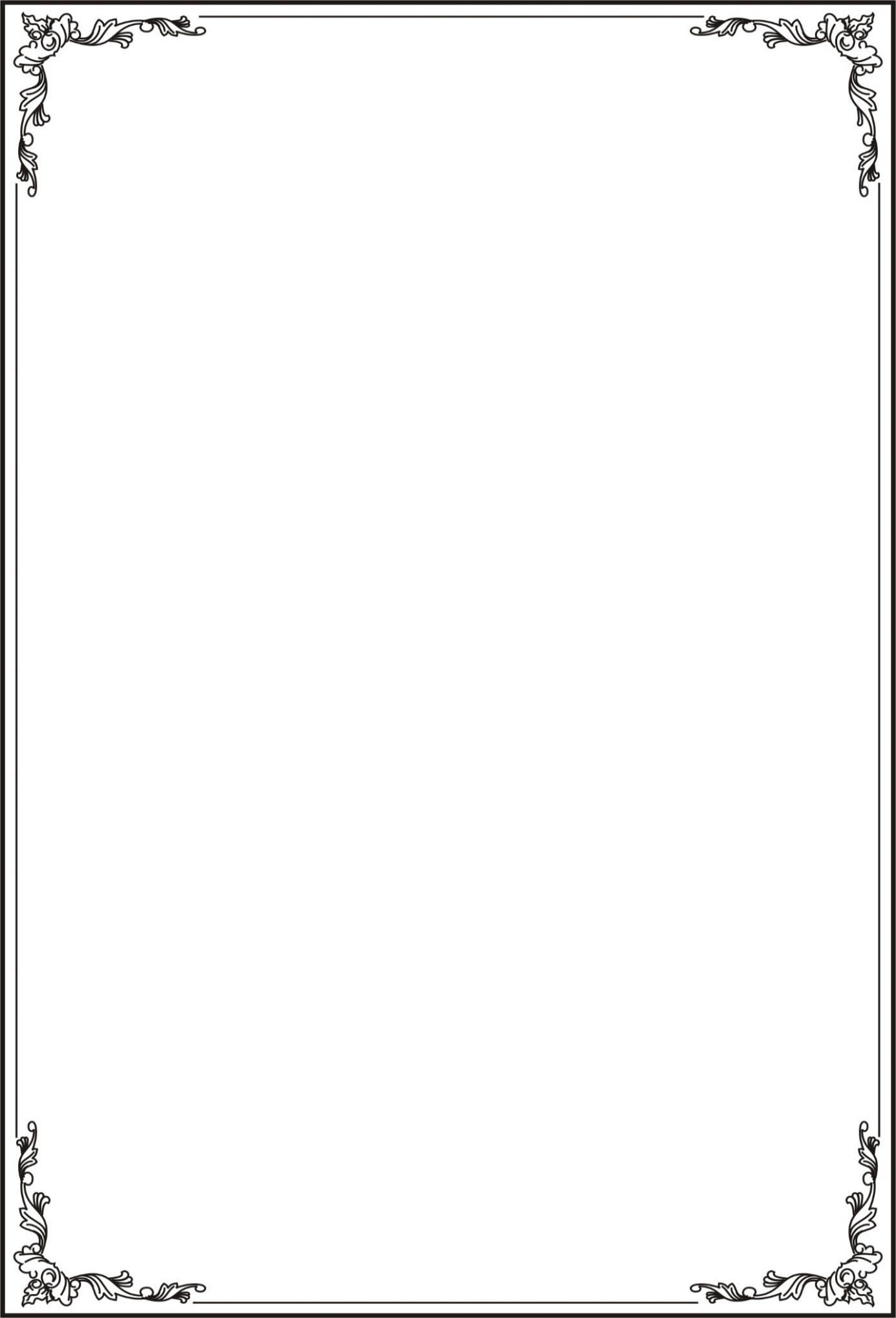
**** **BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**

**ĐỒ ÁN 1**

**ĐỀ TÀI: XÂY DỰNG GIAO DIỆN PYTHON**

**SINH VIÊN THỰC HIỆN: LÊ HỮU DANH**

**MSSV: 16110031**

**GIẢNG VIÊN HƯỚNG DẪN: TRẦN TIẾN ĐỨC**

**TP. HỒ CHÍ MINH – NĂM 2020**

**LỜI CẢM ƠN**

*Trước hết, em tỏ lòng biết ơn sâu sắc đến Thầy Trần Tiến Đức đã tận tình hướng dẫn trong đồ án 1. Với những kiến thức, nhận xét đã giúp chúng em chỉnh sửa những thiếu sót và hoàn thành tốt Đồ án 1.*

*Ngoài ra, những kiến thức và ví dụ thực tiễn từ thầy đã giúp chúng em hiểu rõ và áp dụng dụng đúng đắn hơn vào Đồ Án. Do các kỹ năng cá nhân cũng như kiến thức còn ít ỏi nên thông qua đồ án môn học nói riêng, môn Đồ Án 1 nói chung, nên việc sai sót là điều không thể tránh khỏi trong quá trình thực hiện. Chúng em kính mong nhận được những lời nhận xét, ý kiến đóng góp quý báo từ quý Thầy Cô để em có thêm kinh nghiệm và hoàn thành tốt hơn những đề tài sau này.*

*Em xin chân thành cảm ơn!*

*Thành phố Hồ Chí Minh, ngày 08 tháng 01 năm 2020*

*Sinh viên thực hiện*

***Lê Hữu Danh***

**MỤC LỤC**

[CHƯƠNG 1: KHÁI NIỆM, ĐẶC ĐIỂM ỨNG DỤNG TRONG LẬP TRÌNH CỦA PYTHON 3](#_Toc29732576)

[1.1 Tìm hiểu về Python 3](#_Toc29732577)

[1.2 Lịch sử phát triển 3](#_Toc29732578)

[1.3 Đặc điểm chung 4](#_Toc29732579)

[1.4 Đặc điểm trong cú pháp, quy tắc lập trình 4](#_Toc29732580)

[1.5 Câu lệnh rẽ nhánh 5](#_Toc29732581)

[1.5.1 Luồng xử lý 5](#_Toc29732582)

[1.5.2 Các câu lệnh rẽ nhánh 5](#_Toc29732583)

[1.6 Vòng lặp 6](#_Toc29732584)

[1.7 Hàm 7](#_Toc29732585)

[1.7.1 Khai báo hàm trong Python 7](#_Toc29732586)

[1.7.2 Gọi hàm 8](#_Toc29732587)

[1.7.3 Hàm có kết quả trả về 8](#_Toc29732588)

[1.7.4 Truyền vô tham số vào hàm 9](#_Toc29732589)

[1.8 Modules 9](#_Toc29732590)

[1.8.1 Modules là gì? 9](#_Toc29732591)

[1.8.2 Modules trong Python 9](#_Toc29732592)

[CHƯƠNG 2: TRÌNH BÀY PHẦN TRIỂN KHAI GIAO DIỆN …………………………...11](#_Toc29732593)

[2.1 Triền khai giao diện đồng hồ kim 11](#_Toc29732594)

[2.2 Triển khai giao diện đồng hồ số 13](#_Toc29732595)

[2.3 Triển khai giao diện đồng hồ bấm giờ 14](#_Toc29732596)

[KẾT LUẬN 16](#_Toc29732597)

MỞ ĐẦU

Ngành công nghệ thông tin là một trong những ngành khoa học đang trên đà phát triển mạnh và ứng dụng rộng rãi trên nhiều lĩnh vực. Cùng với xu hướng phát triển của các phương tiện truyền thông như TV, Radio, báo giấy,... thì việc sử dụng Internet ngày càng phổ biến. Truy cập Internet, chúng ta có được một kho thông tin khổng lồ phục vụ mọi nhu cầu, mục đích của chúng ta chỉ bằng một cái nhấp chuột.

Nhận thức được nhu cầu tìm hiểu thông tin, giải trí của xã hội, từ đó ra đời hàng loạt website cho các mục đích thương mại, giải trí, tin tức…

Việc tham gia thực tập ngay khi còn ngồi trên ghế nhà trường trở thành một nhu cầu cấp bách cho mỗi người sinh viên, giúp sinh viên trau dồi kiến thức làm việc thực tế cũng như văn hóa làm việc nơi công sở.

**Mục tiêu của đề tài:**

Trong thời điểm Công nghệ lập trình phát triển mạnh và vô cùng nhanh chóng như hiện nay, những ngôn ngữ mới, thư viện mới, framework mới xuất hiện ngày càng nhiều và có rất nhiều điểm mạnh riêng đáp ứng hầu hết nhu cầu ngày càng cao của các công ty, các đối tượng khách hàng. Cuối năm 2019, Python nằm trong mọi bảng xếp hạng xu hướng lập trình được yêu thích hàng đầu, phổ biến nhất thế giới, cũng như đáp ứng đủ nhu cầu tuyển dụng. Nắm bắt điểm mạnh vượt trội của chúng, trong Đồ Án 1 của mình, em đã tìm hiể về chúng và ứng dụng trong lập trình.

**Kết cấu của đề tài:**

Bố cục của đồ án 1 gồm:

Mở đầu

Chương 1: Khái niệm, đặc điểm và ứng dụng trong lập trình của Python

* 1. Tìm hiểu về Python
  2. Lịch sử phát triển
  3. Đặc điểm chung
  4. Đặc điểm trong cú pháp, quy tắt lập trình
  5. Câu lệnh rẽ nhánh
  6. Vòng lặp
  7. Hàm
  8. Modules

Chương 2: Trình bày phần triển khai giao diện Python

2.1 Triển khai giao diện đồng hồ kim

2.2 Triển khai giao diện đồng hồ số

2.3 Triển khai giao diện đồng hồ bấm giờ

Kết luận

# KHÁI NIỆM, ĐẶC ĐIỂM ỨNG DỤNG TRONG LẬP TRÌNH CỦA PYTHON

## Tìm hiểu về Python

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

## Lịch sử phát triển

Python là một ngôn ngữ khá cũ được tạo ra bởi Guido Van Rossum. Thiết kế bắt đầu vào cuối những năm 1980 và được phát hành lần đầu tiên vào tháng 2 năm 1991.

Vào cuối những năm 1980, Guido Van Rossum làm việc trong Amoeba, phân phối một nhóm hệ điều hành. Ông muốn sử dụng một ngôn ngữ thông dịch như ABC (ABC có cú pháp rất dễ hiểu) để truy cập vào những cuộc gọi hệ thống Amoeba. Vì vậy, ông quyết định tạo ra một ngôn ngữ mở rộng. Điều này đã dẫn đến một thiết kế của ngôn ngữ mới, chính là Python sau này.

## Đặc điểm chung

* 1. **Ưu điểm:**

1. Là một ngôn ngữ có hình thức sáng sủa, cấu trúc roc ràng, cú pháp ngắn gọn
2. Có trên tất cả các nền tảng hệ điều hành từ UNIX, MS – DOS, Mac OS, Windows và Linix và các OS khác thuộc họ Unix.
3. Tương thích mạnh mẽ với Unix, hardware, thirt-party software với số lượng thư viện khổng lồ (400 triệu người sử dụng)
4. Python với tốc độ xử lý cực nhanh, python có thể tạo ra những chương trình từ những script siêu nhỏ tới những phần mềm cực lớn như Biender 3D.
   1. **Nhược điểm:**
      * 1. Python không có các thuộc tính như :protected,private hay public, không có vòng lặp do…while và switch….case.
        2. Mặc dù tốc độ xử lý của Python nhanh hơn PHP nhưng không bằng JAVA và C++.

