Bố cục file code python

1. **Import thư viện**
2. **cấu hình cổng COM**
3. **Vòng lặp**

Các từ khóa trong python

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| False | Await | Else | Import | Pass |
| None | Break | Except | In | Raise |
| True | Class | Finally | Is | Return |
| And | Continue | For | Lamda | Try |
| As | Def | From | Nonlocal | While |
| Assert | Del | Global | Not | With |
| Async | Elif | If | Or | Yield |

1. Biến là gì ?

Biến được gán bằng 1 giá trị

* Khởi tạo biến

Tên biến = giá trị của biến

Vd : a = 849738

b = 775365

* Tên của biến chỉ chứa các chữ cái , số và dấu gạch dưới “\_”
* Tên của biến không được trùng với từ khóa trong python
* Khởi tạo nhiều biến

< tên biến T1>, < tên biến T2>, …< tên biến n> = < giá trị biến T1>, < giá trị biến T2> …< giá trị biến n>

* Kiểm tra kiểu dữ liệu của biến :sử dụng hàm “type()”

Type (<tên biến>)

* Các kiểu dữ liệu

int (Intergers) : số nguyên

Str (string): chuỗi

float : số thực

1. Chuỗi

* Chuỗi (str( string)) là 1 kiểu dữ liệu dùng để lưu trữ 1 dãy ký tự như : chữ cái, số, ký hiệu, khoảng trắng,…
* Chuỗi thường nằm bên trong dấu ‘..’ hoặc dấu “..”

VD : S3 = “ Hello”

S2 = ‘ Python’

S1 = “”” chuỗi nhiều dòng ”””

* Các kiều định dạng chuỗi
* Kiểu toán tử % : sử dụng dấu % kết hợp với specifier (%s , %d, %f ,…)

+ Các specifier phổ biến

* %s : chuỗi (str)
* %d : số nguyên (int)
* %f : số thực (fload)
* Str. Format()
* Sử dụng dấu {} để chèn biến
* Có thể đánh chỉ số hoặc đặt tên
* Hỗ trợ định dạng nâng cao
* F- strings
* Thêm chữ “ f “ trước chuỗi
* Có thể nhúng biến hoặc biểu thức python trực tiếp trong {}
* Căn lề bằng phương thức format

|  |  |
| --- | --- |
| Căn lề trái | {:(c)<n} |
| Căn lề phải | {:(c)>n} |
| Căn giữa | {:(c)^n} |

Với c là kí tự muốn thay thế vào chỗ trống , nếu để trống thì sẽ là kí tự khoảng trắng , còn n là số kí tự dùng để căn lề

* Phương thức biến đổi chuỗi
* Capitalize : viết hoa kí tự đầu của 1 chuỗi và viết thường các kí tự đằng sau

“Hello” .capitalize()

* Title : Viết hoa chữ cái đầu mỗi từ
* Swapcase : Đổi chữ hoa thành chữ thường và ngược lại
* Lower : chuyển toàn bộ chuỗi thành chữ thường
* Upper : chuyển toàn bộ chuỗi thành chữ hoa
* Str.center() : a= “Hello”, b = a.center(width,[fillchar]): trả về 1 chuỗi được căn giũa với chiều rộng width, fillchar là 1 chuỗi có độ dài là 1
* Rjust : căn phải
* Ljust : căn trái
* Encode : là phương thức để mã hóa 1 chuỗi
* Join : cộng các chuỗi
* Replace : thay thế chuỗi
* Strip : trả về 1 chuỗi với phần đầu và phần đuôi của chuỗi được bỏ đi các kí tự char
* Phương thức tách chuỗi
* Split : cắt chuỗi
* Find : tìm 1 chuỗi trong chuỗi
* Partition : trả về 1 tuple gồm 3 phần ( trước, c, sau)
* Count : đếm số lần xuất hiện của một chuỗi con trong chuỗi gốc
* Startswith : kiểm tra một chuỗi có bắt đầu bằng một chuỗi con nào đó hay không

