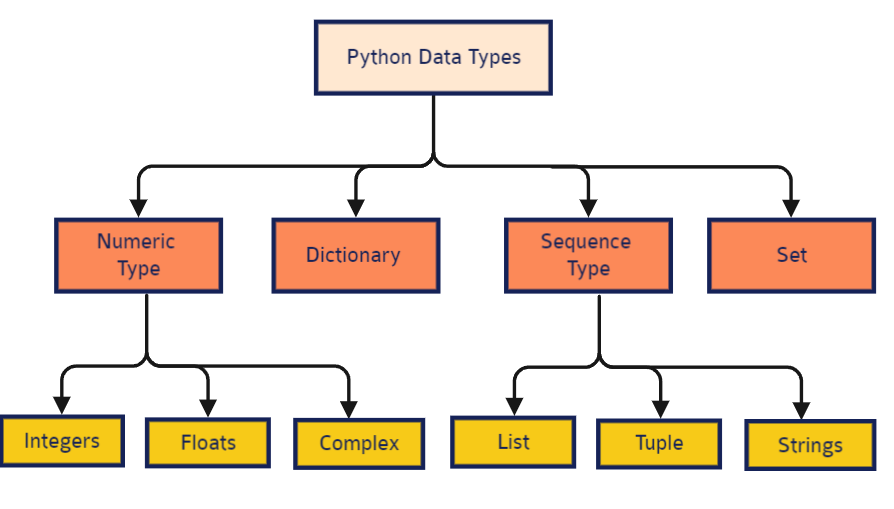
**Comments 🡪** Triple quotations are not comment, it is just a string.

**Data types:**



**Variable 🡪** is just a serial / label to the data location in the memory, Not storing.

**All the data** in Python are 🡪 object.

# Strings “ “

**Triple** double quotes for print multiple lines.

**Indexing --🡪** second number is ignored [0,5] : [0,4]] and last number for step size.

**String methods:**

**\* Strip() & rstrip() & lstrip() : removes whitespaces / or item passed in their parameters.**

**\* title() : Make first character and character after number capital.**

**\* zfill(total size) : Zero fill ( 001 / 011 / 111 ).**

**\* split() & rsplit() : Do tokenization in a list, take parameter for separator and max split.**

**\* swapcase() : reverse capital to small and the reverse is correct.**

**\*startswith() & endswith() : returns Boolean value.**

**\* replace() : (Old value, New value, How many)**

**\* join() : Separator.**

**String formatting:**

***\* Place holder* (%s % “y”)**

**%s 🡪 String**

**%d 🡪 Number**

**%f 🡪 Float**

**🡪 F(“string”) : My fav format‎ ❤**

**………………………………………………………………………….**

**Operations 🡪 ( + , - , / , \* , % , \*\* 🡪 Power, // 🡪 floor division ).**

**………………………………………………………………………….**

# List [ ]

**- Not array. - Mutable.**

**- Not unique. - Can have different formats.**

**List methods:**

**\* append() 🡪 When you append external list in your list, It added totally as element not merged to the original list.**

