|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **BỘ CÔNG THƯƠNG** | **KHOA**  **CÔNG NGHỆ THÔNG TIN** | | **TRƯỜNG ĐẠI HỌC KINH TẾ -**  **KỸ THUẬT CÔNG NGHIỆP** |   **BÁO CÁO TỔNG KẾT**  **ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**  **NĂM HỌC 2019 - 2020**  Tên đề tài:  **NGHIÊN CỨU NGÔN NGỮ LẬP TRÌNH PYTHON. ỨNG DỤNG XÂY DỰNG PHẦN MỀM KIỂM TRA TRẮC NGHIỆM TRỰC TUYẾN, HỖ TRỢ SINH VIÊN TRONG HỌC TẬP**  **HÀ NỘI05/2020**   |  |  | | --- | --- | | **BỘ CÔNG THƯƠNG** | **KHOA**  **CÔNG NGHỆ THÔNG TIN** | | **TRƯỜNG ĐẠI HỌC KINH TẾ -**  **KỸ THUẬT CÔNG NGHIỆP** |   **BÁO CÁO TỔNG KẾT**  **ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**  **NĂM 2020**  Tên đề tài:  **NGHIÊN CỨU NGÔN NGỮ LẬP TRÌNH PYTHON. ỨNG DỤNG XÂY DỰNG PHẦN MỀM KIỂM TRA TRẮC NGHIỆM TRỰC TUYẾN, HỖ TRỢ SINH VIÊN TRONG HỌC TẬP**  **Giảng viên hướng dẫn: Th.s Nguyễn Thu Hiền**  **Chủ nhiệm đề tài: Hứa Mạnh Tuấn Lớp: ĐHTI11A5HN**  **Bùi Thị Hoa Mai Lớp: ĐHTI11A5HN**  **Bùi Xuân Vĩ Lớp: ĐHTI11A5HN**  **HÀ NỘI 05/2020** |

MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU VỀ PYTHON 11](#_Toc38491807)

[1.1 Tổng quan về Python 11](#_Toc38491808)

[1.1.1 Python là gì? 11](#_Toc38491809)

[1.1.2 Lịch sử phát triển 11](#_Toc38491810)

[1.2 Các đặc điểm của ngôn ngữ Python 13](#_Toc38491811)

[1.3 Cú pháp trong Python 17](#_Toc38491812)

[1.3.1 Biến số 17](#_Toc38491813)

[1.3.2 Toán tử số học 17](#_Toc38491814)

[1.3.3 Boolean và Toán tử logic 17](#_Toc38491815)

[1.3.4 Cấu trúc điều khiển 18](#_Toc38491816)

[1.3.4.1 If…elif…else 18](#_Toc38491817)

[1.3.4.2 Switch…case 19](#_Toc38491818)

[1.3.4.3 For…in 19](#_Toc38491819)

[1.3.4.4 While 20](#_Toc38491820)

[1.3.5 Hàm 20](#_Toc38491821)

[1.3.6 Xử lý chuỗi 21](#_Toc38491822)

[1.3.6.1 Nối chuỗi 22](#_Toc38491823)

[1.3.6.2 Trích xuất chuỗi con 22](#_Toc38491824)

[1.3.6.3 Lấy độ dài của chuỗi 22](#_Toc38491825)

[1.3.6.4 Tìm & thay thế nội dung 23](#_Toc38491826)

[1.3.6.5 Tìm vị trí chuỗi con 23](#_Toc38491827)

[1.3.6.6 Tách chuỗi 23](#_Toc38491828)

[1.3.6.7 Tìm ký tự khoẳng trắng 24](#_Toc38491829)

[1.3.6.8 Một số hàm xử lý chuỗi 24](#_Toc38491830)

[1.3.7 List 24](#_Toc38491831)

[1.3.7.1 Kiểm tra sự tồn tại của một phần tử 25](#_Toc38491832)

[1.3.7.2 Trích xuất mảng con 26](#_Toc38491833)

[1.3.7.3 Xóa phần tử của mảng 27](#_Toc38491834)

[1.3.7.4 Nối 2 mảng 27](#_Toc38491835)

[1.3.7.5 Thêm phần tử vào mảng 27](#_Toc38491836)

[1.3.7.6 Lấy phần tử cuối mảng 28](#_Toc38491837)

[1.3.7.7 Tìm một giá trị trong mảng 28](#_Toc38491838)

[1.3.7.8 Đảo ngược giá trị của mảng 29](#_Toc38491839)

[1.3.7.9 Sắp xếp giá trị các phần tử 29](#_Toc38491840)

[1.3.8 Tuple 30](#_Toc38491841)

[1.3.9 Dictionary 30](#_Toc38491842)

[1.3.9.1 Thêm một phần tử 30](#_Toc38491843)

[1.3.9.2 Một số hàm, phương thức thông dụng: 31](#_Toc38491844)

[CHƯƠNG 2: GIỚI THIỆU VỀ DJANGO FRAMEWORK 32](#_Toc38491845)

[2.1 Giới thiệu về Django 32](#_Toc38491846)

[2.2 Khả năng 32](#_Toc38491847)

[2.3 Cách sử dụng 34](#_Toc38491851)

[2.3.1 Tạo project 34](#_Toc38491852)

[2.3.2 Chạy server 35](#_Toc38491853)

[2.3.3 Tạo web app 36](#_Toc38491854)

[2.3.4 Tạo View 36](#_Toc38491855)

[2.3.5 Django - Model 38](#_Toc38491856)

[2.3.6 Tạo mô hình dữ liệu cho ứng dụng 39](#_Toc38491857)

[2.3.7 Tạo bảng trong CSDL từ Model 41](#_Toc38491858)

[2.3.8 Thao tác với các bảng 42](#_Toc38491859)

[2.3.9 Hệ thống Admin 45](#_Toc38491860)

[2.3.9.1 Tạo user 45](#_Toc38491861)

[2.3.9.2 Truy cập trang admin 45](#_Toc38491862)

[2.3.10 Django – View và Template 47](#_Toc38491863)

[2.3.10.1 Tạo View 48](#_Toc38491864)

[2.3.10.2 Template 49](#_Toc38491865)

[2.3.11 Django - Upload file 50](#_Toc38491866)

[CHƯƠNG 3: XÂY DỰNG CƠ SỞ DỮ LIỆU VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN 54](#_Toc38491867)

[3.1 Xây dựng cơ sở dữ liệu 54](#_Toc38491868)

[3.1.1 Bảng User: Lưu thông tin sinh viên 54](#_Toc38491869)

[3.1.2 Bảng Category: Lưu thông tin chuyên ngành 54](#_Toc38491870)

[3.1.3 Bảng Question: Lưu thông tin câu hỏi 55](#_Toc38491871)

[3.1.4 Bảng Quiz: Lưu thông tin môn kiểm tra 55](#_Toc38491872)

[3.1.5 Bảng Question\_quiz: Lưu thông tin các câu hỏi có trong 1 môn kiểm tra 56](#_Toc38491873)

[3.1.6 Bảng Answer: Lưu thông tin câu trả lời cho các câu hỏi 56](#_Toc38491874)

[3.1.7 Bảng Progress: Lưu thông tin tiến trình thực hiện bài làm của sinh viên 56](#_Toc38491875)

[3.1.8 Bảng Sitting: Lưu lại thông tin của bài kiểm tra khi sinh viên làm bài 57](#_Toc38491876)

[3.1.9: Biểu đồ mô tả cơ sở dữ liệu 58](#_Toc38491877)

[3.1.10 Kết nối CSDL 58](#_Toc38491878)

[3.2 Phân tích và thiết kế hệ thống thông tin 59](#_Toc38491879)

[3.2.1 Biểu đồ Usecase Quản trị tổng quan 59](#_Toc38491880)

[3.2.2 Biểu đồ Usecase Sinh viên tổng quan 59](#_Toc38491881)

[3.2.3 Usecase miêu tả đăng nhập admin 60](#_Toc38491882)

[3.1.4 Usecase thay đổi mật khẩu admin 60](#_Toc38491883)

[3.2.5 Usecase mô tả Quản lí đề thi 61](#_Toc38491884)

[3.2.6 Usecase mô tả đăng nhập của sinh viên 62](#_Toc38491885)

[3.2.7 Usecase thực hiện bài kiểm tra 62](#_Toc38491886)

[3.2.8 Sơ đồ phân rã chức năng 63](#_Toc38491887)

[3.2.8.1 Sơ đồ phân rã chức năng tổng quát 63](#_Toc38491888)

[3.2.8.2 Sơ đồ phân rã chức năng Admin 64](#_Toc38491889)

[3.2.9 Sơ đồ luồng dữ liệu 65](#_Toc38491890)

[3.2.9.1 Sơ đồ luồng dữ liệu Admin 65](#_Toc38491891)

[3.2.9.2 Sơ đồ luồng dữ liệu Sinh viên 65](#_Toc38491892)

[3.2.10 Mô hình quan hệ thực thể 66](#_Toc38491893)

[CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG 67](#_Toc38491894)

[4.1 Tổng quan giao diện 67](#_Toc38491895)

[4.2 Giao diện 67](#_Toc38491896)

[4.2.1 Giao diện trang chủ Website 67](#_Toc38491897)

[4.2.2 Giao diện trang đăng nhập 68](#_Toc38491898)

[4.2.3 Giao diện trang đăng kí 69](#_Toc38491899)

[4.2.4 Giao diện trang thay đổi mật khẩu 70](#_Toc38491900)

[4.2.5 Giao diện trang hồ sơ cá nhân 71](#_Toc38491901)

[4.2.6 Giao diện trang giới thiệu 72](#_Toc38491902)

[4.2.7 Giao diện trang môn thi 73](#_Toc38491903)

[4.2.8 Giao diện trang kiểm tra 74](#_Toc38491904)

[4.2.9 Giao diện trang kết quả 74](#_Toc38491905)

[4.2.10 Giao diện trang lịch sử làm bài 75](#_Toc38491906)

[KẾT LUẬN 77](#_Toc38491907)

