Python là ngôn ngữ lập trình bậc cao ( ngôn ngữ lập trình bậc cao là ngôn ngữ gần với con người)

I)Các kiểu dữ liệu trong python

* Int là kiểu số nguyên.
* Float là kiểu số thập phân
* Str là kiểu ký tự ( dạng chuỗi)
* Bool là kiểu logic trả về
* List là kiểu dữ liệu để lưu nhiều mục trong một biến duy nhất (list[])
* Set là giống list và tuple không có chỉ mục và không thay đổi
* Tuple giống list nhưng không thể thay đổi và được viết bằng dấu ngoặc
* Dict là dùng để lưu trữ các giá trị giữ liệu trong cặp key:value và không cho phép trùng lặp

+Nếu bạn muốn kiểm tra kiểu dữ liệu thì hãy sử dụng câu lệnh type

+ Nếu bạn muốn tìm số ngẫu nhiên hãy sử dụng thư viện random (import random và sử dụng câu lệnh random.randrange(1,10))

+ Chỉ có thể nối 2 chuỗi với nhau chứ không thể nối một chuỗi với 1 số để nối đc thì phải ép kiểu.

+ Về chuỗi nếu bạn muốn một chuỗi có thể viết thành nhiều dòng có thể sử dụng “””

* **Toán tử**

+) Toán tử trong python Trong python bạn có thể sử dụng các phép tính toán học cơ bản bằng cách sử dụng những kí tự vd như , công = + , trừ = - , nhân \*, chia= /, chia lấy phần nguyên //, chia lấy dư %, toán tử mũ \*\*

Table

Description automatically generated

+) Toán tử so sánh là toán tử để so sánh hai giá trị

Table

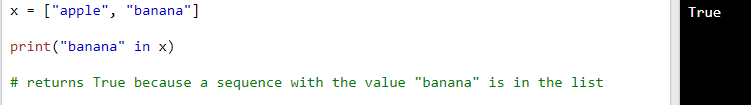
Description automatically generated

+) Toán tử logic dùng để kết hợp các câu lệnh

Table

Description automatically generated

Kiểm tra một chuỗi có được trình bày trong một đối tượng hay không thì chúng ta sử dụng câu lệnh in ( not in )



* **Câu lệnh lựa chọn: Câu lệnh if-else:**

Mệnh đề if-else được dùng để kiểm tra một điều kiện có đúng hay không, nếu đúng thì khối lệnh bên trong if sẽ được thực thi, ngược lại nếu điều kiện là sai thì khối lệnh bên trong else sẽ được thực thi.

VD:

Text

Description automatically generated

* **Vòng lặp:**

Vòng lặp while được dùng để lặp lại một hành động cho tới khi điều kiện lặp không còn thỏa mãn nữa.

Text

Description automatically generated

vòng lặp for được dùng để lặp qua một tập hợp cho trước, vòng lặp for thường được sử dụng với hàm range()

Text

Description automatically generated with medium confidence

**-Câu lệnh Break**

Câu lệnh break được dùng để thoát khỏi vòng lặp, khi chương trình gặp câu lệnh break thì vòng lặp sẽ không được thực thi tiếp.

Graphical user interface, text, application, chat or text message

Description automatically generated

**-Câu lệnh continue**

Khi gặp câu lệnh continue trong vòng lặp, các đoạn code bên dưới câu lênh này trong vòng lặp sẽ không được thực thi

Graphical user interface, text, application, chat or text message

Description automatically generated

* **List**

**1. Hàm len**

Đây là hàm trả về số phần tử có trong list :

Text

Description automatically generated

**2. Hàm max, min**

Đây là hai hàm được dùng để trả về phần tử lớn nhất và nhỏ nhất trong list :

Text

Description automatically generated

**3. Hàm insert**

Đây là hàm dùng để thêm một phần tử vào một ví trí trong list:

Text

Description automatically generated

Kết quả khi chạy



**4. Hàm remove**

Hàm này dùng để xóa một phần tử khỏi list:

Text

Description automatically generated

**5. Hàm pop**

Hàm pop() được dùng để xóa một phần tử với chỉ số cho trước trong list:

Text

Description automatically generated

**6. Hàm sort**

Hàm này được dùng để sắp xếp các phần tử trong list theo một thứ tự nhất định

lst = [4, 5, 3, 7, 6, 1]

lst.sort()

print(lst)

lst.sort(reverse=True)

print(lst)

Kết quả khi chạy:

[1, 3, 4, 5, 6, 7]

[7, 6, 5, 4, 3, 1]

**7. Hàm reverse**

Đây là hàm dùng để đảo ngược list:

lst = [4, 5, 3, 7, 6, 1]

lst.reverse()

print(lst)

Kết quả khi chạy:

[1, 6, 7, 3, 5, 4]

**8. Hàm count**

Đây là hàm dùng để đếm số lần xuất hiện của một thành phần trong list:

lst = [6, 2, 3, 8, 2]

print(lst.count(2))

Kết quả khi chạy:

2

**9. Hàm clear**

Đây là hàm dùng để xóa hết các phần tử bên trong list:

lst = [1, 2, 3]

lst.clear()

print(lst)

Kết quả khi chạy:

[]

* **Chuỗi trong python**

Trong Python, chuỗi là một tập hợp các kí tự Unicode. Có thể sử dụng 3 cặp nháy đơn, nháy đôi để tạo ra một chuỗi trên nhiều dòng.