**\* extend(list) 🡪 Merge the lists.**

**\* remove(“salah”) 🡪 Remove your selected element.**

**\* sort() , sort(reverse = True) 🡪 Sort the list which has the same type of elements.**

**\*reverse() 🡪 Reverse the list.**

**\* clear() 🡪 Make the list Empty.**

**\*copy 🡪 shallow copy (clone: not affected by original list)**

**\* pop() 🡪 (Index): Removes and returns the item/ By default the last element.**

**………………………………………………………………………….**

# Tuple ( )

**-Parentheses are optional.**

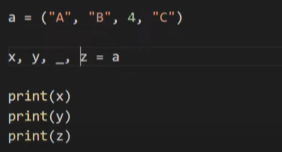
**- immutable.**

**- Not unique, Can store different data types.**

**- Can be concatenated via ‘+’ sign.**

**- Can be repeated via ( n\* 6)**

**Look at this**

****

**The ‘\_’ was ignored and the output will be : A, B, C**

**………………………………………………………………………….**

# Set { }

**\* Not Ordered + No (Indexing / Slicing)**

**\* Has only Immutable data types (Numbers, strings,...)**

**A screen shot of a computer

Description automatically generated**

**\* Its items are unique.**

**Set methods:**

**\* union() 🡪 Make unity of two sets or more.**

**\* add() 🡪 Add element to the set.**

**\* copy() 🡪 Do a shallow copy.**

**\* remove() 🡪 Remove specified element and generates error if you pass non-existing element.**

**\* discard() 🡪 Do the same but does not generate error if you pass non-existing element.**

**pop() 🡪 Generate random element.**

**update() 🡪 Do like union function.**

**difference() 🡪 Make subtraction between sets. (It represents the elements in the first set and not exists in the other + But not updating the original set so when you print the original set it return the original set elements not after updating)**

**difference\_update() 🡪 Do the same of difference() but it updates the original set so when you print the original set it return the updated set elements.**

**Symmetric\_difference 🡪 Represents the elements not existing in the both sets.**

**intersection() 🡪 Represents the intersection between two sets.**

**Intersection\_update() 🡪 Updates the original set.**

**issuperset() & issubset () & isdisjoint() 🡪 Returns Boolean value.**

**………………………………………………………………………….**

# Dectionary { : }

* **Immutable.**
* **Use any data type.**
* **The key must be unique.**
* **Nor ordered, you access via the key not the index.**
* **2D dictionary : dictionary inside dictionary.**
* **You can access to key of subset dictionary via double [key: subset dictionary] [key].**

* **You can define key via inserting variable inside the value.**

**Dictionary methods:**

**clear() 🡪 Make the dictionary empty.**

**update( { k:v } ) 🡪 Add new key & value.**

**copy() 🡪 Do a shallow copy.**

**setdefault( k:v ) 🡪 If the entered key not found in the dictionary, this key and value will be added to the set.**

**popitem() 🡪 Returns last added item.**

**Items() 🡪 Return the set items.**

**Dict.Fromkeys(a,b) 🡪 Add the keys via “a” variable and values via “b” variable.**

**………………………………………………………………………….**

# Boolean

**bool() 🡪** **Returns always true until you pass zero / blank data / False / None .**

**Boolean operators 🡪** AND, OR, NOT

**Assignment operators**

**+= (x += y : x = x+y)**

**- = (x -= y : x = x-y)**

**\*= (x \*= y : x = x\*y)**

**/= (x /= y : x = x/y)**

**\*\*= (x \*\*= y : x = x\*\*y)**

**%= (x %= y : x = x%y)**

**//= (x //= y : x = x//y)**

**Comparison operators**

**==, !=, <, >, <=, >=**

**Type Conversion**

**str(x), tuple(x), set(x), dict(x), int(x), float(x) || X🡪 Is the iterator that will be converted.**

**User Input**

**input(‘String’) and you can:**

* **apply the string methods to clarify the text via x.fn()**
* **apply chain rule to apply multiple methods. ( x.fn().fn() )**

**………………………………………………………………………….**

**Membership Operators**

**( in / not in ) 🡪 Condition returns Boolean value.**

**………………………………………………………………………….**

Practical task recaps on all the previous topics.

**………………………………………………………………………….**

**While Loop**

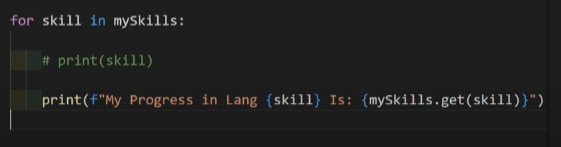
**Loop components: Truly condition 🡪 execution 🡪 loop boundary.**

**………………………………………………………………………….**

**For Loop**

**In dictionary 🡪 the iterator iterates on the key and if you need to print the values you have to select the dictionary and the value’s key you want as follows:**

**A computer screen with text and numbers

Description automatically generated **

**Nested loop 🡪 Inner loop iterates after Outer loop iterates (iterate by iterate) as follows:**

**A computer screen shot of text

Description automatically generated A screenshot of a computer

Description automatically generated**

**Essential topics**

**Continue 🡪 Skip the current iteration.**

**Break 🡪 Stops the loop to the current iteration**

**(If you need to stop the loop at a particular iteration after a particular execution, Don’t forget to put the “break” after the execution statement).**

**Pass 🡪 Ignore the error.**

**………………………………………………………………………….**