[TÀI LIỆU THAM KHẢO 78](#_Toc38491908)

**DANH MỤC HÌNH**

[Hình 3. 1: Biểu đồ mô tả cơ sở dữ liệu 58](#_Toc38491477)

[Hình 3. 2: Biểu đồ Usecae Quản trị tổng quan 59](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491478)

[Hình 3. 4: Biểu đồ Usecase Sinh viên tổng quan 59](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491480)

[Hình 3. 5: Usecase miêu tả đăng nhập admin 60](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491481)

[Hình 3. 6: Usecase thay đổi mật khẩu admin 60](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491482)

[Hình 3. 7: Usecase mô tả Quản lí đề thi 61](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491483)

[Hình 3. 8: Usecase mô tả đăng nhập của sinh viên 62](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491484)

[Hình 3. 9: Usecase thực hiện bài kiểm tra 63](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491485)

[Hình 3. 10: Sơ đồ phân rã chức năng tổng quát 63](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491486)

[Hình 3. 11: Sơ đồ phân rã chức năng Admin 64](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491487)

[Hình 3. 12: Sơ đồ luồng dữ liệu Admin 65](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491488)

[Hình 3. 13: Sơ đồ luồng dữ liệu Sinh viên 65](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491489)

[Hình 3. 14: Mô hình quan hệ thực thể 66](file:///D:\Góc%20học%20tập%20của%20Mai%20ahihi\Năm%203(%20kì%202)\Lập%20trình%20C%23\NCKH_Web%20Python_15_4(bs).docx#_Toc38491490)

[Hình 4. 1: Giao diện trang chủ Website 68](#_Toc38491497)

[Hình 4. 2: Giao diện trang đăng nhập 68](#_Toc38491498)

[Hình 4. 3: Giao diện trang đăng kí 69](#_Toc38491499)

[Hình 4. 4: Giao diện đăng kí thành công 70](#_Toc38491500)

[Hình 4. 5: Giao diện tài khoản được đăng nhập 70](#_Toc38491501)

[Hình 4. 6: Giao diện trang thay đổi mật khẩu 71](#_Toc38491502)

[Hình 4. 7: Giao diện trang hồ sơ cá nhân 72](#_Toc38491503)

[Hình 4. 8: Giao diện trang giới thiệu 72](#_Toc38491504)

[Hình 4. 9: Giao diện trang môn thi 73](#_Toc38491505)

[Hình 4. 10: Giao diện trang kiểm tra 74](#_Toc38491506)

[Hình 4. 11: Giao diện trang kết quả 75](#_Toc38491507)

[Hình 4. 12: Giao diện trang lịch sử làm bài 75](#_Toc38491508)

**DANH MỤC BẢNG**

[Bảng 3.1: Bảng User 54](#_Toc38491575)

[Bảng 3. 2: Bảng Categogy 54](#_Toc38491576)

[Bảng 3. 3: Bảng Question 55](#_Toc38491577)

[Bảng 3. 4: Bảng Quiz 56](#_Toc38491578)

[Bảng 3. 5: Bảng Question\_quiz 56](#_Toc38491579)

[Bảng 3. 6: Bảng Answer 56](#_Toc38491580)

[Bảng 3. 7: Bảng Progress 57](#_Toc38491581)

[Bảng 3. 8: Bảng Sitting 58](#_Toc38491582)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Giải nghĩa** |
| 1. | NCKH-CN | Nghiên cứu khoa học-Công nghệ |
| 2. | NCKH | Nghiên cứu khoa học |
| 3. | CSDL | Cơ sở dữ liệu |
| 4. |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |

**LỜI MỞ ĐẦU**

Trong những năm gần đây, hình thức thi trắc nghiệm đã trởnên phổbiến trên toàn thế giới, với những ưu điểm tuyệt vời của mình, nó đã dần trởthành hình thức thi chính của các tổchức giáo dục quốc tế. Nhận thấy những ưu điểm của hình thức thi này, rất nhiều trường Đại học, Cao đẳng, các trường từ Tiểu học đến THPT đã quyết định đưa trắc nghiệm vào trong các kì thi đánh giá năng lực cũng như hỗ trợ học sinh, sinh viên trong quá trình học tập. Khi mà mạng internet phát triển ngày lớn mạnh, thếgiới đã được đem đến từng nhà qua chiếc máy tính thì cùng với nó, những dịch vụ hỗtrợngười dùng ra đời ngày càng nhiều hơn, và những website trắc nghiệm trực tuyến ra đời cũng không ngoài mục đích là giúp đỡhọc sinh, sinh viên nước ta có được sựhỗtrợcao nhất, thuận lợi nhất đểlàm quen với hình thức thi này, đểnền giáo dục nước ta tiến dần đến chuẩn quốc tế.

Thực hiện nghị quyết của BCH Đảng bộ, Ban giám hiệu nhà trường về việc triển khai các hoạt động NCKH-CN năm học 2019-2020, chúng em đã tham gia NCKH với đề tài **“Xây dựng ứng dụng phần mềm kiểm tra trắc nghiệm trực tuyến hỗ trợ sinh viên trong học tập”**mong ước được cùng với các trang web trắc nghiệm trực tuyến khác, góp một phần nhỏvào việc giúp đỡhọc sinh, sinh viên nước nhà hiện nay.

Cùng với sự nỗ lực của nhóm và không thể thiếu sự hướng dẫn nhiệt tình của cô **Nguyễn Thu Hiền**, chúng em đã hoàn thành đề tài của mình một cách tốt nhất. Tuy nhiên không thể tránh khỏi những thiếu sót, chúng em kính mong các thầy cô thông cảm và đóng góp ý kiến để đề tài của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

# CHƯƠNG 1: GIỚI THIỆU VỀ PYTHON

# 1.1Tổng quan về Python

## 1.1.1 Python là gì?

* Python là một ngôn ngữ lập trình bậc cao, thông dịch, hướng đối tượng, đa mục đích và cũng là một ngôn ngữ lập trình động.
* Cú pháp của Python là khá dễ dàng để học và ngôn ngữ này cũng mạnh mẽ và linh hoạt không kém các ngôn ngữ khác trong việc phát triển các ứng dụng. Python hỗ trợ mẫu đa lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và mệnh lệnh hoặc là các phong cách lập trình theo thủ tục.
* Python không chỉ làm việc trên lĩnh vực đặc biệt như lập trình web, và đó là tại sao ngôn ngữ này là đa mục đích bởi vì nó có thể được sử dụng với web, enterprise, 3D CAD, …
* Bạn không cần sử dụng các kiểu dữ liệu để khai báo biến bởi vì kiểu của nó là động, vì thế bạn có thể viết a=15 để khai báo một giá trị nguyên trong một biến.
* Với Python, việc phát triển ứng dụng và debug trở nên nhanh hơn bởi vì không cần đến bước biên dịch và chu trình edit-test-debug của Python là rất nhanh

## 1.1.2 Lịch sử phát triển

* Python là một ngôn ngữ lập trình thông dịch do Guido van Rossum tạo ra năm 1990. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.
* TheođánhgiácủaEric S.Raymond, Pythonlàngônngữcóhìnhthức rất sang sủa,cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép ngườisửdụngviếtmãlệnhvớisốlầngõphímtốithiểu,nhưnhậnđịnhcủachínhGuido van Rossum trong một bài phỏng vấn ông.
* Banđầu,Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, nó đã “bành trướng” sang mọi hệ điều hành từ MS-DOS đến MacOS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guidovan Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.
* Sự phát triển Python đến nay có thể chia làm các giai đoạn:
* Python 1: bao gồm các bản phát hành 1.x. Giai đoạn này, kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995, Guido làm việc tại CWI (Centrum voor Wiskunde en Informatica - Trung tâm Toán-Tin học tại Amsterdam, Hà Lan). Vì vậy, các phiên bản Python đầu tiên đều do CWI phát hành. Phiên bản cuối cùng phát hành tại CWI là 1.2.

Vào năm 1995, Guido chuyển sang CNRI (Corporation for National Research Initiatives) ở Reston, Virginia. Tại đây, ông phát hành một số phiên bản khác. Python 1.6 là phiên bản cuối cùng phát hành tại CNRI.

Sau bản phát hành 1.6, Guido rời bỏ CNRI để làm việc với các lập trình viên chuyên viếtphầnmềmthươngmại.Tạiđây,ôngcóýtưởngsửdụngPythonvớicácphầnmềm tuântheochuẩnGPL.Sauđó,CNRIvàFSF(FreeSoftwareFoundation-Tổchứcphần mềm tự do) đã cùng nhau hợp tác để làm bản quyền Python phù hợp với GPL. CùngPython (ngôn ngữ lập trình nămđó,GuidođượcnhậnGiảithưởngFSFvìSựpháttriểnPhầnmềmtựdo(Awardfor the Advancement of Free Software).

Phiênbản1.6.1rađờisauđólàphiênbảnđâutiềntuântheobảnquyềnGPL.Tuynhiên, bản này hoàn toàn giống bản 1.6, trừ một số sửa lỗi cần thiết.

* Python 2: vào năm 2000, Guido và nhóm phát triển Python dời đến BeOpen.com và thành lập BeOpen PythonLabs team. Phiên bản Python 2.0 được phát hành tại đây. Sau khi phát hành Python 2.0, Guido và các thành viên PythonLabs gia nhập Digital Creations.

Python 2.1 ra đời kế thừa từ Python 1.6.1 và Python 2.0. Bản quyền của phiên bản này được đổi thành Python Software Foundation License. Từ thời điểm này trở đi, Python thuộc sở hữu của Python Software Foundation (PSF), một tổ chức phi lợi nhuận được thành lập theo mẫu Apache Software Foundation.

* Python 3, còn gọi là Python 3000 hoặc Py3K: Dòng 3.x sẽ không hoàn toàn tương thích với dòng 2.x, tuy vậy có công cụ hỗ trợ chuyển đổi từ các phiên bản 2.x sang 3.x. NguyêntắcchủđạođểpháttriểnPython3.xlà"bỏcáchlàmviệccũnhằmhạnchếtrùng lặp về mặt chức năng của Python". Trong PEP (Python Enhancement Proposal) có mô tảchitiếtcácthayđổitrongPython.CácđặcđiểmmớicủaPython3.0sẽđượctrìnhbày phần cuối bài này.

