

Python

**Class 4**

# Introduction to Python

Azmain Adel

August 18, 2024

**01.**

Review of Previous Class



# Review Topics

- List methods
- Functions

# Solution to Problem 1

Write a program to find sum of all numbers in a list.

```
def sum_list(numbers):  
    total = 0  
    for num in numbers:  
        total += num  
    return total  
  
# Example usage  
my_list = [1, 2, 3, 4, 5]  
result = sum_list(my_list)  
print(f"The sum of the numbers in the list is: {result}")
```

## Solution to Problem 2

Write a Python function to print the maximum of three numbers.

```
def print_max_of_three(a, b, c):  
    max_num = a  
    if b > max_num:  
        max_num = b  
    if c > max_num:  
        max_num = c  
    print(f"The maximum of {a}, {b}, and {c} is: {max_num}")  
  
# Example usage  
print_max_of_three(10, 30, 20)
```

## Solution to Problem 2

Write a Python function to check whether a number falls within the range (1,100). Print "Yes" or "No"

```
def check_range(number):  
    if 1 <= number <= 100:  
        print("Yes")  
    else:  
        print("No")
```

```
# Example usage  
check_range(50)
```

02.

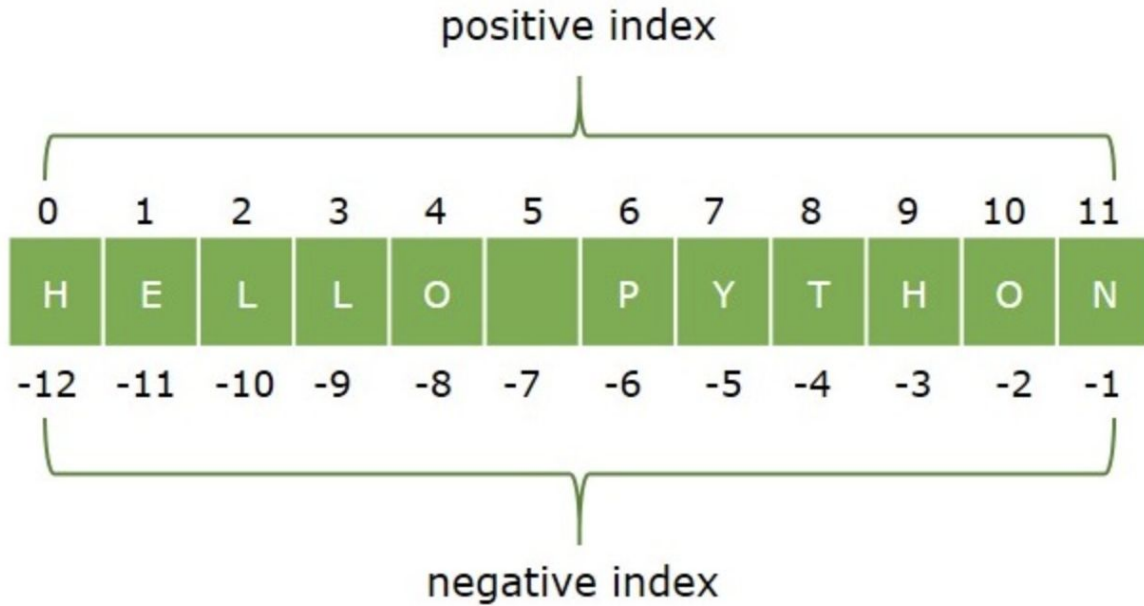
String

# Strings in Python

- A string is an **immutable sequence** of Unicode characters.
- Enclosed in single or double or triple quotes.



# String Slicing



# Slicing

```
var = "HELLO PYTHON"
```

```
print(var[2])      # L
```

```
print(var[-12])   # H
```

```
print(var[3:8])   # LO PY
```

## String Modification (List to String)

```
s1="WORD"
print ("original string:", s1)
l1=list(s1)

l1.insert(3,"L")

print (l1)

s1=''.join(l1)
print ("Modified string:", s1)
```

# String Concatenation

```
new_s = s1 + s2
```

# String Formatting

1. Using **%** operator
2. Using **format()** method of str class
3. Using **f-string**

# String Formatting

```
name = "Tutorialspoint"  
print("Welcome to %s!" % name)
```

```
str = "Welcome to {}"  
print(str.format("Tutorialspoint"))
```

```
item1_price = 2500  
item2_price = 300  
total = f'Total: {item1_price + item2_price}'  
print(total)
```

## Some String Methods

1. `capitalize()`
2. `lower()`
3. `upper()`
4. `strip()`, `lstrip()`, `rstring()`
5. `split(element)`
6. `join()`
7. `isnumeric()`
8. `isdigit()`
9. `replace(old_sub_str, new_sub_str)`
10. `count(sub_str, start, end)`

03.

Set



# Sets in Python

- A set is an unordered collection of **unique** elements.
- Does not allow duplicates.
- Defined inside **{ }** braces or **set()**

```
my_set = {1, 2, 3}
```

# Set Operations

- Add: `my_set.add(4)`
- Remove: `my_set.remove(4)`
- Clear: `my_set.clear()`
- Update: `my_set.update([5, 6])`
- Union: `set_1.union(set_2)` or `set_1 | set_2`
- Looping: `for item in my_set:`

04.

Tuple

# Tuples in Python

- A tuple is a **sequence of comma separated items**
- Enclosed in parentheses ( )
- **Immutable**. Cannot be changed, removed, added.
- Modifications can be by done by converting to list first.

```
my_tup = (1, 2, 3)
```

# Tuple Operations

- Access: `tup[3]`
- Count of an item: `tup.count(4)`
- Index of an item: `tup.index(4)`
- Join: `tup_1 + tup_2`
- Looping: `for item in tup:`

**05.**

Recap and Q&A



Open floor for questions and clarifications

06.

To-do at Home





# Problem to Solve 1

Python program to find number of vowels in a given string.

## Problem to Solve 2

Python program to list unique characters with their count in a string

Input: "Hello"

Output: h=1, e=1, l=2, o=1

## Problem to Solve 3

Program to find common elements in two lists with the help of set operations

Input:

$l1 = [1, 2, 3]$

$l2 = [3, 4]$

Output:  $[3]$



Script review in the next class

**Thank you.**

[azmainadel47@gmail.com](mailto:azmainadel47@gmail.com)  
[linkedin.com/in/azmainadel](https://www.linkedin.com/in/azmainadel)  
[azmainadel.com](http://azmainadel.com)