Lecture 5

built in functions part 2

Bin function

In [ ]:

```
The bin() function in Python is used to
convert an integer number to its
binary representation as a string prefixed with '0b'.
It takes an integer as an argument
and returns its binary representation.
```

In [2]:

```
binary_rep = bin(10)
print(binary_rep)
```

```
0b1010
```

In [3]:

```
z = bin(200)
print(z)
```

```
0b11001000
```

In [4]:

```
c = bin(46652669462322632)
print(c)
```

```
0b10100101101111100101101100001111001000000001110111001000
```

sum function

In [ ]:

```
The sum() function in Python is used to calculate the sum
of elements in an iterable, such as lists, tuples, and other iterable objects.
```

In [9]:

```
my_list = [7,8,9,5,8,629,2659,652595,5,6,8,2,9,5]
a = sum(my_list)
print(a)
```

```
655955
```

In [10]:
```
str = ['swati','nishant']
print(sum(str))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 2
      1 str = ['swati','nishant']
----> 2 print(sum(str))

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

In [11]:
```
a = "swati"
b = "nishant"
print(a+b)
```

```
swatinishant
```

In [12]:
```
a = [1,2,3,4,5]
starting_value = 10
total = sum(a,starting_value)
print(total)
```

```
25
```

In [13]:
```
a = [1,2,3,4,5]
starting_value = 2
total = sum(a,starting_value)
print(total)
```

```
17
```

Eval function

In [ ]:
```
eval(): Evaluates a Python expression stored in a string
```

In [14]:
```
x = 10
y = 5
expression = "x+y*2"
print(eval(expression))
```

```
20
```

In [16]:
```
exp = " 5/2 +3.5"
print(eval(exp))
print(type(exp))
```

```
6.0
<class 'str'>
```

In [17]:

```
exp1 = " 234*25+63-98/2"
print(eval(exp1))
```

```
5864.0
```

In [19]:

```
x = 5
y = 2.5
z = 32
exp = "x+y-z*23/5-9*3"
print(eval(exp))
```

```
-166.7
```

Help function

In [ ]:

```
The Python help function is used to display the
documentation of modules, functions, classes, keywords, etc
```

In [20]:

```
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 [21]:

```
help(int)
```

```
 |          bytes and bytearray are examples of built-in objects that support
the
 |          buffer protocol.
 |        byteorder
 |          The byte order used to represent the integer.  If byteorder is 'b
ig',
 |          the most significant byte is at the beginning of the byte array.
If
 |          byteorder is 'little', the most significant byte is at the end of
the
 |          byte array.  To request the native byte order of the host system,
use
 |          `sys.byteorder' as the byte order value.
 |        signed
 |          Indicates whether two's complement is used to represent the integ
er.
 |
 |    ----------------------------------------------------------------------
 |    Static methods defined here:
 |
 |        __new__(*args, **kwargs) from builtins.type
```

In [22]:

```
help(float)
```

```
 |        Return divmod(value, self).
 |
 |    __repr__(self, /)
 |        Return repr(self).
 |
 |    __rfloordiv__(self, value, /)
 |        Return value//self.
 |
 |    __rmod__(self, value, /)
 |        Return value%self.
 |
 |    __rmul__(self, value, /)
 |        Return value*self.
 |
 |    __round__(self, ndigits=None, /)
 |        Return the Integral closest to x, rounding half toward even.
 |
 |        When an argument is passed, work like built-in round(x, ndigits).
 |
 |    __rpow__(self, value, mod=None, /)
```

In [23]:
```
help(sum)
```

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

sum(iterable, /, start=0)
    Return the sum of a 'start' value (default: 0) plus an iterable of number
s

    When the iterable is empty, return the start value.
    This function is intended specifically for use with numeric values and ma
y
    reject non-numeric types.
```

In [24]:
```
sum??
```

input function

In [28]:
```
name = input("enter your name")
print("name entered by user is", name)
print(type(name))
```

```
enter your nameswati
name entered by user is swati
<class 'str'>
```

In [30]:
```
age = input("enter your age pls")
print("age of user is",age)
print(type(age))
```

```
enter your age pls30
age of user is 30
<class 'str'>
```

Practice Questions

Question 1

In [ ]:
```
Get two numbers from the user and calculate their sum
```

In [32]:
```
num1 = int(input("enter first number"))
num2 = int(input("enter second number"))
sum = num1 + num2
print(sum)
```

```
enter first number3
enter second number5
8
```

Question 2

In [ ]:

```
As part of your role in a data-driven project,
you need to take input from the user for their
age, weight, and height. However, there's an issue:
the user has entered their age as a negative number.
You need to correct the age of the user, and after
correcting the age, calculate their Basal Metabolic Rate (BMR)
using the Harris-Benedict equation for men.
Ensure that the final answer of BMR is an integer.

Basal Metabolic Rate (BMR):
Formula (Harris-Benedict equation for men):
BMR = 88.362 + (13.397 * weight in kg) +
(4.799 * height in cm) - (5.677 * age in years)
```

In [38]:

```
age = int(input("enter your age"))
weight = int(input("enter your weight"))
height = float(input("enter your height"))
correct_age = abs(age)
print("the correct age of user is", correct_age)
BMR = "88.362 + (13.397 * weight ) + (4.799 * height) - (5.677 * age )"
bmr = eval(BMR)
print("the BMR of the patient is", int(bmr))
```

```
enter your age-45
enter your weight85
enter your height5.6
the correct age of user is 45
the BMR of the patient is 1509
```

homework

Question 3

In [ ]:

```
write a python code for hostel students,  ask them to mention thier course name
for 5 students and calculate thier avg age.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: