abs() — returns the absolute (non-negative) value of a number

In [1]:
```
print(abs(5))
print(abs(-10))
print(abs(3.14))
```

```
5
10
3.14
```

divmod(a, b) : Returns a tuple (quotient, remainder) for integer division a // b and a % b.

Python divides a by b and gives you two results at the same time:

In [2]:
```
print(divmod(21, 3))
print(divmod(29, 7))
print(divmod(19, 2))
```

```
(7, 0)
(4, 1)
(9, 1)
```

Power : Returns raise to value

In [3]:
```
print(pow(3, 4))
print(pow(9, 5))
print(pow(2, 10, 1000))
```

```
81
59049
24
```

bin(x) — converts an integer to its binary string

oct(x) — converts an integer to its octal string

hex(x) — converts an integer to its hexadecimal string

In [4]:
```
print(bin(10))
print(bin(5))
print(bin(25))
print(oct(90))
print(oct(100))
print(oct(150))
print(hex(20))
print(hex(9))
print(hex(120))

```

```
0b1010
0b101
0b11001
0o132
0o144
0o226
0x14
0x9
0x78
```

Int() — converts a value into an integer number.

In [5]:
```
print(int(3.99))
print(int(25.5009982899))
print(int(30.37893986546))

```

```
3
25
30
```

format(x, space) - helps to do formatting integers

In [6]:
```
print(format(255, 'b'))
print(format(3005, 'f'))
print(format(255984682, ','))
```

```
11111111
3005.000000
255,984,682
```

isinstance(object, type) — checks if an object belongs to a specified type (returns True/False).

In [7]:
```
print(isinstance(10, int))
print(isinstance("hello", str))
print(isinstance([1, 2, 3], (list, tuple)))

```

```
True
True
True
```

round(x[, n]) - Round up the value by considering decimal values

In [8]:
```
print(round(18.9999999))
print(round(25.78297488929999999))
print(round(98.9999999))
```

```
19
26
99
```

String Upper - convert string into upper case letters

String Lower - convert string into lower case letters

In [9]:
```
s='my name is chanchal'
print(s.upper())
print(s.lower())
```

```
MY NAME IS CHANCHAL
my name is chanchal
```

String Strip - Removes extra spaces from both ends

lstrip - Removes spaces from left side

rstrip - Removes spaces from right side

In [10]:
```
print("   hello".lstrip())
print("###python".lstrip("#"))
print("data***".rstrip("*"))
```

```
hello
python
data
```

str.split(sep=None, maxsplit=-1)Splits a string into a list based on a separator (default = space); maxsplit controls how many splits happen.

In [11]:
```
print("apple banana mango".split())
print("a,b,c,d".split(","))
print("x:y:z".split(":", 1))
```

```
['apple', 'banana', 'mango']
['a', 'b', 'c', 'd']
['x', 'y:z']
```

str.rsplit(sep=None, maxsplit=-1)-Same as split(), but splitting happens from the right side

In [12]:
```
print("apple banana mango".rsplit())
print("a,b,c,d".rsplit(",", 1))
print("1-2-3-4".rsplit("-", 2))

```

```
['apple', 'banana', 'mango']
['a,b,c', 'd']
['1-2', '3', '4']
```

str.join(iterable) - joins elements of iterable string

In [13]:
```
print(','.join(['ababab','bcbcbc','cdcdcd']))
print(' '.join(['hello','world']))
print(':'.join('123'))
```

```
ababab,bcbcbc,cdcdcd
hello world
1:2:3
```

Str.replace - helps to make changes of alphabets within elements

In [14]:
```
s = 'chanchal'
print(s.replace('a', 'o'))
print(s.replace('an', ''))
print(s.replace('n', 'o', 1))
```

```
chonchol
chchal
chaochal
```

str.find(sub[, start[, end]]) and str.rfind() - Returns the index of the first occurrence of sub; returns -1 if not found.

str.rfind(sub, start=0, end=len(string))-Returns the index of the last

In [15]:
```
s = 'abracadabra'
print(s.find('ra'))
print(s.find('z'))      #
print(s.rfind('ra'))
```

```
2
-1
9
```

str.startswith(prefix) - Checks if the string begins with the given prefix (returns True/False).

str.endswith(suffix) - Checks if the string ends with the given suffix (returns True/False).

In [16]:
```
s = 'hello.chanchal'
print(s.startswith('hel'))
print(s.endswith('.ch'))     #
print(''.startswith('a'))
```

```
True
False
False
```

str.count(sub[, start[, end]])-count number of time alphabets comes

In [17]:
```
s = 'chanchal'
print(s.count('a'))
print(s.count('c'))
print(''.count('n'))
```

```
2
2
0
```

str.isalpha() - Returns True if the string contains only alphabetic letters (A–Z, a–z).

str.isdigit()- Returns True if the string contains only digits (0–9).

str.isalnum()- Returns True if the string contains letters or digits only (no symbols/spaces).

str.isspace()- Returns True if the string contains only whitespace (spaces, tabs, newlines).

In [18]:
```
print('abc'.isalpha())   # True
print('abc123'.isalnum())# True
print('123'.isdigit())   # True
print('\t\n'.isspace())# True
```

```
True
True
True
True
```

str.capitalize() -Make First letters of each word capital

str.title() - Make Title Form,

str.swapcase() - Make upper case to lower case and vv

In [19]:
```
print('hello world'.capitalize())
print('hello world'.title())
print('Hello'.swapcase())
```

```
Hello world
Hello World
hELLO
```

str.zfill(width) - Filled spaces defined with 0

In [33]:
```
print('9'.zfill(3))
print('123'.zfill(5))
print('-7'.zfill(3))
```

```
009
00123
-07
```

str.partition(sep) and str.rpartition(sep)

In [21]:
```
print('a=b=c'.partition('='))
print('a=b=c'.rpartition('='))
```

```
('a', '=', 'b=c')
('a=b', '=', 'c')
```

str.center(width, fillchar=' '), str.ljust(), str.rjust() - Filled extra spaces with define characters

In [22]:
```
print('hello'.center(6))
print('hi'.ljust(6, '-'))
print('hi'.rjust(6, '.'))
```

```
hello
hi----
....hi
```

INPUT

In [23]:
```
name = input("Enter your name: ")
print("Welcome,", name)
print("Type:", type(name))
```

```
Enter your name: Chanchal Bhangale
Welcome, Chanchal Bhangale
Type: <class 'str'>
```

In [24]:
```
sentence = input("Enter a sentence: ")
print(sentence)

```

```
Enter a sentence: My name is chanchal
My name is chanchal
```

In [25]:
```
a, b = input("Enter two words: ").split()
print(a, b)
```

```
Enter two words: Chanchal Bhangale
Chanchal Bhangale
```

In [27]:
```
words = input("Enter words: ").split()
print(words)
```

```
Enter words: Chanchal Bhangale
['Chanchal', 'Bhangale']
```

In [28]:
```
num = int(input("Enter a number: "))
print("You entered:", num)
print("Type:", type(num))
```

```
Enter a number: 9
You entered: 9
Type: <class 'int'>
```

In [29]:
```
a, b = map(int, input("Enter two integers: ").split())
print(a, b)
```

```
Enter two integers: 9 5
9 5
```

In [32]:
```
numbers = list(map(int, input("Enter integers: ").split()))
print(numbers)
```

```
Enter integers: 9 8
[9, 8]
```

int.bit_length() - Returns the number of bits required to represent the integer in binary (excluding the sign).

.to_bytes(length, byteorder, signed=False)

.bit_count() - Returns the number of 1-bits in the number's binary representation.

In [37]:
```
print((10).bit_length())
print((0).bit_length())
print((1023).bit_length())
```

```
4
0
10
```

In [40]:
```
print((1024).to_bytes(2, 'big'))
print((1024).to_bytes(2, 'little'))
print((255).to_bytes(1, 'big'))
```

```
b'\x04\x00'
b'\x00\x04'
b'\xff'
```