# 1.2 Các đặc điểm của ngôn ngữ Python

* **Ưu điểm:**
* Dễ dàng để sử dụng: Python là một ngôn ngữ bậc cao rất dễ dàng để sử dụng. Python có một số lượng từ khóa ít hơn, cấu trúc của Python đơn giản hơn và cú pháp của Python được định nghĩa khá rõ ràng, … Tất cả các điều này là Python thực sự trở thành một ngôn ngữ thân thiện với lập trình viên.
* Bạn có thể đọc code của Python khá dễ dàng. Phần code của Python được định nghĩa khá rõ ràng và rành mạch.
* Python có một thư viện chuẩn khá rộng lớn. Thư viện này dễ dàng tương thích và tích hợp với UNIX, Windows, và Macintosh.
* Python là một ngôn ngữ thông dịch. Trình thông dịch thực thi code theo từng dòng (và bạn không cần phải biên dịch ra file chạy), điều này giúp cho quá trình debug trở nên dễ dàng hơn và đây cũng là yếu tố khá quan trọng giúp Python thu hút được nhiều người học và trở nên khá phổ biến.
* Python cũng là một ngôn ngữ lập trình hướng đối tượng. Ngoài ra, Python còn hỗ trợ các phương thức lập trình theo hàm và theo cấu trúc.
* Ngoài các đặc điểm trên, Python còn khá nhiều đặc điểm khác như hỗ trợ lập trình GUI, mã nguồn mở, có thể tích hợp với các ngôn ngữ lập trình khác, …
* **Các bản hiện thực:**
* Python được viết từ những ngôn ngữ khác, tạo ra những bản hiện thực khác nhau. Bản hiện thực Python chính, còn gọi là CPython, được viết bằng C, và được phân phối kèm một thư viện chuẩn lớn được viết hỗn hợp bằng C và Python. CPython có thể chạy trên nhiều nền và khả chuyển trên nhiều nền khác. Dưới đây là các nền trên đó, CPython có thể chạy.

• Các hệ điều hành họ Unix: AIX, Darwin, FreeBSD, Mac OS X, NetBSD, Linux, OpenBSD, Solaris,…

• Các hệ điều hành dành cho máy desktop: Amiga, AROS, BeOS, Mac OS 9, Microsoft Windows, OS/2, RISC OS.

• Các hệ thống nhúng và các hệ đặc biệt: GP2X, Máy ảo Java, Nokia 770 Internet Tablet, Palm OS, PlayStation 2, PlayStation Portable, Psion, QNX, Sharp Zaurus, Symbian OS, Windows CE/Pocket PC, Xbox/XBMC, VxWorks.

• Các hệ máy tính lớn và các hệ khác: AS/400, OS/390, Plan 9 from Bell Labs, VMS, z/OS.

* Ngoài CPython, còn có hai hiện thực Python khác: Jython cho môi trường Java và IronPython cho môi trường .NET và Mono.
* **Khả năng mở rộng**
* Python có thể được mở rộng: nếu ta biết sử dụng C, ta có thể dễ dàng viết và tích hợp vào Python nhiều hàm tùy theo nhu cầu. Các hàm này sẽ trở thành hàm xây dựng sẵn (built-in)củaPython.Tacũngcóthểmởrộngchứcnăngcủatrìnhthôngdịch,hoặcliên kết các chương trình Python với các thư viện chỉ ở dạng nhị phân (như các thư viện đồ họa do nhà sản xuất thiết bị cung cấp).
* Hơn thế nữa, ta cũng có thể liên kết trình thông dịchcủaPythonvớicácứngdụngviếttừCvàsửdụngnónhưlàmộtmởrộnghoặcmột ngôn ngữ dòng lệnh phụ trợ cho ứng dụng đó.
* **Trình thông dịch**
* Pythonlàmộtngônngữlậptrìnhdạngthôngdịch,dođócóưuđiểmtiếtkiệmthờigian phát triển ứng dụng vì không cần phải thực hiện biên dịch và liên kết.
* Trình thông dịch cóthểđượcsửdụngđểchạyfilescript,hoặccũngcóthểđượcsửdụngtheocáchtương tác. Ởchếđộtương tác, trìnhthôngdịchPythontương tựshell củacác hệđiềuhànhhọ Unix,tại đó,tacóthểnhậpvàotừngbiểuthức rồigõEnter, vàkếtquảthực thisẽđược hiển thị ngaylập tức. Đặc điểm nàyrất hữuích chongười mới học,giúphọnghiên cứu tínhnăngcủangônngữ;hoặcđểcáclậptrìnhviênchạythửmãlệnhtrongsuốtquátrình pháttriểnphầnmềm.Ngoàira,cũngcóthểtậndụngđặcđiểmnàyđểthựchiệncácphép tính như với máy tính bỏ túi.
* **Module**
* Python cho phép chia chương trình thành các module để có thể sử dụng lại trong các chương trình khác.
* Nó cũng cung cấp sẵn một tập hợp các modules chuẩn mà lập trình viêncóthểsửdụnglạitrongchươngtrìnhcủahọ.Cácmodulenàycungcấpnhiềuchức nănghữuích,nhưcáchàmtruyxuấttậptin,cáclờigọihệthống,trợgiúplậptrìnhmạng (socket),…
* **Đa năng**
* Python là một ngôn ngữ lập trình đơn giản nhưng rất hiệu quả.

• So với Unix shell, Python hỗ trợ các chương trình lớn hơn và cung cấp nhiều cấu trúc hơn.

• So với C, Python cung cấp nhiều cơ chế kiểm tra lỗi hơn. Nó cũng có sẵn nhiều kiểu dữ liệu cấp cao, ví dụ như các mảng (array) linh hoạt và từ điển (dictionary) mà ta sẽ phải mất nhiều thời gian nếu viết bằng C.

* Pythonlàmộtngônngữlậptrìnhcấpcaocóthểđápứngphầnlớnyêucầucủalậptrình viên:

• Python thích hợp với các chương trình lớn hơn cả AWK và Perl.

• Python được sử dụng để lập trình Web. Nó có thể được sử dụng như một ngôn ngữ kịch bản.

• Python được thiết kế để có thể nhúng và phục vụ như một ngôn ngữ kịch bản để tuỳ biến và mở rộng các ứng dụng lớn hơn.

• Python được tích hợp sẵn nhiều công cụ và có một thư viện chuẩn phong phú, Python cho phép người dùng dễ dàng tạo ra các dịch vụ Web, sử dụng các thành phần COM hay CORBA, hỗ trợ các loại định dạng dữ liệu Internet như email, HTML, XML và các ngôn ngữ đánh dấu khác. Python cũng được cung cấp các thư viện xử lý các giao thức Internet thông dụng như HTTP, FTP,…

• Python có khả năng giao tiếp đến hầu hết các loại cơ sở dữ liệu, có khả năng xử lí văn bản, tài liệu hiệu quả, và có thể làm việc tốt với các công nghệ Web khác.

• Python đặc biệt hiệu quả trong lập trình tính toán khoa học nhờ các công cụ Python Imaging Library, pyVTK, MayaVi 3D Visualization Toolkits, Numeric Python, ScientificPython,…

• Python có thể được sử dụng để phát triển các ứng dụng desktop. Lập trình viên có thể dùng wxPython, PyQt, PyGtk để phát triển các ứng dụng giao diện đồ họa (GUI) chất lượng cao. Python còn hỗ trợ các nền tảng phát triển phần mềm khác như MFC, Carbon, Delphi, X11, Motif, Tk, Fox, FLTK, …

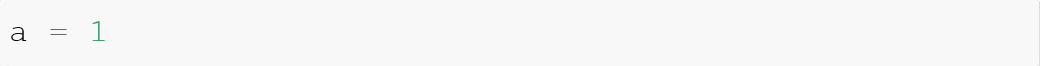
• Python cũng có sẵn một unit testing framework để tạo ra các các bộ test (test suites).

* **Multiple paradigms (đa biến hóa)**
* Pythonlàmộtngônngữđabiếnhóa(multiple paradigms). Cónghĩalà,thayvìépbuộc mọi người phải sử dụng duy nhất một phương pháp lập trình, Python lại cho phép sử dụngnhiều phương pháplập trình khác nhau: hướng đốitượng, cócấu trúc, chức năng, hoặcchỉhướngđếnmộtkhíacạnh.
* Pythonkiểuđộngvàsửdụngbộthugomrácđể quản lí bộ nhớ.
* Một đặc điểm quan trọng nữa của Python là giải pháp tên động, kết nối tên biến và tên phương thức lại với nhau trong suốt thực thi của chương trình.

# 1.3 Cú pháp trong Python

## 1.3.1 Biến số

Khai báo biến bằng một câu lệnh gán.



Bạn có thể gán nhiều loại giá trị (số, chuỗi) cho một biến.



## 1.3.2 Toán tử số học

Python cũng hỗ trợ một số toán tử toán học thông dụng như:

+ phép cộng

- phép trừ

\* phép nhân

/ phép chia

% phép chia lấy dư (modulo)

## 1.3.3 Boolean và Toán tử logic

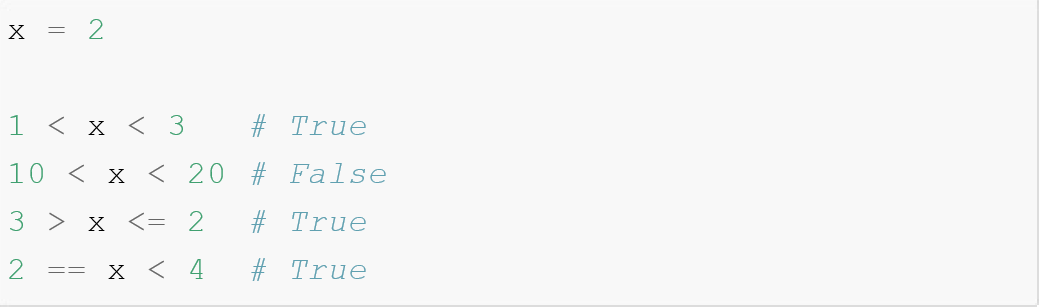
-Giá trị đúng và sai tương ứng là True vàFalse

. not: đểđảo giá trị

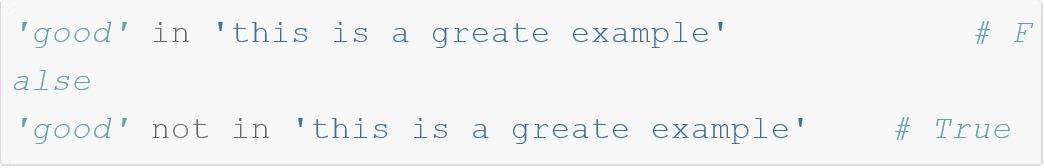
. and: phép tính logic và (AND)

. or: phéptínhlogic hoặc (OR)

* Một số phép so sánh thông thường như < (bé hơn), <= (bé hơn hoặc bằng), >(lớn hơn), >= (lớn hơn hoặc bằng), == (bằng), != (khác) để so sánh 2giá trị. Hỗtrợdạng so sánh kép như:



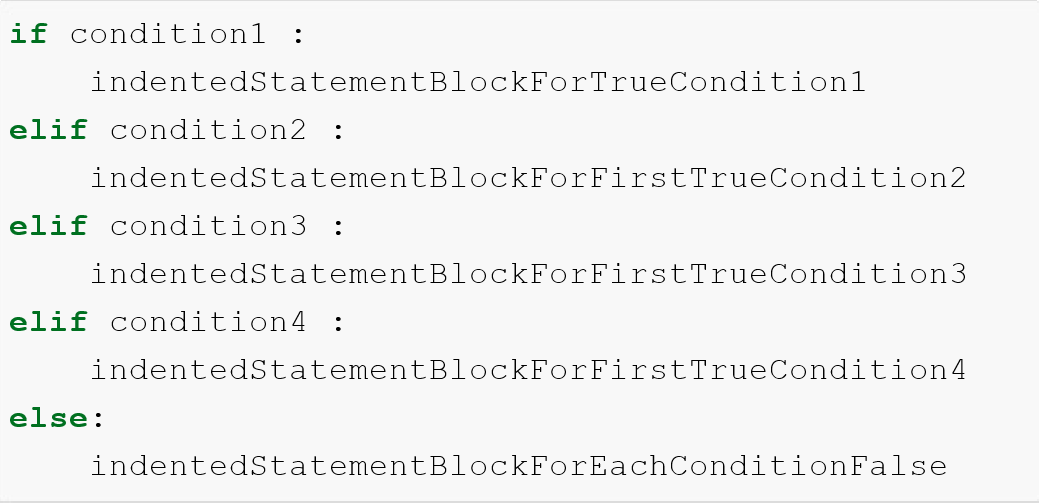
Toán tử kiểm tra phần tử trong một tập hợp: - in kiểm tra có tồn tại - not in kiểm không tồn tại



## 1.3.4 Cấu trúc điều khiển

Python hỗ trợ một số cấu trúc điều khiển thông dụng. Hầu hết các cấu trúc điều khiển đều dựa vào thụt đầu dòng (indention) để tạo thành một block xử lý, thay vì sử dụng {…} như các ngôn ngữ khác (PHP, Javascript) ###

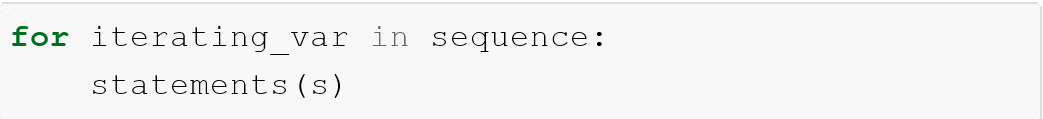
### 1.3.4.1 If…elif…else



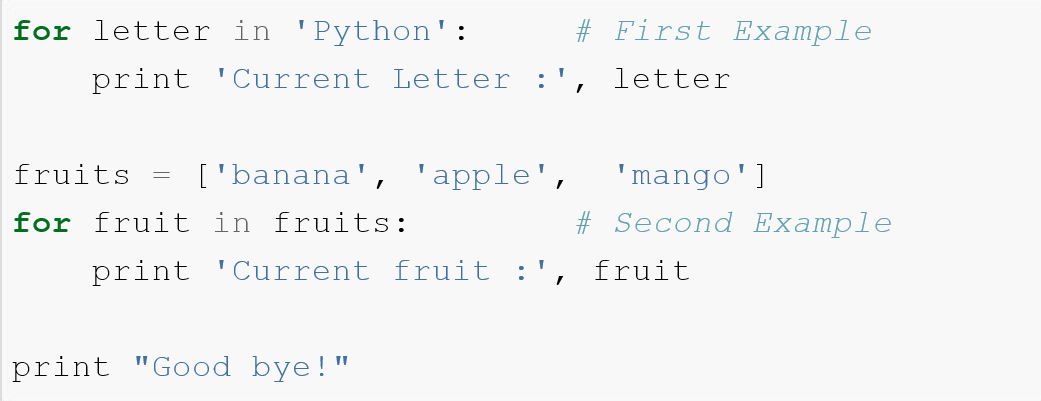
### 1.3.4.2 Switch…case

Python không có cấu trúc switch…case

### 1.3.4.3 For…in



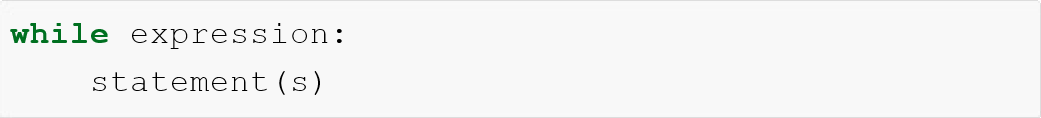
Ví dụ:



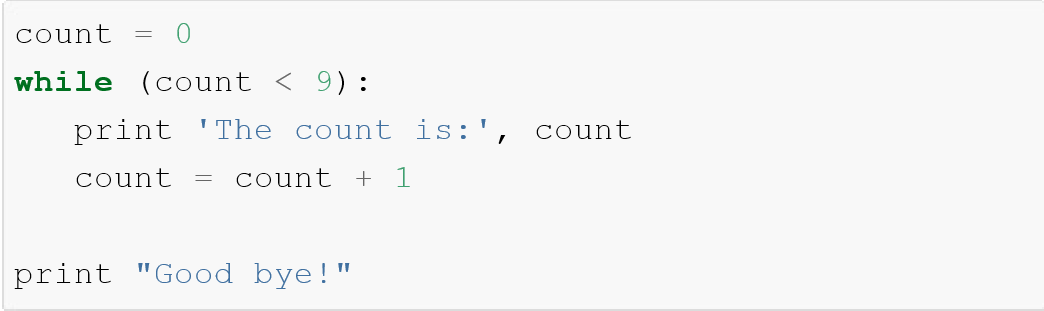
Kết quảhiển thịcủa ví dụtrên:



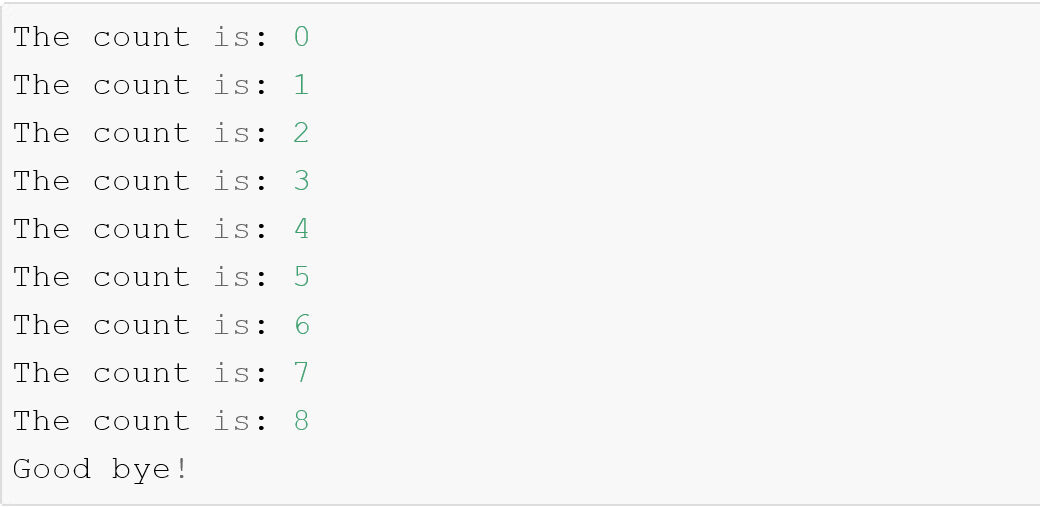
### 1.3.4.4 While



Ví dụ:

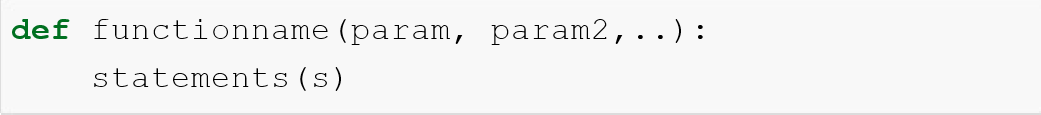


Kết quả hiển thị của ví dụ trên:



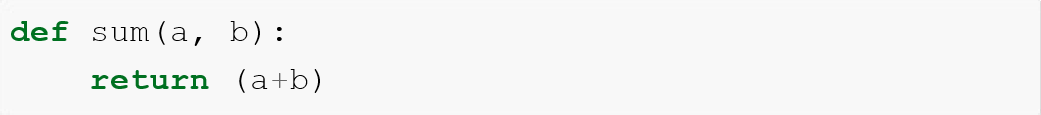
## 1.3.5 Hàm

Khai báo hàm theo cú pháp:

