Programming with Python - Open Elective

Feb 28, 2025

Revision: variables (containers for data), input(), print(),
datatype: int, float, bool, str, None
arithmetic, relational and logical operator, operator associativity, operator precedence (arithmetic > relational > logical)
if-else, for loop, while loop,
strings, string indexing, string slicing

Strings: Sequence of characters enclosed within quotes (single, double or triple quotes).

```
msg = 'Hello World'
print(msg[0])
print(msg[-1])
print(msg[:8:2])
print(msg[7:1:-1])
print(msg[::-1)
print(msg[-1:-5]
```

Membership

```
'H' in 'Hello' #True
#space separated string
st = ''
for c in 'python':
    st = st + c + ' '
```

#Count hashtags in a post Or display hashtags in a post.

Built-in Functions on Strings: Some common built-in string methods are listed below with examples for a few of them. Explore python string documentation for details.

```
len('Hello World') #length of a string
11
```

max/min

```
max('AB', 'BC', 'AC')
min('ABCD', 'BCD, 'CA')
```

find/rfind

```
colors = 'green, red, blue, red'
colors.find('red') #1, first location, -1 if not found
colors.rfind('red') #3
```

lower, upper

```
'heLLo'.lower()
'heLLo'.upper()
```

replace

```
'abcdefab'.replace('ab', 'aa') #'aacdefab'
```

strip

```
'     hello world      '.strip() #'hello world
```

split

```
colors ='red, green, blue'
colors.split(',') #['red', ' green', ' blue']
```

join

```
'+'.join('abc') #'a+b+c'
```

startswith/endswith

```
'hellooooo'.startswith('he') #True
```

Function:

- A function is a reusable block of code that performs a specific task.
- Functions make code modular, readable, and reusable.
- Functions can accept 0 or more arguments/parameters. A function may or may not have a return statement.

Basic Syntax

```
def function_name():
    <statements>
```

Example:

```
def greet():
    print("Hello! Welcome to Python.")

greet()  #function call.
```

Function parameter and return value:

- **Finding GC content in a DNA sequence**

```
def gc_content(dna):
    g_count = dna.count("G")
    c_count = dna.count("C")
    return (g_count + c_count) / len(dna) * 100

print(gc_content("ATGCGTACGTAGCGT"))
```

Calculating percentage profit

```
def profit_percentage(cost_price, selling_price):
    profit = selling_price - cost_price
    return (profit / cost_price) * 100

print(profit_percentage(200, 250))
```

Simulating bacterial growth

```
def bacterial_growth(initial_population, hours):
    population = initial_population
    for i in range(hours):
        population *= 2
    return population

print(bacterial_growth(10, 5))
```

Checking prime numbers:

```
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

print(is_prime(11))
print(is_prime(15))
```

Exercises:

1. Write a function that takes two strings and returns `True` if they are anagrams and `False` otherwise. A pair of strings is anagrams if the letters in one word can be arranged to form the second one.

2. Write a program to remove all vowels from a given string.

3. Write a program to reverse the order of words in a sentence.

4. Write a program to extract and print all digits from a given string.

5. Write a function that takes a sentence as input parameter and returns the number of words in the sentence.

6. Write a program to display the most frequently occurring character in a string.

7. Write a program to:
    a. Remove all spaces from a given string.
    b. Check if a given substring is present in a string.
    c. Convert the first letter of each word in a sentence to uppercase.

8. Count uppercase and lowercase letters:
9. Remove digits from a string.