## Đặc điểm trong cú pháp, quy tắc lập trình

Một định danh (identifier) trong Python là một tên được sử dụng để nhận diện một biến, một hàm, một lớp, hoặc một đối tượng. Một định danh bắt đầu với một chữ cái từ A tới Z hoặc từ a tới z hoặc một dấu gạch dưới (\_) được theo sau bởi 0 hoặc nhiều ký tự, dấu gạch dưới hoặc các chữ số (từ 0 tới 9).

Python không hỗ trợ các ký tự đặc biệt chẳng hạn như @, $ và % bên trong các định danh. Python là một ngôn ngữ lập trình phân biệt chữ hoa- chữ thường, do đó định danh HOUROFCODE và hourofcode là hai định danh hoàn toàn khác nhau trong lập trình Python. Dưới đây là một số qui tắc nên được sử dụng trong khi đặt tên các định danh:

* Tên có thể là một dãy ký tự hoặc 1 dãy số bắt đầu bằng ký tự hoặc dấu gạch dưới
* Không được phép sử dụng ký tự đặc biệt để đặt tên (ngoại trừ dấu gạch dưới). Ký tự đầu tiên có thể là chữ cái, dấu gạch dưới, nhưng không được sử dụng chữ số làm ký tự đầu tiên.
* Khi đặt tên không nên đặt trùng với từ khóa trong Python (phần dưới sẽ trình bày về khác từ khóa này).
* Tên lớp bắt đầu với một chữ cái hoa. Tất cả tên khác bắt đầu với một chữ cái thường.
* Một tên được bắt đầu với một dấu gạch dưới đơn chỉ ra rằng tên (định danh) đó là private.
* Bắt đầu một định danh với hai dấu gạch dưới chỉ rằng định danh đó thực sự là private.
* Nếu định danh cũng kết thúc với hai dấu gạch dưới, thì định danh này là một tên đặc biệt được định nghĩa bởi ngôn ngữ (ví dụ như \_\_init\_\_ chẳng hạn).

## Câu lệnh rẽ nhánh

### Luồng xử lý

Nếu như trong thực tế, chúng ta thường có các trường hợp để xử lý vấn đề thì trong ngôn ngữ lập trình nó cũng như thế khái niệm này trong lập trình gọi là rẽ nhánh.

Nói một cách dễ hiểu hơn thì "nếu điều kiện này đúng thì thực hiện khối lệnh A, nếu sai thì thực hiện khối lệnh B". Ở đây các điều kiện trả về giá trị khác 0 hoặc bằng True thì coi là đúng và trả về 0, None hoặc False thì coi là sai.

### Các câu lệnh rẽ nhánh

* **Câu lệnh if-else**

If-else dịch ra tiếng Việt có nghĩa là nếu thì và nó ở trong lập trình cũng thế. Các điều kiện mà mệnh đề if đưa ra trả về là đúng (True) thì nó sẽ thực thi code bên trong mệnh đề if và ngược lại nếu điều kiện đó sai thì nó sẽ thực hiện code trong mệnh đề else.

* **Câu lệnh if-elif-else**

 Trong thực tế thì không phải lúc nào chúng ta cũng có 2 trường hợp nếu thì. Mà có đôi lúc tồn tại vô vàn điều kiện khác nhau và để giải quyết điều này thì trong Python có cung cấp thêm co chúng ta mệnh đề if-elif-else với cú pháp như sau:.

## Vòng lặp

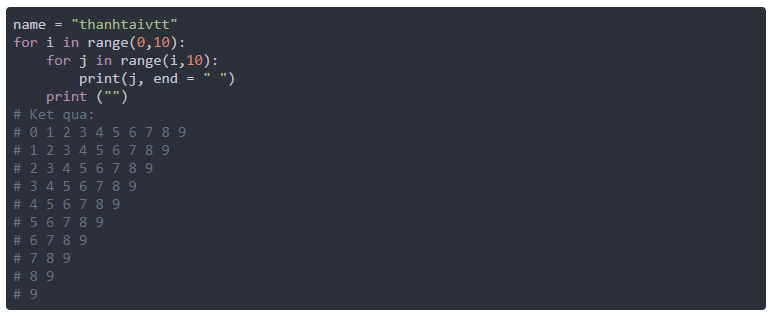
Vòng lặp (tiếng anh là loop) là một thuật ngữ dùng để diễn tả một hành động hay một cụm hành động được lặp đi lặp lại nhiều lần.

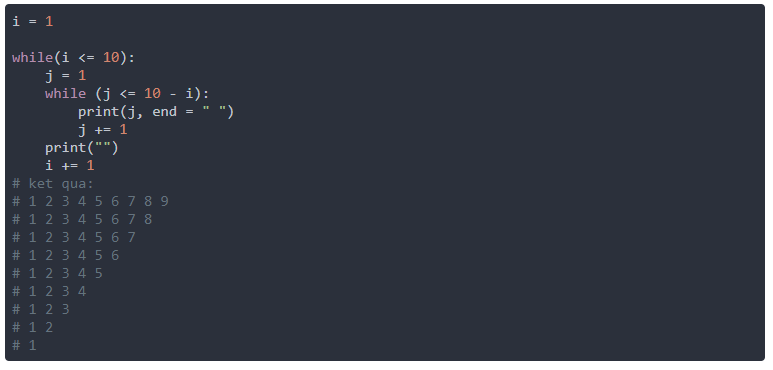
VD: Như chúng ta cứ uống nước xong là lại phải đi tiểu và hành động đó cứ xảy ra đến khi bạn die mới dừng. Thì đó cũng có thể coi là một vòng lặp.

Và trong Python thì nó cung cấp cho chúng ta 2 dạng vòng lặp chính đó là:

*for*

*while*

Ví dụ cho vòng lặp ***for***

Ví dụ cho vòng lặp ***while***

## Hàm

*Hàm là một tập các khối code được viết ra nhằm cho việc tái sử dụng code*

### Khai báo hàm trong Python

Để khai báo hàm trong Python thì chúng ta sử dụng keyword def với cú pháp như sau:

Trong đó:

ten\_ham là tên của hàm mà bạn muốn đặt. Lưu ý: tên hàm không được bắt đầu bằng số và không được chứa các ký tự đặc biệt trừ ký tự \_

Parma… là các tham số bạn muốn truyền vào hàm, nếu không có tham số thì để trống trường này.

 Ví dụ:

### Gọi hàm

Để gọi một hàm đã được khai báo rồi, thì chúng ta sử dụng cú pháp sau:

Trong đó:

Ten\_ham là tên của hàm mà chúng ta muốn gọi

Param… là các tham số chúng ta muốn truyền vào trong hàm

### Hàm có kết quả trả về

Trong trường họp bạn muốn sử dụng kết quả của hàm vừa tính để thực hiện các mục đích khác. Thì bạn chỉ cần thêm keyword return trước kết quả bạn muốn trả về.

### Truyền vô tham số vào hàm

Trên thực tế, không phải lúc nào chúng ta cũng biết được chính xác số lượng biến truyền vào trong hàm. Chính vì thế trong Python có cũng cấp cho chúng ta khai báo một param đại diện cho các biến truyền vào hàm bằng cách thêm dấu \* vào trước param đó.

## Modules

### Modules là gì?

Modules là cách mà chúng ta phân hóa chương trình ra các nhánh nhỏ cho dễ quản lý và gọi lại chúng khi nào cần, như thế chương trình của chúng ta sẽ có tính tái sử dụng, bảo trì cao.

