In [1]:
```
print("Welcome to second day of TTL lab")
print("Wishing you All a very Hapy Makar Sankrati")
```

```
Welcome to second day of TTL lab
Wishing you All a very Hapy Makar Sankrati
```

In [15]:
```
type(34-90)
```

Out[15]:
```
int
```

In [3]:
```
x=34-90
x
```

Out[3]:
```
-56
```

Datatypes

Check the type of x

PYTHON TYPES

In [4]:
```
print("Hello!")
print("Tarang")
print("Sultania!!")
```

```
Hello!
Tarang
Sultania!!
```

In [5]:
```
## Dispaly 3 senetences in a singleline with space: end is used as separator
```

In [8]:
```
print("Hello!",end="@")
print("Tarang",end="#")
print("Sultania!!")
```

```
Hello!@Tarang#Sultania!!
```

In [9]:
```
print(x)
```

```
-56
```

In [10]:
```
type(x)
```

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

In [11]:
```
a=-89
b=56.34
c="Tools & Technique Lab"
d=False
e=True
f=59-67j
g='T'
```

In [12]:
```
print(type(a))
```

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

In [13]:
```
type(a)
```

Out[13]:
```
int
```

In [14]:
```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
print(type(g))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'bool'>
<class 'complex'>
<class 'str'>
```

In [17]:
```
type('78j+56')
```

Out[17]:
```
str
```

In [19]:
```
roll=input("Enter your roll no: ")
```

```
Enter your roll no: 1234
```

In [20]:
```
print(roll)
```

```
1234
```

In [21]:
```
type(roll)
```

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

type casting

In [22]:
```
roll=int(roll)
type(roll)
```

Out[22]:
```
int
```

In [23]:
```
id=int(input("Enter your student id no: "))
```

```
Enter your student id no: 678
```

In [24]:
```
type(id)
```

Out[24]:
```
int
```

In [25]:
```
name=input("your name please!")
```

```
your name please!Lipika Dewangan
```

In [28]:
```
print("Your T&T teacher name is"+":"+name)
```

```
Your T&T teacher name is:Lipika Dewangan
```

In [32]:
```
roll
type(roll)
```

Out[32]:
```
int
```

In [33]:
```
print("Your T&T teacher emp id is"+str(roll))
```

```
Your T&T teacher emp id is1234
```

In [34]:
```
print("Your T&T teacher emp id is",roll)
```

```
Your T&T teacher emp id is 1234
```

In [36]:
```
sub="Python"
teacher="Lipika"
students=80
day="Monday"
print("I am {},I am teaching {} ".format(teacher,sub,students))
```

```
I am Lipika,I am teaching Python
```

In [37]:
```
## Display your product details using format keyword.
#product name
#item
```

In [38]:
```
marks=input("Your marks please")
```

```
Your marks please56.89
```

In [39]:
```
marks
```

Out[39]:
```
'56.89'
```

In [42]:
```
marks=float(marks)
type(marks)
```

Out[42]:
```
float
```

In [43]:
```
marks
```

Out[43]:
```
56.89
```

In [ ]:
```
#by default input method is str type
#type cast the ouput as per your choice
```

In [54]:
```
product_id="205"
pname="Hp"
item_nos=10
print("product is {n},id {p},total {i}".format(p=product_id,n=pname,i=item_nos))
```

```
product is Hp,id 205,total 10
```

In [55]:
```
print("total no of items are",item_nos)
```

```
total no of items are 10
```

In [57]:
```
print("total no of items are"+str(item_nos))
```

```
total no of items are10
```

Operators

In [58]:
```
25/6
```

Out[58]:
```
4.166666666666667
```

In [59]:
```
25//6  # floor division
```

Out[59]:
```
4
```

In [61]:
```
25%6 #modulo or reminder
```

Out[61]:
```
1
```

In [60]:
```
# single line coment is given by '#' symbol
""" Multiline
comments are given with 3 quotes(single or double quotes) """

'''
IT-1 students are Rocking!
They are curious to learn
And very interactive as well.
'''
```

Out[60]:
```
'\nIT-1 students are Rocking!\nThey are curious to learn\nAnd very interactive as well.\n'
```