**Một số hàm và phương thức để xử lý chuỗi trong Python**

**1. Hàm len()**

Đây là hàm trả về độ dài của chuỗi:

s = "abc"

print(len(s))

KQ khi chạy:

3

**2. Phương thức lower()**

Đây là phương thức được dùng để chuyển 1 chuỗi về dạng in thường:

s = "CODELEARN123"

print(s.lower())

KQ chạy:

codelearn123

**3. Phương thức upper():(Ngược lại với lower)**

Đây là phương thức được dùng để chuyển 1 chuỗi về dạng in hoa:

**4. Phương thức isalnum()**

Đây là phương thức được dùng để kiểm tra xem một xâu có chỉ chứa các ký tự chữ và số hay không.

s = "codelearn2020"

print(s.isalnum())

kq chạy :

True

**5. Phương thức isalpha()**

Phương thức này được dùng để kiểm tra xem một chuỗi có chứa toàn các ký tự chữ không:

s = "codelearn"

print(s.isalpha())

kq chạy :

True

**6. Phương thức isnumeric()**

Phương thức này dùng để kiểm tra xem một xâu có chứa toàn các ký tự số hay không:

**7. Phương thức split()**

Phương thức này được dùng để cắt một chuỗi ra thành list các chuỗi khác dựa trên một phần tử trong chuỗi đầu vào:

s = "A1B1C1D1E1"

print(s.split("1"))

KQ chạy:

['A', 'B', 'C', 'D', 'E', '']

**8. Phương thức join()**

Phương thức này được dùng để nối một tập hợp thành một chuỗi sử dụng kí tự cho trước.

lst = ["A", "B", "C"]

print("-".join(lst))

KQ chạy:

A-B-C

**9. Phương thức replace()**

Phương thức này được dùng để thay thế các chuỗi con tìm thấy thành chuỗi con mới.

name = "Cod3l3arn"

print(name.replace("3", "e"))

KQ chạy:

Codelearn

**10. Cắt chuỗi trong Python**

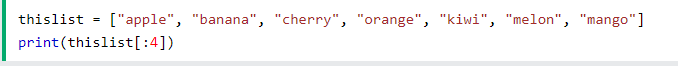
+)Để lấy ra một dãy các kí tự liên tiếp trong chuỗi.

+) Có thể thêm bước nhảy để bỏ qua các kí tự .

str = "bigcityboy"  
print (str[3:7:2])

KQ : ict

+) Cách để truy xuất phần tử trong danh sách làhttps://f20-zpc.zdn.vn/9025171273006474152/f4f72e6a49468c18d557.jpg

+) Bạn cũng có thể dùng phương pháp slicing 

+) Các phương thức để thêm phần tử vào list

1. Insert(vị trí, phần tử)
2. Append(phần tử)
3. Extend(phần tử) có thể dùng để nối 2 list

+) các phương thức để xóa phần tử trong list

Remove(phần tử)

Pop(vị trí(có hoặc không))

Del xóa bỏ hẳn

Clear() xóa phần tử trong list

+) phương thức để sắp xếp list (sort())

* Nếu phần tử là str thì nó sẽ sắp theo bảng chữ cái
* Nếu phần tử là số nó sẽ xếp theo tăng dần
* Còn muốn đảo ngược lại thì thêm reverse = true
* Lưu ý sort phân biệt chữ hoa và chữ thường

III) Tuple

Y hệt list chỉ khác là không thay đổi vì vậy muốn them sửa xóa bạn phải ép kiểu thành list[] rồi sau khi xong bạn ép kiểu lại

Bạn có thể kết hợp các tuple với nhau

IV) SET

Sẽ loại bỏ giá trị trùng

Khi bạn muốn xóa phần tử trong set thì nên dùng remove(phần tử) và discard(phần tử) và không nên sử dụng pop vì nó xóa ngẫu nhiên

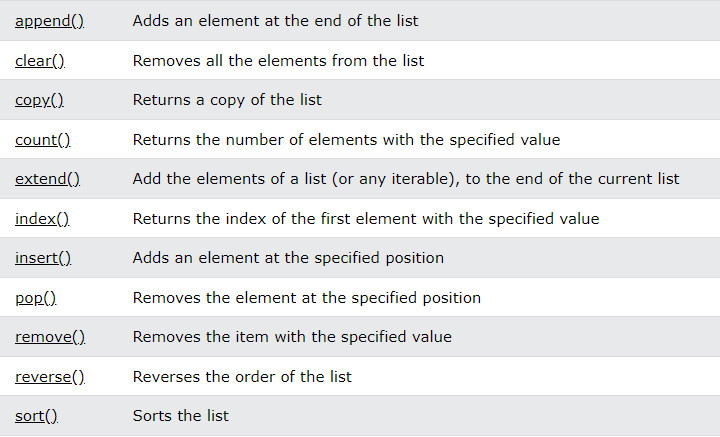
Muốn kết nối 2 set bạn có thể sử giữ dụng