### Modules trong Python

Để import một module có vào trong file hiện tại thì mọi người sử dụng từ khóa import với cú pháp sau:

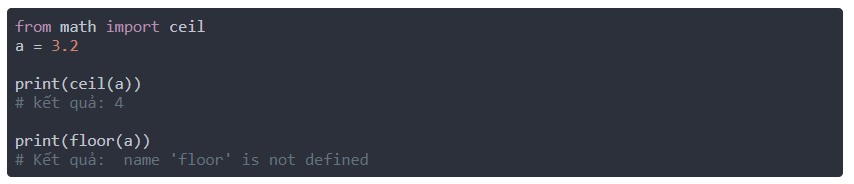
Trong đó: module1, module2,... là các modules mà các bạn muốn import vào file hiện tại

Giả sử trong một trường họp nào đó khi bạn không muốn sử dụng hết toàn bộ module mà chỉ muốn sử dụng một số thứ trong đó mà thôi thì sẽ phải làm sao ? Trong trường hợp này chúng ta sử dụng từ khóa from…import theo cú pháp như sau

Trong đó:

Modules là tên của module mà các bạn muốn import

Something,something2,… là những thứ mà bạn muốn sử dụng ở trong modules. Nếu như bạn muốn import tất cả những gì trong modules có và cho phép thì bạn sử dụng keyword \*

Ví dụ:

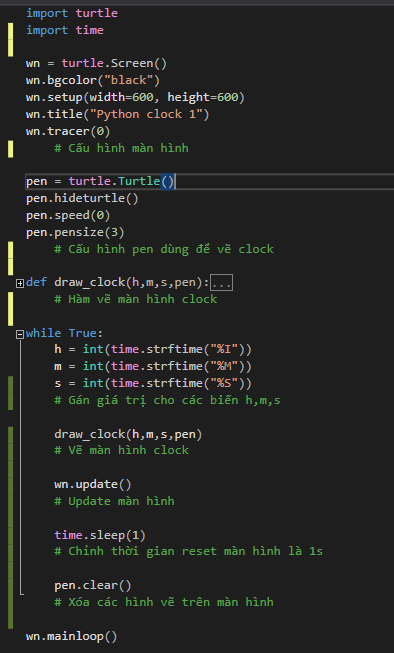
# TRÌNH BÀY PHẦN TRIỂN KHAI GIAO DIỆN

## Triền khai giao diện đồng hồ kim

Modules sử dụng:

* **Turtle** dùng để vẽ các hình đơn giản
* **Time** dùng để truy xuất thời gian thật vào trong project

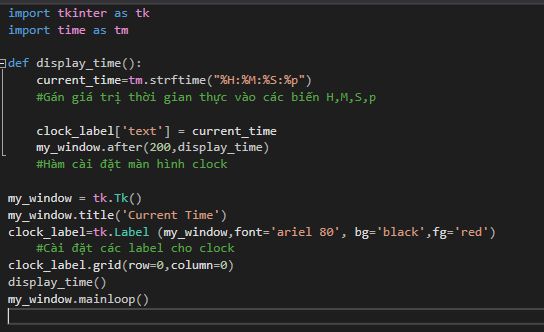
Thuật toán:



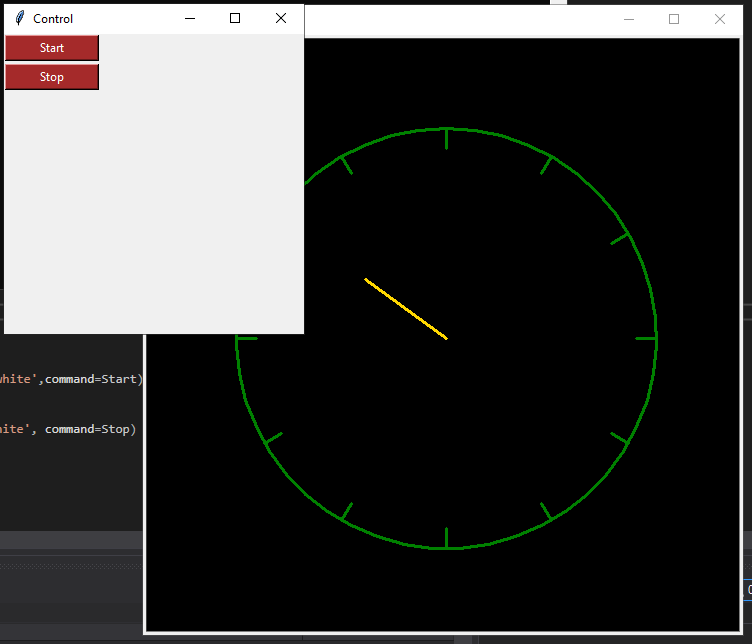
## Triển khai giao diện đồng hồ số

Modules sử dụng:

* **Tkinter** dùng để sử dụng các label có sẵn của modules
* **Time** dùng để truy xuất thời gian thật vào trong project

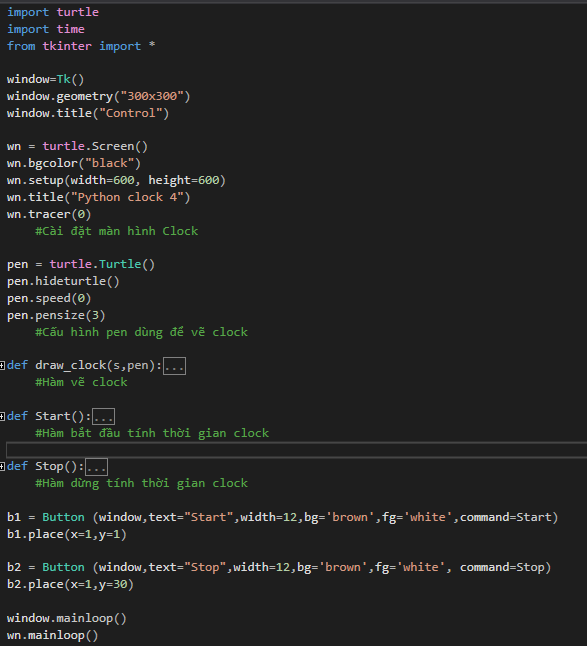
Thuật toán

## Triển khai giao diện đồng hồ bấm giờ



Modules sử dụng:

* **Tkinter** dùng để sử dụng các label có sẵn của modules
* **Time** dùng để truy xuất thời gian thật vào trong project
* **Turtle** dùng để vẽ các hình đơn giản

Thuật toán

KẾT LUẬN

**Những kết quả đạt được:**

Trang web Hôm nay ăn gì cơ bản đã hoàn thiện những chức năng cơ bản. Với giao diện đơn giản được xây dựng trên bootstrap nên trang web hỗ trợ hiển thị trên các kích thước cho các loại màn hình khác nhau, người dùng có thể dễ dàng sử dụng các tính năng mà trang web dang có sẵn. Người dùng có thể tìm kiếm các công thức nấu ăn hiện đang có trên trang web một cách dễ dàng, ngoài ra họ có thể tự tạo cho mình các công thức nấu ăn. Sau đây là các tính năng của trang web đã hoàn thành và chưa hoàn thành

|  |  |
| --- | --- |
| **Tính năng đã hoàn thành** | **Tính năng chưa hoàn thành** |
| Bài tập 1: Màn hình đồng hồ kim | Bài tập 3: Chọn màu và các hình chữ nhật, tam giác và hình tròn |
| Bài tập 2: Màn hình đồng hồ số |
| Bài tập 4: Màn hình đồng hồ bấm giờ |

**Ưu điểm:**

* Hiểu được giao diện Python
* Áp dụng thành công Python vào các giao diện đơn giản

**Nhược điểm:**

* Vẫn chưa hiểu rõ vài Modules nên dẫn đến chưa hoàn thành bài tập 3

**TÀI LIỆU THAM KHẢO**

[1]. <https://toidicode.com/modules-trong-python-354.html>

[2]. [https://quantrimang.com/python-la-gi-tai-sao-nen-chon-python-140518](https://quantrimang.com/python-la-gi-tai-sao-nen-chon-python-140518%20)