Graphical user interface, application

Description automatically generated

Tạo những code riêng lẻ - khi chạy 1 cái thì những cái kia không ảnh hưởng

Tuy nhiên python cũng sẽ lưu những cái code đã chạy xong – nếu cái sau có liên quan cái trước mà chưa chạy cái trước thì cái sau sẽ bị lỗi (ví dụ ở dưới, thứ tự trong note kh quan trọng, trước sau ở đây là precedence )

**ipynb khác py ntn 🡪 py là gốc của python (chạy hết lun), ipynb là jupyter notebook (chạy code từng block)**

Cách search: A picture containing text

Description automatically generated

Variable: a=5

Biến là 1 ô nhớ trong ram chứa giá trị

Biến chứa dữ liệu: số (integer, float) và chữ (string, charaters - ở python thì 2 cái này là 1 cái là string)

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

* Phép tính:

Các phép tính sẽ nhập trực tiếp vô tính như bth

Graphical user interface, application

Description automatically generated

Riêng phép chia có đặc biệt – 2 số int chia nhau ra số float vì không phải phép chia sao cũng ra số nguyên

Chia lấy phần dư

Graphical user interface

Description automatically generated

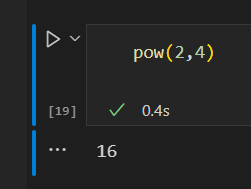
Nhập số giây ra số giờ và số giây lẻ

Phải import math trước để có hàm math.floor để làm tròn xuống 🡪 chỉ cần gọi 1 lần

Text

Description automatically generated

* Các hàm khác:

 Graphical user interface, application

Description automatically generated

* Array: Mảng 🡪 Không thể thay đổi được phần tử (kích thước không thay đổi)

Ở python kh xài array 🡪 xài list 🡪 có thể thay đổi được phần tử 🡪 dynamic

Mỗi lần thên phần tử sẽ tạo 1 ô nhớ mới 🡪 mỗi lần thêm sẽ tạo 1 bộ nhớ tạm để chứa 🡪 đọc từ sau tới trước

A picture containing text

Description automatically generated

\*một vô chứa value, 1 ô chứa địa chỉ (vị trí)

Nhược điểm: Ngoài lưu trữ giá trị thì cần lưu trữ vị trí của thằng trước đó (vì dynamic quá)

Graphical user interface

Description automatically generated A screenshot of a car dashboard

Description automatically generated with low confidence

Mỗi lần run cái append 🡪 add thêm 1 phần tử mới mở rộng cái list ra

Có thể append chữ nma phải consistent 🡪 trước số sau số, trước chữ sau chữ

Cách lấy ra 1 phần tử từ list 🡪 pop thứ tự của cái mún bỏ 🡪 đếm từ 0

Graphical user interface, text

Description automatically generated

Print lại chỉ có 1 số 3

Text

Description automatically generated

Cách truy cập 1 số bất kỳ trong list:

Graphical user interface

Description automatically generated

* Vòng lặp:

For và While

For biết số lần lặp rồi

While chưa biết và chạy theo điều kiện

Text

Description automatically generated with medium confidence

Nhớ phải tab vô ở chỗ print

Text

Description automatically generated

Cách này là nhập đầu cuối và step

Cái cuối là phải cộng 1 thêm 🡪 như trên thì nó sẽ chạy từ 1 tới 10 và step 2

Text

Description automatically generated

**Có tab vô là thực thi trong vòng for**

Cách nhập theo len

* Golden ratio

Dãy số mà số sau bằng 2 số trước nó cộng lại 🡪 các con số càng tăng thì khoảng cách giữa các con số dần tới tỉ lệ vàng là 0.7….

Text

Description automatically generated

0, 1 là 2 số cơ bản của dãy số

Số 2 là số thứ tự 2 (bắt đầu từ 0)

Append tổng của của 2 thằng trước

* Tìm số nguyên tố

2 dấu bằng nghĩa là so sánh

Break là ngưng vòng lặp khi mà có số chia hết

Break khi if đúng

🡪 không chạy từ 1 vì nó nhận số 1 chia hết

 🡪 không check n lun vì chính nó chia hết

Text

Description automatically generated

Break là khi đúng điều kiện rồi thì sẽ dừng kh chạy cái dưới nữa

Diagram

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

<https://quantrimang.com/lenh-break-va-continue-trong-python-142623>

Có 3 TH đặc biệt là 0 và 1 và 2

* Xài function:

Return 🡪 bao gồm break ở trỏng lun

Graphical user interface

Description automatically generated or return (2\*x +1) 🡪 gán giá trị cho hàm

Text

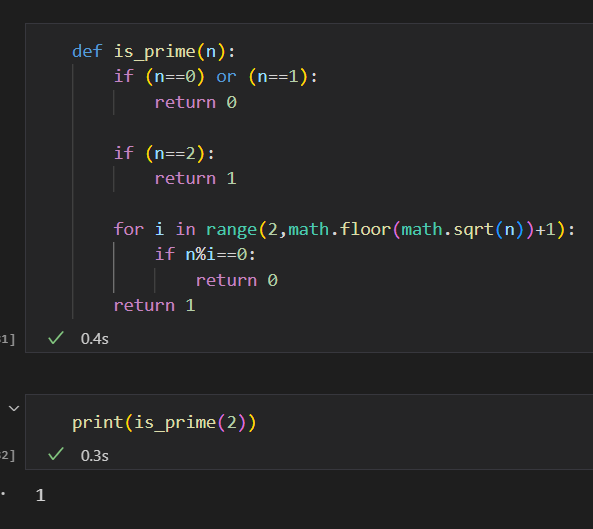
Description automatically generated

Text

Description automatically generated

Hàm số nguyên tố

A screenshot of a computer

Description automatically generated with medium confidence 

Để chạy tới căn n để hạn chế số lần chạy 🡪 chạy nhanh hơn s

* Knapsack

<https://drive.google.com/file/d/1rvS2GaB3Rt2uA4Vrycpw23W2_b4CFOSE/view?usp=sharing>

w[i-1] vì trong python bắt đầu đếm từ 0

NumPy

Cộng trừ nhân chia ma trận 🡪 Linear

Phải để Array vô numpy thì mới cộng trừ nhân chia dc

Graphical user interface, text, application

Description automatically generated

Ghép mảng 1 chiều có 2 cách ghép, ghép mảng 2 chiều có 3 cách

* Object oriented programming

Lập trình hướng đối tượng: tạo ra 1 kiểu dữ liệu mới cho riêng mình thay vì float, integer

Object 🡪 ví dụ: sinh viên: tên, tuổi,….

2 thứ quan trọng:

Properties: Thuộc tính

Method 🡪 đầu tiên phải tạo hàm khởi tạo (constructor)

* GA (genetic algorithm):

Meta heuristic 🡪 Ali Mirjalili

Những gen tốt thích nghi với môi trường có khả năng sinh sôi, những gen khôgn tốt thì bị loại bỏ 🡪 càng về sau những gen tốt sẽ càng tốt hơn

1 DNA trong GA là 1 cục decision var (input) 🡪 bỏ vô f(x) 🡪 tạo ra output

Cái nào tốt mình lấy để làm các thế hệ kế tiếp

**Trao đổi chéo** – 1 DNA là bố 1 DNA là mẹ 🡪 tạo ra thế hệ tiếp

**Đột biến** – tự thay đổi 1 số trong DNA

Khi nào nó tốt đến mức không thể tăng thêm được thì dừng lại

Định nghĩa DNA

Knapsack decision var là 1 array [1,0,1,1,…] 🡪 tìm value lấy sumproduct

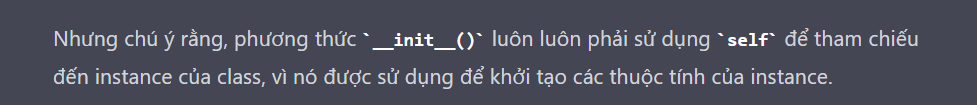
Tạo ra 1 mảng random đầu tiên xong tiến dần tới cái optimal

