No More Confusion in

- Python List
- Python Tuple
- Python String
- Python Set
- Python Dictionary

SWIPE >>>

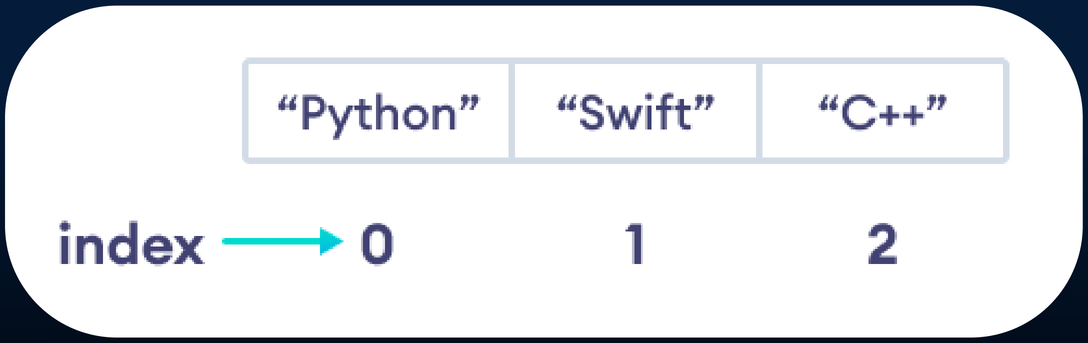
LISTS

★ A list is a collection of similar or different types of data.

★ A list is created in Python by placing items inside [], separated by commas.

★ A list can have any number of items and they may be of different types (integer, float, string, etc.).

★ A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements.



SWIPE >>>

Method	Description
append()	add an item to the end of the list
extend()	add items of lists and other iterables to the end of the list
insert()	inserts an item at the specified index
remove()	removes item present at the given index
pop()	returns and removes item present at the given index
clear()	removes all items from the list
index()	returns the index of the first matched item
count()	returns the count of the specified item in the list
sort()	sort the list in ascending/descending order
reverse()	reverses the item of the list
copy()	returns the shallow copy of the list

SWIPE >>>

TUPLE

★ A tuple is created by placing all the items (elements) inside parentheses (), separated by commas.

★ A tuple can have any number of items and they may be of different types (integer, float, list, string, etc.).

★ Since tuples are immutable, iterating through a tuple is faster than with a list. So there is a slight performance boost.

```
my_tuple = ('a', 'p', 'p', 'l', 'e',)

print(my_tuple.count('p'))  # prints 2
print(my_tuple.index('l'))  # prints 3
```

SWIPE >>>

String

- In computer programming, a string is a sequence of characters.
- For example, "hello" is a string containing a sequence of characters 'h', 'e', 'l', 'l', and 'o'.
- Python strings are "immutable" which means they cannot be changed after they are created

```
# create string type variables

name = "Vaishnavi Pandey"   #output:- Vaishnavi Pandey
print(name)
message = "Vaishnavi loves Python"  #output:- Vaishnavi
print(message)                                loves python
```

SWIPE >>>

Methods	Description
upper()	converts the string to uppercase
lower()	converts the string to lowercase
partition()	returns a tuple
replace()	replaces substring inside
find()	returns the index of first occurrence of substring
rstrip()	removes trailing characters
split()	splits string from left
startswith()	checks if string starts with the specified string
isnumeric()	checks numeric characters
index()	returns index of substring

SWIPE >>>

SET

- ★ A set is a collection of unique data. That is, elements of a set cannot be duplicate.
- ★ In Python, we create sets by placing all the elements inside curly braces {}, separated by comma
- ★ A set can have any number of items and they may be of different types (integer, float, tuple, string etc.).
- ★ But a set cannot have mutable elements like lists, sets or dictionaries as its elements.
- ★ Sets are mutable. However, since they are unordered, indexing has no meaning.

```
# create a set of integer type
student_id = {12, 14, 16, 118, 115}
print('Student ID:', student_id)
#output:-
Student ID: {12, 14, 115, 16, 118}
```

SWIPE >>>

Built-in Functions with Set

Function	Description
all()	Returns True if all elements of the set are true (or if the set is empty).
any()	Returns True if any element of the set is true. If the set is empty, returns False.
enumerate()	Returns an enumerate object. It contains the index and value for all the items of the set as a pair.
len()	Returns the length (the number of items) in the set.
max()	Returns the largest item in the set.
min()	Returns the smallest item in the set.
sorted()	Returns a new sorted list from elements in the set(does not sort the set itself).
sum()	Returns the sum of all elements in the set.

SWIPE >>>

Method	Description
add()	Adds an element to the set
clear()	Removes all elements from the set
copy()	Returns a copy of the set
difference()	Returns the difference of two or more sets as a new set
difference_update()	Removes all elements of another set from this set
discard()	Removes an element from the set if it is a member. (Do nothing if the element is not in set)
intersection()	Returns the intersection of two sets as a new set
intersection_update()	Updates the set with the intersection of itself and another
isdisjoint()	Returns True if two sets have a null intersection
issubset()	Returns True if another set contains this set
issuperset()	Returns True if this set contains another set

SWIPE >>>

pop()	Removes and returns an arbitrary set element. Raises KeyError if the set is empty
remove()	Removes an element from the set. If the element is not member, raises a KeyError
symmetric_difference()	Returns the symmetric difference of two sets as a new set
symmetric_difference_update()	Updates a set with the symmetric difference of itself and another
union()	Returns the union of sets in a new set
update()	Updates the set with the union of itself and others

SWIPE »»

DICTIONARY

- Python dictionary is an ordered collection (starting from Python 3.7) of items.
- It stores elements in key/value pairs. Here, keys are unique identifiers that are associated with each value.
- We can also have keys and values of different data types.
- It is mutable in nature. so entries can be added, removed, and changed at any time. Note, though, that because entries are accessed by their key, we can't have two entries with the same key.

```
capital_city = {"Nepal": "Kathmandu", "Italy": "Rome",
"England": "London"}
print(capital_city)
Output:-
{'Nepal': 'Kathmandu', 'Italy': 'Rome', 'England':
'London'}
```

SWIPE >>>

Method	Description
clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and value
get()	Returns the value of the specified key
items()	Returns a list containing a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
setdefault()	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

SWIPE >>>

If you really like my content
Please don't forget to like,comment
and share

Thank you!

in Vaishnavi Pandey