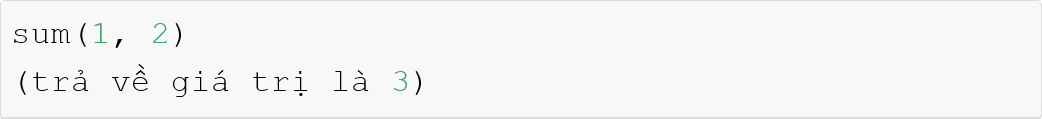


Hàm nếu khôngtrảdữliệu thì mặc định sẽtrảvề giá trị None

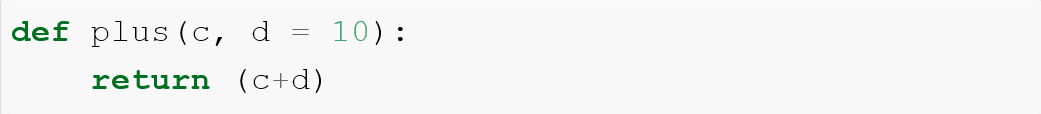
Ví dụ khai báo hàm tính và trảvề giá trịtổng của 2 tham sốđầu vào:



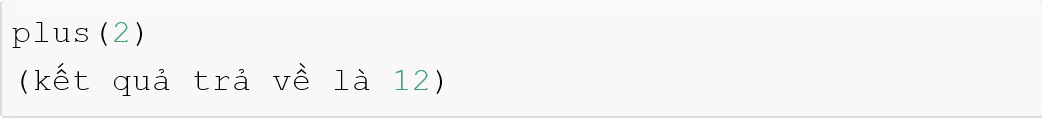
Cách gọi hàm:



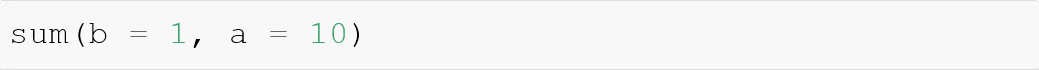
Hàm có hỗtrợ giá trịmặc định chotham số khi không truyền vào. Ví dụ hàm sau:



Nếu gọi hàm trên như sau:

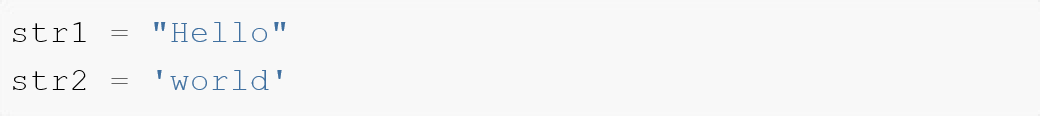


Một khác biệt trong cách gọi hàm của Python so với PHP là chúng ta có thể thay đổi thứtự tham sốtruyền vào bằng cách đặt tên tham số khi gọi hàm.Ví dụta có thểgọi hàm sum (a,b) ở ví dụtrên bằng cách truyền tham sốb trước a như sau:

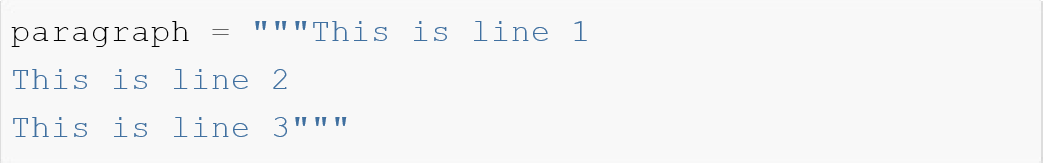


## 1.3.6Xử lý chuỗi

Một chuỗi có thể khai báo bằng dấu nháy đôi " hoặc đơn '. Ví dụ các chuỗi sau:

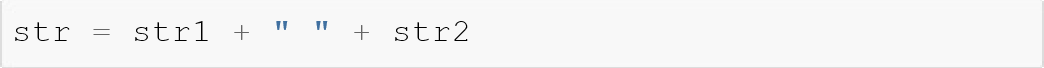


Có thể truy xuấttừng ký tự trong một chuỗi theo hình thức index, ví dụ: str1[0], str1[1]… Có thểsửdụng 3 dấu nháy (đôi hoặc đơn) để khai báo chuỗi trên nhiều dòng. Ví dụ:



### 1.3.6.1 Nối chuỗi

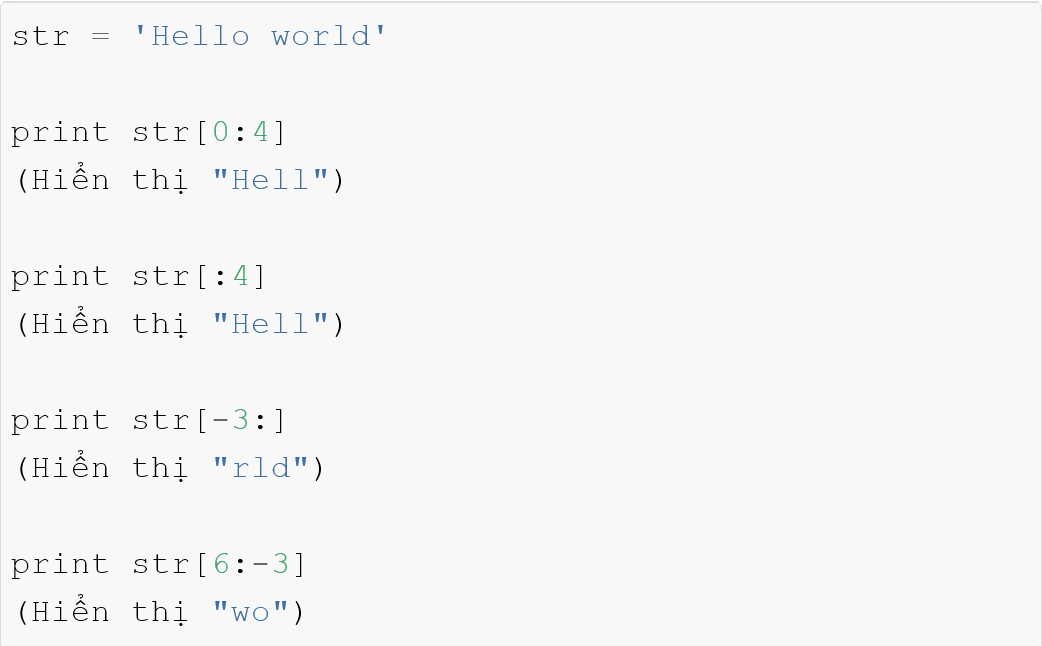
Cóthểtạo một chuỗi dài từviệc nối các chuỗi lại theo cú pháp:



### 1.3.6.2 Trích xuất chuỗi con

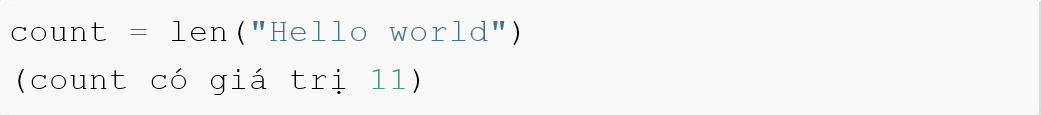
Có thểtạo các chuỗi con thông qua toán tửlấy khoản [start:end] (range). Mặc định start là từvị trí đầu chuỗi (0) và end là đến vị trí cuối chuỗi.

Ví dụ:



### 1.3.6.3 Lấy độ dài của chuỗi

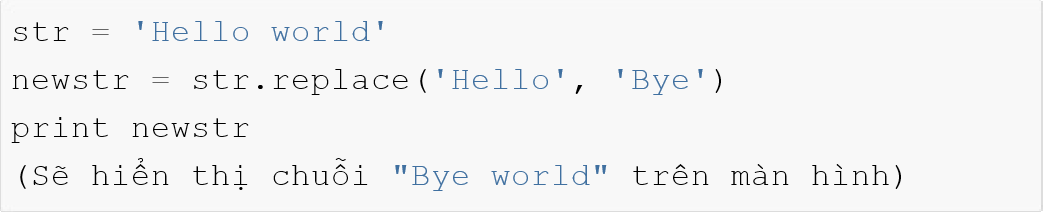
Sử dụng hàm len(...) đểtrảvềđộ dài của chuỗi.Ví dụ:



### 1.3.6.4 Tìm & thay thếnội dung

Có thể tìm và thay thế trong chuỗi bằng cách gọi phươngthức replace(search, replace,[max]) của một chuỗi.

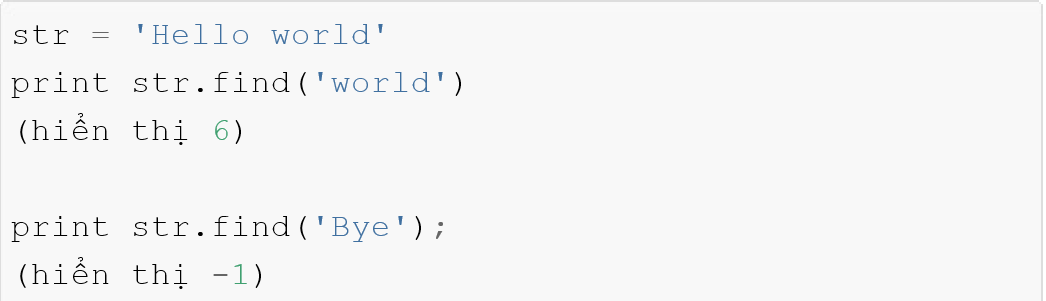
Ví dụ:



### 1.3.6.5 Tìm vịtríchuỗi con

Có thể tìm vịtrí của một chuỗi con trong chuỗi lớn bằng cách gọi phương thức find (str, beg=0 end=len(string)). Bắt đầu là vị trí 0, nếu không tìm ra thì trả về -1.

Ví dụ:



Hàm find () sẽtìm theo thứtựtừ trái qua phải của chuỗi,tức là từlần xuất hiện đầu tiên. Có thể dùng hàm find() để tìm theo vị trí từcuối chuỗi về phía trước.

### 1.3.6.6 Tách chuỗi

Có thể tách chuỗi dựa theo mộtchuỗi delimeter bằng cách gọi phương thức split (str="", num=string.count(str)). Ví dụ:

Cóthểsửdụng hàm splitlines () để tách chuỗi theo từng hàng và loại bỏ ký tựNEWLINE.

### 1.3.6.7 Tìm ký tựkhoẳng trắng

Cóthểloại bỏ các ký tự(mặc định là ký tựkhoảng trắng) trước và sau một chuỗi, bằng cách gọi các phương thức sau:

* strip([chars]): loại bỏtrước và sau chuỗi
* lstrip([chars]): loại bỏ phía trước chuỗi
* rstrip([chars]): loại bỏ phía sau chuỗi

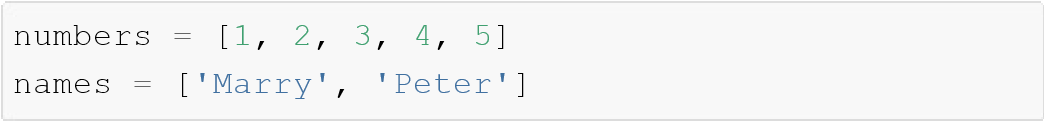
### 1.3.6.8 Một số hàm xử lý chuỗi

* isnumeric(): Kiểm tra một chuỗi có phải là chuỗi số
* lower(): Chuyển chuỗi hết thành chữthường
* upper(): Chuyển chuỗi hết thành chữHOA

## 1.3.7 List

List trong Python là cấu trúc mảng và cácphầntử có index có thứtự. Không như PHP, key của một mảng có thểvừa là số, vừa là chuỗi (associated array). Trong Python, muốn tạo một mảng có key là chuỗi thì sẽ sửdụng cấu trúc Dictionary (phần tiếp tiếp). Trong phần này, chúng ta sẽ nói đến List. Một List được khai báo như mảng trong JSON. Sử dụng [..] để khai báo một mảng.

Ví dụ:



Có thể truy xuất từng phần tửcủa mảng bằng index, phần tửđầu tiên có thứtự là 0.

Ví dụ:



Để biết được sốlượng phần tửcủa 1 List, có thểsửdụng hàm len(array) đểlấy sốlượng phần tửcủa mảng tham sốtruyền vào.

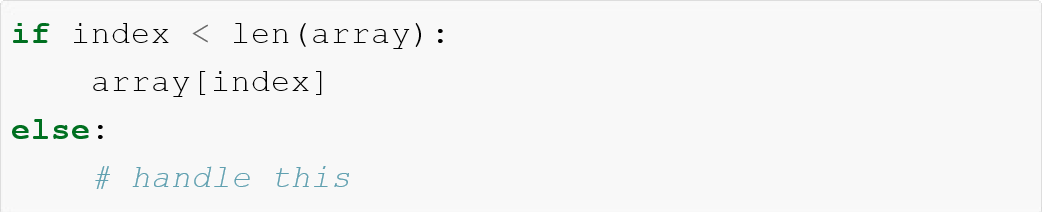
### 1.3.7.1 Kiểm tra sựtồn tại của một phần tử

#### 1.3.7.1.1 Kiểm tra theo Index

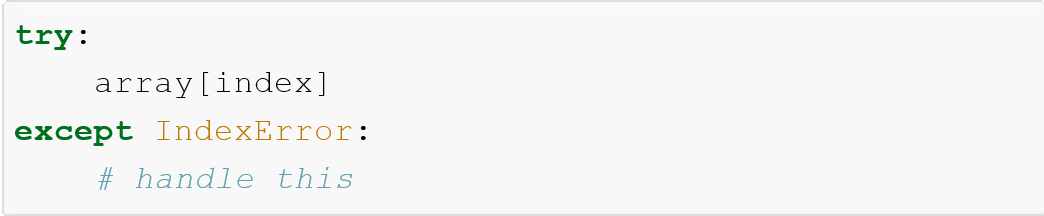
Trong nhiều trường hợp bạn muốn truy xuất một phần tửbất kỳ(dựa vào index) của mảng thì nếu truy xuất đến một phần tử không tồn tại thì ứng dụng sẽ báo lỗi. Do đó, trước khi truy xuất một phần tử, bạn cần kiểm tra xem phần tử này đã tồn tại hay chưa. Hiện tại python không hỗtrợ hàm nào đểkiểm tra sựtồn tại của một phần tử trong mảng.

Có 2 cách thường thấy đểkiểm tra đó là “Look before you leap” (LBYL) và “Easier to ask forgiveness than permission” (EAFP).

Ví dụvề “Look before you leap(LBYL)”:

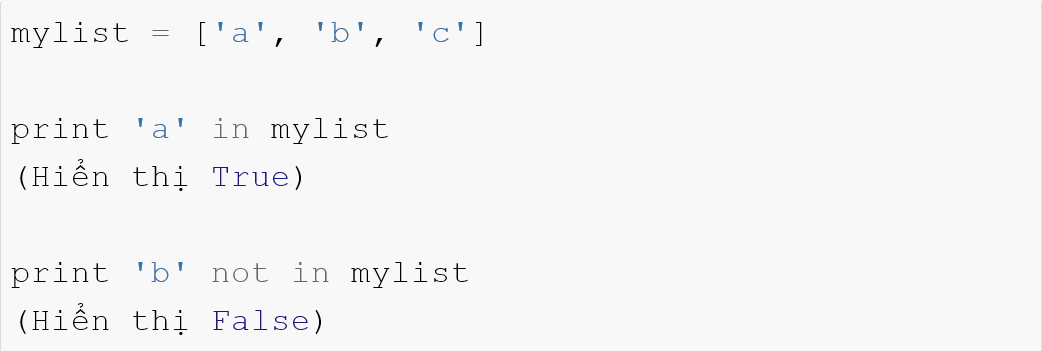


Ví dụvề “Easier to ask forgiveness than permission”(EAFP):



#### 1.3.7.1.2 Kiểm tra theo giá trị

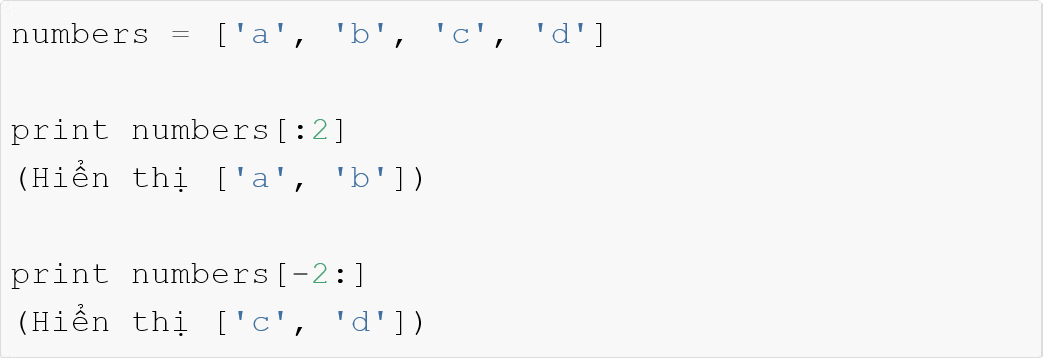
Đểkiểm tra một giá trị có tồn tại / không tồn tại trong mảng hay không thì có thể sửdụng toán tử in / not in. Ví dụ:



### 1.3.7.2Trích xuất mảng con

Tương tự như chuỗi, có thểtạo các mảng con thông qua toántửlấy khoản [start:end] (range). Mặc định start làtừvị trí đầu chuỗi (0) và end là đến vị trí cuối chuỗi.

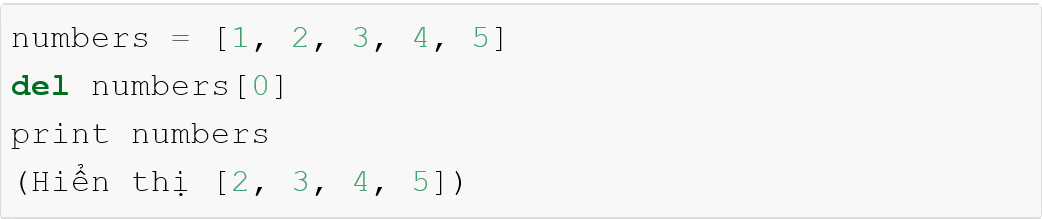
Ví dụ:



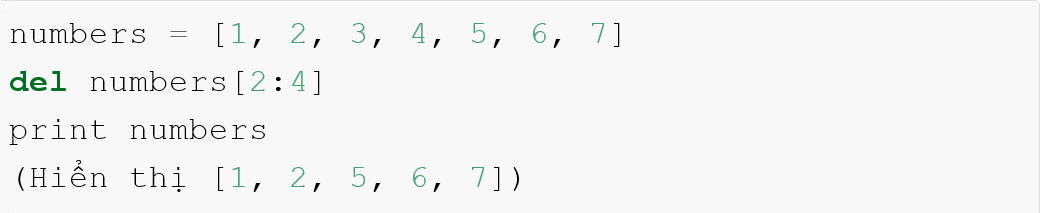
### 1.3.7.3 Xóa phầntửcủa mảng

Cóthể xóa một phần tử thông qua toán tửdel. Thứtựcủa các phần tửsẽdịch chuyển tùy vào vị trí của phần tửbị xóa.

