# Programming with Python - Open Elective

Feb 21, 2025

**Revision:** variables (containers for data), input(), print(),
datatype: int, float, bool, str
arithmetic, relational and logical operator, operator associativity, operator precedence (arithmetic > relational > logical)
if-else, for loop, while loop

Loops: for loop

```
    for variable in sequence:
        Statements
```

while loop

```
    while condition:
        Statements
```

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

~~~
>> msg = ‘Hello World’
>> msg = “Hello World”
>> msg = ```Hello
             World```
~~~

- Triple quotes are typically used for stings spanning multiple lines.
- string containing single quotes can be enclosed with double quotes and vice-versa. E.g. `“It’s a sunny day!”`

```
>>len(‘Hello World’) #length of a string
11
```

**Indexing**: indices start from 0 (left or right) or -1 (from right to left)

```
>> msg = ‘hello world’
>> msg[0]
```

```
>> msg[-1]
>> msg[20] #string out of range, not a problem when
slicing
>> msg[-15] #string out of range
```

Strings in Python are **immutable**(Assignment to part of string gives error)

```
>> msg[1] = ‘H’ #error
>> msg = “Bye world” #why this doesn’t give errors.
```

**String concatenation(+) and Repetition(*)**

```
‘Hello’ + ‘world’
‘Hello’ * 3
```

**max/min**

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

**Slicing [start:end:stepsize]**

Forward (0:) and backward indexing(:-1)

```
msg = ‘hello world’
msg[0: 4]
msg[:4]
msg[2:5]
msg[-1]
msg[-1:-5]
msg[::-1] #string reverse
```

**Membership**

```
‘H’ in ‘Hello’ #True
```

**Example**: Write a program to insert space among characters in a give word.

```
st = ‘’
for c in ‘python’:
    st = st + c + ‘ ‘
print(st)
```

Output: ‘p y t h o n’

## Exercises:

1. Write a program to count the number of vowels in a given string. (Example: ‘Good day’ has 3 occurrences of vowel)

2. Write a program to count the unique vowels in a string. (Example: ‘Good day’ has 2 unique vowels)

3. Write a program that counts the number of vowels (a, e, i, o, u) in a given sentence (sentence to be input by the user).

4. Write a program to check if a word is a palindrome or not.

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

Here’s a table with common Python string functions and their explanations:

| **`Function`** | **Explanation** |
|---|---|
| **`str.lower()`** | Converts all characters in a string to lowercase. |
| `str.upper()` | Converts all characters in a string to uppercase. |
| `str.title()` | Converts the first character of each word to uppercase. |
| `str.capitalize()` | Capitalizes the first character of the string. |
| **`str.strip()`** | Removes leading and trailing whitespace. |
| `str.lstrip()` | Removes leading whitespace. |
| `str.rstrip()` | Removes trailing whitespace. |
| `str.replace(old, new)` | Replaces all occurrences of `old` with `new`. |
| **`str.find(sub)`** | Returns the index of the first occurrence of `sub`, or -1 if not found. |

| | |
|---|---|
| `str.index(sub)` | Similar to `find()`, but raises an error if `sub` is not found. |
| `str.count(sub)` | Returns the number of times `sub` appears in the string. |
| **`str.split(sep)`** | Splits the string into a list of substrings based on `sep`. |
| **`str.join(iterable)`** | Joins elements of `iterable` into a single string, using the string as a separator. |
| `str.startswith(prefix)` | Returns `True` if the string starts with `prefix`. |
| `str.endswith(suffix)` | Returns `True` if the string ends with `suffix`. |
| `str.isdigit()` | Returns `True` if the string contains only digits. |
| `str.isalpha()` | Returns `True` if the string contains only alphabetic characters. |
| `str.isalnum()` | Returns `True` if the string contains only alphanumeric characters. |
| `str.isspace()` | Returns `True` if the string contains only whitespace. |
| `str.zfill(width)` | Pads the string with zeros on the left to match `width`. |