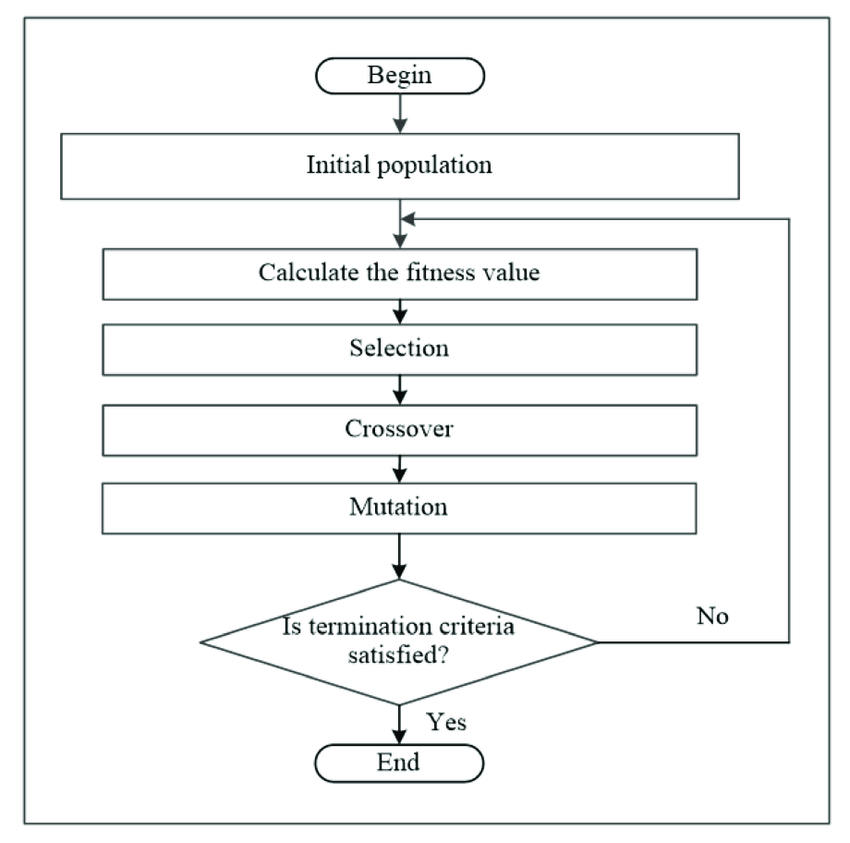
Mình viết code để tự dừng:

* Đến khi đạt giá trị mình mún
* Đến 1 số vòng lặp nhất định

Text

Description automatically generated





\*Traveling salesman problemm: đi qua tất cả các điểm và quay về với tổng cót nhỏ nhất

VRP: vehical rooting problem 🡪 bản chất là nhiều TSB

Số nguyên xài GA

Số thực xài PSO

* Optimization

Deterministics: xác định cthuc sẵn rùi 🡪 khó xác định cthuc chung

* Dynamic programming (quy hoạch động):
* simplex method, branch and bound

Metaaaaaa: mò tới khi có đáp án tốt nhất

* GA: xài khi các decision var discrete (evolutionary - Solver) 🡪 có nhiều loại GA
* xài cho continuos des var (Ali Mirjalili)

+ ACO:

+ PSO:

+ GWO:

+ Bee algorithm:

+ HHO:

* Máy học: suppervise, unsuppervise

Regression

* Logistics regression
* Linear regression
* Polynomio regression
* Neural network regression

Classification

* SVM
* KNN

Weight là mức quan trọng của liên kết

Weight mới khởi tạo là random 🡪 sau đó minh train nó để ra đúng ý minh

Fit lần đầu weight sẽ kh đung 🡪 sửa dần dần để ra đung

Fit foward 🡪 back-propagation

Bằng cách tinh lỗi, cái nào lỗi nhiều thì chỉnh nhiều

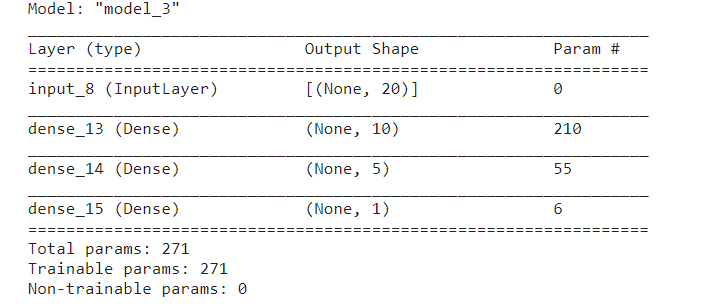
Các weight tạo thành ma trận

Nẻon network là tìm điểm chung data và giảm dimension

1 perceptron chai dc 1 dg thẳng phân laoij, nhiều per thì chia ra đường kh thẳng

Graphical user interface, text, application

Description automatically generated



271 = 20\*20 + 10\*5 + 5\*1 + 10 + 5 +1 🡪 params là những cái minh phải sảuw

Sigmoid – relu – tanh