Union()

Update

Intersection giữ lại giống nhau

Symmetric\_diference bỏ sự trùng lặp



Append: thêm một phần tử vào cuối danh sách

Count: Trả về số phần tử có giá trị được chỉ định

Extend: thêm các phần tử của danh sách (hoặc bất kỳ có thể lặp lại nào), vào cuối danh sách hiện tại

Index : Trả về chỉ mục của phần tử đầu tiên với giá trị được chỉ định

Insert: thêm một phần tử ở vị trí được chỉ định

Pop: Loại bỏ phần tử ở vị trí đã chỉ định

* **Git**

Gitlab là một dịch vụ cung cấp quyền truy cập từ xa vào kho Git. Ngoài việc lưu trữ mã của bạn, các dịch vụ cung cấp các tính năng bổ sung được thiết kế để giúp quản lý vòng đời phát triển phần mềm. Các tính năng bổ sung này bao gồm quản lý việc chia sẻ mã giữa những người khác nhau, theo dõi lỗi, không gian wiki và các công cụ khác để 'mã hóa xã hội'.

- Tạo 1 kho lưu trữ

Nhấp vào Repositories => New => vào file ấn cdm => Git status => Git add . => git commit -m "tên muốn lwuu " => git push -u origin main

Các lệnh git;

Đẩy file lên githud

Git status

Git add .

git commit -m "first commit"

git push -u origin main

git pull để sửa lại code

OPP Python

**Khái niệm:** Python là một ngôn ngữ hướng đối tượng. Vì vậy mọi thứ trong Python đều là những đối tượng với những thuộc tính ( properties ) và phương thức (Methods).

- OOP là một kĩ thuật lập trình cho phép tạo ra các đối tượng để trừu tượng hóa 1 đối tượng thực tế (đưa các đối tượng trong thực tế vào trong code). Cho phép lập trình viên tương tác với các đối tượng này.

- Một đối tượng bao gồm: thuộc tính (attributes) và phương thức (methods).

-Thuộc tính (attributes) chính là những thông tin, đặc điểm của đối tượng.

VD: Con chó ( màu sắc, 4 chân,…)

- Phương thức (methods) là những thao tác, hành động mà đối tượng đó có thể thực hiện.

VD: Con chó ( sủa, chạy, …)

- Lớp có thể hiểu là một bản thiết kế để tạo ra một thực thể nào đó, là tập hợp nhiều thuộc tính đặc trưng cho đối tượng được tạo ra từ lớp này. VD: con chó ( màu sắc, 4 chân, sủa, chạy, …)

- Đối tượng là một thực thể của 1 lớp nào đó, được tạo ra từ lớp đó. VD: chó husky ( tên là husky, màu đen, mắt xanh,…)

- OOP tuân theo một số nguyên lý cơ bản, gồm có 4 nguyên lý: Tính đóng gói, Tính kế thừa, Tính bao đóng, Tính đa hình.

**Các nguyên lý cơ bản của OOP**

* Tính đóng gói (Encapsulation): Các dữ liệu và phương thức có liên quan với nhau được đóng gói thành 1 lớp. Tức là mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng lớp đó. Che giấu các thông tin của lớp đó đối với bên ngoài thể hiện ở public, protected, private đối với từng thuộc tính và phương thức.
* TÍnh kế thừa (Inheritance): Nguyên tắc này cho phép xây dựng một lớp mới dựa trên 1 lớp đã khai báo từ trước. Lớp con có thể sử dụng lại các thuộc tính và phương thức của lớp cha mà không cần khai báo lại. Tùy thuộc vào từng ngôn ngữ cho phép việc kế thừa 1 hoặc nhiều class cha.
* Tính trừu tượng (Abstraction): tổng quát hóa phương thức của đối tượng không quan tâm phương thức thực hiện như thế nào, được thể hiện bởi interface (có các tên phương thức nhưng ko có body của phương thức, khi class nào impliment interface thì thực hiện nó).
* Tính đa hình (Polymorphism): Tính đa hình được thể hiện bởi một phương thức, hành động có thể thực hiện theo nhiều cách khác nhau.

**Class, methods, attributes**

-Trong python có nhiều lớp đã được định nghĩa sẵn. Do mọi thứ trong python đều là đối tượng ví dụ như list, tuple, dictionary, string, int… là các lớp. Khi chúng ta khai báo biến thuộc các lớp này thì chúng là các đối tượng.

Githud

Opp python

Khái niệm: Python là một ngôn ngữ hướng đối tượng. Vì vậy mọi thứ trong Python đều là những đối tượng với những thuộc tính ( properties ) và phương thức (Methods)