In [63]:
```
25**6 #exponential 25 t0 the power
```

Out[63]:
```
244140625
```

In [64]:
```
name
```

Out[64]:
```
'Lipika Dewangan'
```

In [65]:
```
sub="Tools and technique lab"
```

In [66]:
```
#display from 6th to 11th char from sub
#Indexin and Slicing
```

In [67]:
```
sub[6:]
```

Out[67]:
```
'and technique lab'
```

In [68]:
```
sub[6:11]
```

Out[68]:
```
'and t'
```

In [69]:
```
sub[6:12] # start index=6 and end index=12-1
```

Out[69]:
```
'and te'
```

In [70]:
```
sub
```

Out[70]:
```
'Tools and technique lab'
```

In [71]:
```
len(sub)
```

Out[71]:
```
23
```

In [72]:
```
sub[-1]
```

Out[72]:
```
'b'
```

In [73]:
```
sub[-5]
```

Out[73]:
```
'e'
```

In [75]:
```
sub[2]
```

Out[75]:
```
'o'
```

In [76]:
```
# check what are the methods availbale in str method
```

In [79]:
```
dir(str)
```

Out[79]:
```
['__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',
 'removeprefix',
 'removesuffix',
```

```
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [80]:
```
help(str.capitalize)
```

```
Help on method_descriptor:

capitalize(self, /)
    Return a capitalized version of the string.

    More specifically, make the first character have upper case and the rest lower
    case.
```

In [81]:
```
sub.capitalize()
```

Out[81]:
```
'Tools and technique lab'
```

In [82]:
```
sub.upper()
```

Out[82]:
```
'TOOLS AND TECHNIQUE LAB'
```

In [83]:
```
sub.title()
```

Out[83]:
```
'Tools And Technique Lab'
```

In [84]:
```
sub.swapcase()
```

Out[84]:
```
'tOOLS AND TECHNIQUE LAB'
```

In [85]:
```
sub
```

Out[85]:
```
'Tools and technique lab'
```

In [86]:
```
sub[4]='k'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_48844\1760427766.py in <module>
----> 1 sub[4]='k'

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

In [87]:
```
#strings are read only: immutable: content can't be changed
```

In [88]:
```
sub.upper()
```

Out[88]:
```
'TOOLS AND TECHNIQUE LAB'
```

In [89]:
```
sub
```

Out[89]:
```
'Tools and technique lab'
```

In [90]:
```
sub=sub.upper()
sub
```

Out[90]:
```
'TOOLS AND TECHNIQUE LAB'
```

In [95]:
```
sub=sub.replace('O','k')
```

In [96]:
```
sub
```

Out[96]:
```
'TkkLS AND TECHNIQUE LAB'
```

In [113…
```
# square root of a number using math module
import math
dir(math)
```

Out[113]:
```
['__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
 'atan2',
 'atanh',
 'ceil',
 'comb',
 'copysign',
 'cos',
 'cosh',
 'degrees',
 'dist',
 'e',
 'erf',
 'erfc',
 'exp',
 'expm1',
 'fabs',
 'factorial',
 'floor',
 'fmod',
 'frexp',
 'fsum',
 'gamma',
 'gcd',
 'hypot',
 'inf',
 'isclose',
 'isfinite',
 'isinf',
 'isnan',
 'isqrt',
 'lcm',
 'ldexp',
 'lgamma',
 'log',
 'log10',
 'log1p',
 'log2',
 'modf',
 'nan',
 'nextafter',
 'perm',
 'pi',
 'pow',
 'prod',
 'radians',
 'remainder',
 'sin',
 'sinh',
 'sqrt',
 'tan',
 'tanh',
 'tau',
 'trunc',
 'ulp']
```

In [99]:
```
help(math.sqrt)
```

```
Help on built-in function sqrt in module math:

sqrt(x, /)
    Return the square root of x.
```

In [100…
```
no=input("Enter a number..")
```

```
Enter a number..25
```

In [101…
```
no
```

Out[101]:
```
'25'
```

In [104…
```
no=float(no)
```

In [105…
```
no
```

Out[105]:
```
25.0
```

In [110…
```
math.sqrt(no)
```

Out[110]:
```
5.0
```

In [112…
```
math.pi
```

Out[112]:
```
3.141592653589793
```

In [114…
```
topic=["nlp","ml","dl","da"]
a,b,c,d=topic
```

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