1. List

* List được giới hạn bởi cặp ngoặc vuông []
* Các phần tử của list cách nhau bởi dấu ,
* List có khả năng chứa mọi giá trị đối tượng của python bao gồm cả chính nó
* Count : đếm số lần xuất hiện của 1 phần tử trong list
* Index : trả về 1 chuỗi là số nguyên là vị trí phần tử trong list

Vd : a.index()

* Copy : tạo ra 1 bản sao
* Clear : xóa mọi phần tử trong list
* Append : thêm 1 phần tử vào cuối list
* Insert : chèn phần tử vào bên trong list
* Extend : mở rộng danh sách bằng cách thêm từng phần tử của một iterable (list, tuple, set, string...) vào cuối list
* Pop : lấy ra 1 phần tử bên trong list
* Remove : bỏ đi phần tử đầu tiên trong list có giá trị x
* Reverse : đảo ngược list
* Sort : sắp xếp các phần tử trong list ngay tại chỗ

1. Tuple

* Được giới hạn bởi cặp ngoặc ()
* Các phần tử của tuple được phân cách nhau bởi dấu ‘,’
* Tuple có khả năng chứa mọi giá trị
* Tốc độ truy xuất của tuple nhanh hơn list
* Dung lượng chiếm dụng bộ nhớ nhỏ hơn list
* Bảo vệ dữ liệu không bị thay đổi
* Có thể dùng làm key của dictionary

1. Set

* Được giới hạn bởi cặp ngoặc nhọn {},tất cả những gì nằm trong đó là phần tử của set
* Các phần tử của set được phân cách nhau bởi dấu (,)
* Set ko chứa nhiều hơn 1 phần tử trùng lặp
* Toán tử & : kết quả trả về là 1 set chứa các phần tử chung.
* Toán tử ^ : lấy phần tử không trùng nhau
* Toán tử ‘|’: gộp tất cả phần tử , bỏ phần tử trùng
* phương thức pop: xóa và trả về 1 phần tử bất kì

1. Dict(dictionary)

* Cũng là 1 container như LIST, TUPLE . khác biệt là những container như list, Tuple có các index để phân biệt các phần tử thì Dict dùng key để phân biệt
* Một Dict gồm các yếu tố sau
* Được giới hạn bởi cặp ngoặc nhọn{}, tất cả những gì nằm trong đó là phần tử của Dict.
* Các phần tử của Dict được phân cách nhau bởi dấu phảy (,)
* Cặp Key-value của phần tử trong Dict được phân cách bởi dấu (:)
* Các key buộc phải là 1 hash object
* Cách khởi tạo Dict

Sử dụng dấu ngoặc {} và đặt giá trị bên trong

Vd : {(key\_1 : value\_1), (key\_2 : value\_2),…,key\_n : value\_n)}

* Copy: Tạo 1 bản sao mới tại 1 vùng dữ liệu mới
* Clear: xóa toàn bộ phần tử trong dict
* Get : là một phương thức dùng để lấy giá trị từ một key
* Items : trả ra 1 tuple với giá trị đầu tiên là key, giá trị thứ 2 là value
* Keys : lấy ra key trong dict
* Values: lấy ra value trong dict
* Pop: lấy ra và loại bỏ phần tử
* SetDefault: trả về giá trị của key trong list nếu key ko có trong dict sẽ trả về giá trị Default , thêm nữa 1 cặp key:value mới sẽ được thêm vào dict với Key=key và value=default , default mặc định là none

1. Cấu trúc rẽ nhánh
   1. If – elif – else
   2. Vòng lặp While

Lặp lại khi nào điều kiện còn đúng(True)

* 1. Vòng lặp for

Vd :For variable\_1, varialble\_2,…,variable\_n in sequence

* Dùng để lặp qua một **dãy** (list, tuple, string, range, dict, set, …)