 number =str(number)

TypeError: 'str' object is not callable
```

In [48]:
```

```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-48-951d2a7d61e0> in <module>
----> 1 number(2)

TypeError: 'int' object is not callable
```

In [1]:
```
str(12)
```

Out[1]:
```
'12'
```

In [2]:
```
num=str
num(45)
```

Out[2]:
```
'45'
```

In [1]:
```
int (45.5)
```

Out[1]:
```
45
```

In [ ]:
```

```