Ví dụ:



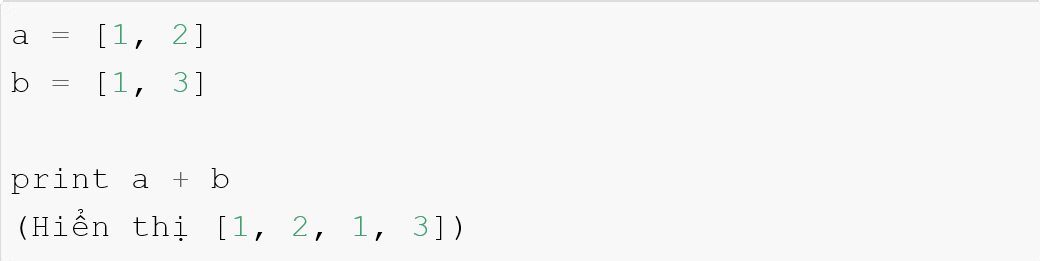
Bạn có thể xóa một khoản dựa vào toán tửlấy khoản [start:end].

Ví dụ:

### 1.3.7.4 Nối 2 mảng

Bạn có thểsửdụng toán tử+ đểnối giátrịcủa 2 mảngvàtạo ra một mảng lớn có sốlượng phần tửlà tổng sốlượng phần tửcủa 2 mảng con.

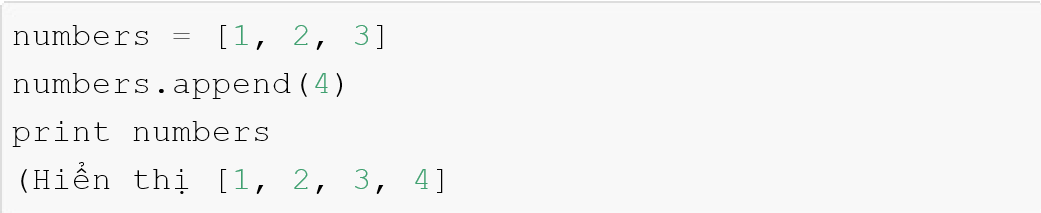
Ví dụ:



### 1.3.7.5 Thêm phần tử vào mảng

Nếu bạn muốn thêm phần tử vào một mảng đã tồn tại, hãy dùng phương thức list.append(newvalue)để thêm phần tử có giá trị newvalue vào cuối mảng list.

Vídụ:



### 1.3.7.6Lấy phầntửcuối mảng

Nếu muốn lấy phần tử cuối cùng của mảng ra khỏi mảng, có thểsử dụng phương thức list.pop(), sẽtrảvề giá trị của phần tửcuối cùng và mảng bây giờsẽ không còn phần tửnày.



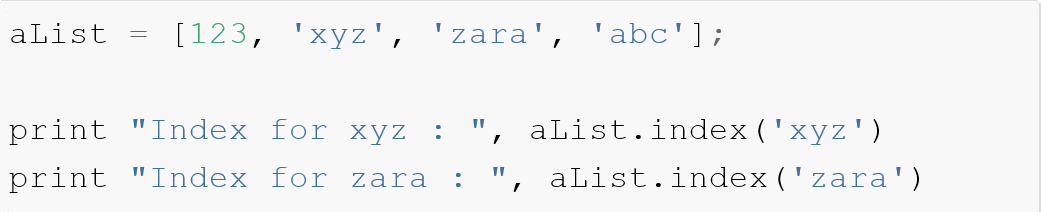
### 1.3.7.7Tìm một giá trị trong mảng

Nếu bạn muốn tìm vị trí (index) của một giá trị trong mộtmảng, có thể dùng phương thức list.index(obj).

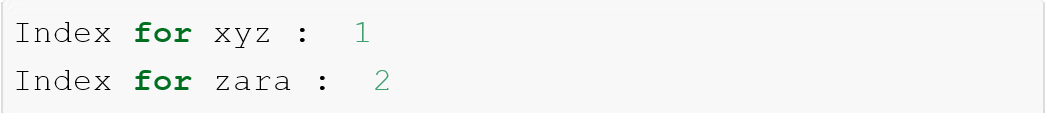
Nếu tìm thấy sẽtrảvềindexcủa phần tửđầu tiên tìm thấy.

Nếu không tìm thấy sẽ quăng Exception.

Ví dụ:



Khi chạy sẽ hiển thị kết quả:

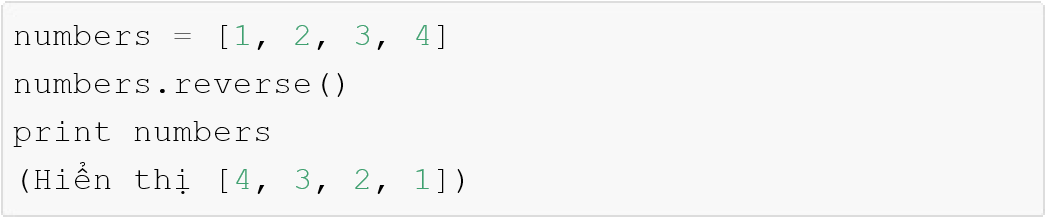


### 1.3.7.8Đảo ngược giá trịcủa mảng

Đểđảo ngược thứtự các giá trịcủamột mảng, sửdụng phương thức list.reverse().

Phương thức này không trảvềkết quả mà thay đổi trực tiếp mảng list.

Ví dụ:



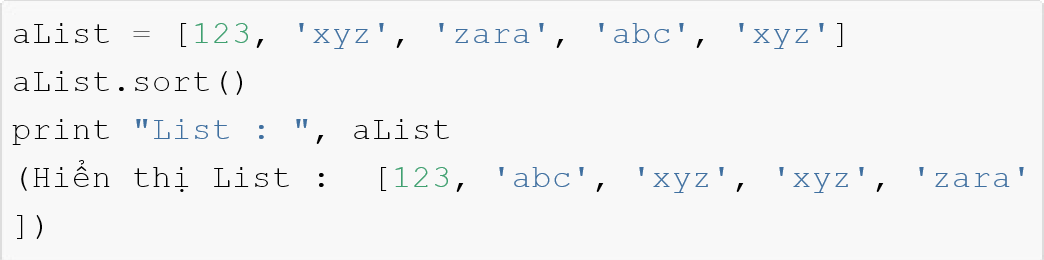
### 1.3.7.9Sắp xếp giá trị các phần tử

Đểsắp xếp thứtựcủa giá trị trong mảng, sửdụng phương thức list.sort([func]) đểsắp xếp.

Nếu tham sốđầu vào là hàm func không truyền vào thì mặc định là sắp xếp theo giá trị tăng dần.

Phương thức này không trảvềkết quả mà thay đổi trực tiếp mảng list.

Ví dụ:



Cách triển khai hàm zompare func() cũng giống như hàm usort trong PHP.

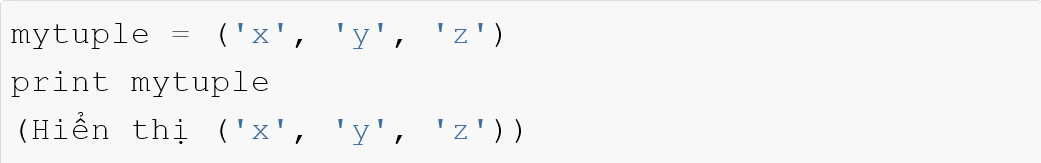
Hàm trảvề các giá trị 0, -1 và 1.

## 1.3.8 Tuple

Tuple cũng là một cấu trúc mảng, tương tựnhư cấu trúc List.

Một sốđiểm khác nhau cơ bản là khai báo Tuple sử dụng cặp dấu ngoặc (...) và một tuple đã được khai báo rồi thì không thay đổi được giá trị (immutable) và không hỗ trợ các phương thức như append(), pop()…

Ví dụ:



Vẫn hỗtrợ các cách để truy xuất phần tử giống List như là truy xuấttheoindex, range,tìm kiếm…

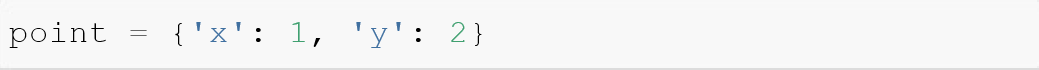
## 1.3.9Dictionary

Dictionary cũng là một cấu trúc mảng, nhưng các phần tử bao gồm key và value.

Nếu bạn có biếtJSON thì cấu trúc Dictionary tương tự như một object json.

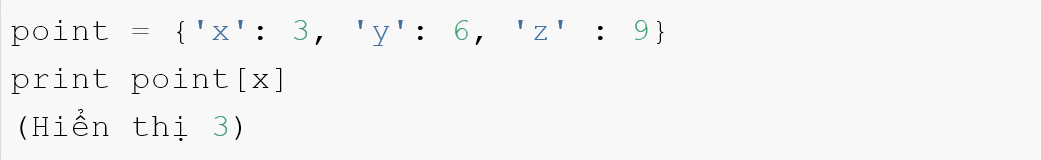
Một Dictionary được khai báo bằng cặp dấu ngoặc {...}.

Ví dụ:



Truy xuất một giá trịdựa vào key của đốitượng.

Ví dụ:

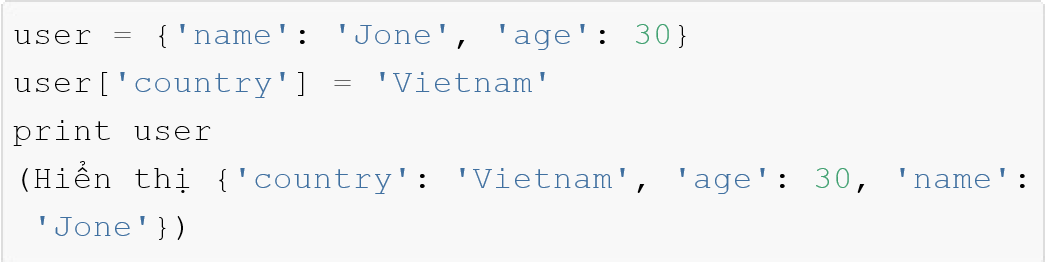


### 1.3.9.1 Thêm một phầntử

Đểthêm một phần tử vào đối tượng đã khai báo, sửdụng cấu trúc:

dict[key] = value.

