

## Python Basics

### 1. Dictionary

- A dictionary in Python is an unordered collection of items. Each item is a key-value pair, where the key must be unique, and the value can be any data type.
- Represented by {}
- Not in order
- Example:

```
q = {'tomato': 5, 1: 3, 4: 'potato', 'tortoise': 'rabbit'}
```

### 2. Set

- A set is an unordered collection of unique elements. Unlike lists, sets automatically remove duplicate entries.
- Example:

```
A = [1, 1, 1, 1, 1, 2, 3, 4, 4, 4, 4]
b = set(A)
# b = {1, 2, 3, 4}
```

### 3. Tuple

- Uses circular brackets ()
- A tuple is an ordered collection of elements enclosed in parentheses.
- Unlike lists, tuples are immutable (can't modify with append, remove, delete)
- Example:

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

## Loop - Iteration

### 1. For Loop

- A for loop is used to iterate over a sequence (e.g., list, tuple, dictionary, set, or string) in Python.

```

# Example 1
for x in A:
    print(x)

# Example 2
for x in range(5, 11):
    print(x)
# Output: 5, 6, 7, 8, 9, 10

# Multiplication table example
for x in range(0, 10):
    for y in range(0, 10):
        print(f"{x} X {y} = {x*y}")
    print("\n")

# Example of continue and break
x = 10
for i in range(1, 100):
    if i % x == 0:
        continue
    else:
        print(i)

for i in range(1, 10):
    if i % x == 0:
        break
    else:
        print(i)

```

## 2. While Loop

```

x = 50
while True:
    print(x)
    x -= 1
    if x < 20:
        break
# Output:
# 50
# 49
# 48
# ...
# 21

x = 50
while x >= 20:

```

```

    print(x)
    x -= 1

# Output:
# 50
# 49
# 48
# ...
# 20

```

## Functions

- Piece of code similar to loop
- Built-in functions: `print()`, `len()`, `split()`, `join()`

```

def avg(x, y, z):
    return (x + y + z) / 3

```

*# Usage: avg(10, 20, 30) returns 20.0*

```

def func1(x, y, z):
    a = x + y + z

```

*# Usage: func1("cat", "dog", "cat") # Output: catdogcat*

## Recursion

- A function calls itself to solve a smaller instance of the same problem.

```

# Fibonacci series
def fib1(x, y):
    print(x + y)
    fib1(y, x + y)

def fib(x, y):
    print(x)
    print(y)
    fib1(x, y)

```

*# Output:*

```

# 1
# 1
# 2
# 3
# 5
# 8
# 13

```

```
# 21
# 34
# 55
#...
#...
#Ctrl+C : To interrupt the loop
```

#### Note:

- Iteration consumes computation time, while recursion consumes memory and may be less efficient for large problems

## Installation

- pip (Python package installer): `pip` is the package installer for Python. You can use it to install packages from the Python Package Index and other indexes.
- To install a package using pip: `pip install package_name`
- For example, to install NLTK: `pip install nltk`
- Reference for NLTK (Read Chapter 0-6)
  - After installing NLTK, you may need to download additional data:

```
import nltk
nltk.download()
```

## Introduction to NLP (Natural Language Processing)

- Subarea of ML, focused on generating and understanding language through computers

#### Key Concepts:

1. **POS (Part-of-Speech) Tagging**
  - Labeling each word in a sentence with its corresponding part of speech (e.g., noun, verb, adjective)
  - Classification task (Supervised ML)
  - Uses training data with annotated POS tags
  - Dependent on previous words
2. **Tokenization**
  - Splitting text into individual words or sentences
3. **Segmentation**
  - Segmentation divides a text into sentences or larger chunks. This can be tricky due to abbreviations and other punctuation issues. As full stops can be end of sentence or abbreviations.
4. **NER (Named Entity Recognition)**
  - NER is the process of identifying and classifying entities (e.g., names of people, organizations, locations) in text.

- Example: *Zanzeer is an interesting movie.*

NER breakdown: *Zanzeer* - (Movie title) is a named entity *interesting* - Not a named entity *movie* - Not a named entity

In this example, **Zanzeer** is recognized as a named entity, specifically a work of art (movie title)

#### 5. **Tagging**

- POS, NER, Relation Tagging

#### 6. **Coreference Resolution**

- Identifying subphrases that refer to the same entity in the text.
- Example:

"John went to the store. He bought milk." → "He" refers to "John"