Ví dụ:



### 1.3.9.2 Một số hàm, phương thức thông dụng:

* dict.clear(): Xóa toàn bộdữliệubêntrongđối tượng
* dict.copy(): Trảvềmộtbảncopycủađối tượng
* dict.fromkeys(seq,[value]): Tạomộtđối tượngvới danhsách keytừseqvànếu có truyền value thìlấy đólàm giá trị cho các phần tử.
* dict.has\_key(key): kiểm tramộtkeycó tồntại trongđối tượnghay không.
* dict.keys(): Trảvềmột List chứa các key
* dict.values(): Trảvềmột List chứa các value

# CHƯƠNG 2: GIỚI THIỆU VỀ DJANGO FRAMEWORK

# 2.1 Giới thiệu về Django

* Django là một web framework khá nổi tiếng được viết hoàn toàn bằng ngôn ngữ Python. Nó không phải là một micro-framework như Flask, mà là mộtframework với đầy đủ các thư viện, module hỗ trợ các web-developer.
* Django là một khung Web Python cấp cao, khuyến khích phát triển nhanh chóng và thiết kế thực dụng, gọn gàng.
* Được xây dựng bởi các nhà phát triển có kinh nghiệm, nó xử lý nhiều rắc rối của việc phát triển Web, vì vậy bạn có thể tập trung vào viết ứng dụng của mình mà không cần phải phát minh lại bánh xe. Nó có nguồn mở và miễn phí.

# 2.2 Khả năng

* Django có đầy đủ các yếu tố phục vụ cho một Startup:
* Nhanh: Django được thiết kế để giúp các nhà phát triển đưa các ứng dụng từ ý tưởng đến hoàn thành càng nhanh càng tốt.
* Có đầy đủ các thư viện/module cần thiết: Django bao gồm hàng tá các tính năng bổ sung mà bạn có thể sử dụng để xử lý các tác vụ phát triển Web phổ biến. Django chăm sóc xác thực người dùng, quản trị nội dung, bản đồ trang web, nguồn cấp dữ liệu RSS và nhiều tác vụ khác ngay lập tức.
* Đảm bảo về tính bảo mật: Django rất coi trọng vấn đề bảo mật và giúp các nhà phát triển tránh được nhiều lỗi bảo mật phổ biến, chẳng hạn như SQL, kịch bản chéo trang, giả mạo yêu cầu chéo trang và nhấp chuột. Hệ thống xác thực người dùng của nó cung cấp một cách an toàn để quản lý tài khoản và mật khẩu người dùng.
* Khả năng mở rộng tốt: Một số địa điểm bận rộn nhất trên hành tinh sử dụng khả năng có thể mở rộng nhanh chóng và linh hoạt của django để đáp ứng nhu cầu giao thông nặng nhất.
* Tính linh hoạt: Các công ty, tổ chức và chính phủ đã sử dụng Django để xây dựng tất cả mọi thứ - từ hệ thống quản lý nội dung đến mạng xã hội đến nền tảng điện toán khoa học.
* Các công ty hàng đầu sử dụng Django Framework
* **Instagram**  
  [Instagram](https://www.instagram.com/) là một mạng xã hội chia sẻ ảnh và video bằng python. Mỗi giây, Instagram xử lý lượng dữ liệu khổng lồ và quản lý số lượng tương tác thậm chí còn lớn hơn giữa nhiều người dùng.

Django đã giúp Instagram xử lý tất cả các công việc này bằng cách giữ cho nó đơn giản đúng như tinh thần của Python.

* **Spotify**  
  [Spotify](https://www.spotify.com/) đã đưa ngành công nghiệp âm nhạc lên một tầm cao mới bằng cách thay đổi cách mọi người nghe nhạc và làm cho mọi người có thể truy cập trên bất kỳ thiết bị nào.

Spotify sử dụng Python cho cả Back end Service và Machine Learning với Django Framework.

* **Youtube**  
  [YouTube](https://www.youtube.com/) là một trong những nền tảng chia sẻ video phổ biến nhất. Ban đầu, nó là một dự án xây dựng trên PHP. Nhưng để cải thiện hiệu suất, YouTube chuyển sang Django.
* **Dropbox**[Dropbox](https://www.dropbox.com/) là một trong những dịch vụ lưu trữ đám mây nổi tiếng cho các tài liệu, video và hình ảnh.Nó được xây dựng dựa trên python cho cả phần mềm Desktop và Client Server. Dropbox sử dụng Django Framework để cho phép lưu trữ, đồng bộ hóa và cung cấp các tùy chọn để chia sẻ các loại tệp khác nhau.
* **Trình duyệt Mozilla**

Mặc dù các thành phần cũ của Mozilla không được viết bằng python nhưng các thành phần mới được triển khai bằng Django.

Ví dụ: Các trang web hỗ trợ và add-ons được xây dựng bằng cách sử dụng Django. Họ đã chuyển từ PHP (CakePHP) sang Python (Django Framework) giúp họ xử lý hàng chục, hàng trăm và hàng triệu lượt dùng mỗi tháng và rất nhiều lưu lượng truy cập API mỗi ngày.

* **Disqus**

[Disqus](https://disqus.com/) là dự án lớn nhất được thực hiện bằng Django. Các nhà phát triển Disqus đã xây dựng ứng dụng từ đầu bằng cách sử dụng Django để mở rộng quy mô để xử lý hàng triệu người dùng mỗi ngày.

Họ cũng đã sử dụng Django trong một trong những dự án của họ có tênlà [Sentry](https://sentry.io/welcome/), đây là một tool report lỗi cũng nổi tiếng với các lập trình viên hiện nay.

* **Hạn chế của Django**

Chỉ định URL với các biểu thức chính quy không phải là một nhiệm vụ dễ dàng để thực hiện cho người mới bắt đầu học Python. Các lỗi Template ẩn có thể khiến bạn lãng phí rất nhiều thời gian.

Django mang lại một cảm giác nguyên khối vì nó là một framework mạnh.  
Nó không có khả năng quản lý các yêu cầu khác nhau cùng một lúc vì nó không trao quyền cho các thủ tục riêng lẻ để xử lý nhiều yêu cầu cùng một lúc. Các lập trình viên thường sẽ cân nhắc nếu họ cần xem xét các phương thức khác nhau để thực hiện một quy trình kiểm soát các yêu cầu khác nhau cùng một lúc.

Hệ thống ORM (Object Relation Mapping) cho phép các lập trình viên làm việc trên các cơ sở dữ liệu khác nhau cùng một lúc nhưng nó thiếu một số tính năng chủ yếu được cung cấp bởi các hệ thống ORM khác. Nó được thiết kế theo cách mà không cho phép các lập trình viên sử dụng SQLAlchemy hoàn toàn.

# 2.3Cách sử dụng

## 2.3.1 Tạo project

|  |  |
| --- | --- |
| 1 | C:\Project>django-admin startproject mysite |

Để tạo một project Django: mở Command Prompt (cmd) lên và chuyển đến thư mục mà bạn muốn tạo, sau đó gõ đoạn lệnh phía trên, ở trên chúng ta tạo một project với tên là mysite. Lệnh startproject sẽ tạo một thư mục có tên là mysite, cấu trúc bên trong thư mục:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | mysite/  manage.py  mysite/  \_\_init\_\_.py  settings.py  urls.py  wsgi.py |

## 2.3.2 Chạy server

Thư mục mysite  có chứa một file tên là manage.py, file này cung cấp các chức năng vận hành server:

|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py runserver |

Để chạy server, chạy file manage.py và đưa vào tham số runserver.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | Performing system checks...   System check identified no issues (0silenced).    You have unapplied migrations; your app may notwork properly until they are applied.  Run 'python manage.py migrate'to applythem.    March 21, 2016-15:50:53  Django version 1.9, using settings 'mysite.settings'  Starting development server at <a class="reference external"href="<http://127.0.0.1:8000/>">http://127.0.0.1:8000/</a>  Quit the server with CTRL-BREAK. |

Để đổi port mặc định của server thì bạn thêm số port mà bạn muốn sau tham số runserver.

|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py runserver 8080 |

## 2.3.3 Tạo web app

|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py startapp polls |

Mở một trình Command Prompt khác lên và trỏ đến thư mục chứa file manage.py, sau đó chạy file này với tham số startapp để tạo một ứng dụng web, ở đây chúng ta tạo một ứng dụng web với tên là polls, nếu như bạn chưa biết thì đây là một ứng dụng đặt câu hỏi và cho phép người dùng trả lời trắc nghiệm, thường dùng để làm các chương trình khảo sát người dùng. Một thư mục với tên polls sẽ được tạo ra và có cấu trúc như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | polls/      \_\_init\_\_.py      admin.py      apps.py      migrations/      \_\_init\_\_.py      models.py      tests.py      views.py |
|  |  |

Thư mục này chứa các file chuẩn của một ứng dụng web Django.

## 2.3.4 Tạo View

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | fromdjango.http importHttpResponse    defindex(request):      response =HttpResponse()      response.write("<h1>Welcome</h1>")      response.write("This is the polls app")      returnresponse |

|  |  |
| --- | --- |
| 5  6  7  8  9  10 | Bên trong file views.py chúng ta viết hàm index, hàm này trả về một đối tượng HttpResponse. Để gọi được tới hàm index này và lấy nội dung HTML thì chúng ta phải tạo đường dẫn URL tới file này.      \_\_init\_\_.py      admin.py      apps.py      migrations/          \_\_init\_\_.py      models.py      tests.py      urls.py      views.py |

Để tạo URL chúng ta tạo một file với tên urls.py bên trong thư mục polls.

|  |  |
| --- | --- |
| polls/urls.py | |
| 1  2  3  4  5  6  7 | fromdjango.conf.urls importurl    from. importviews     urlpatterns =[       url(r'^$', views.index, name='index'),  ] |

Bên trong file urls.py khai báo một đối tượng List có tên urlpatterns, Django sẽ tìm thông tin về các url trong list này, list này chứa các đối tượng url, mỗi url bao gồm 3 tham số, tham số thứ nhất là biểu thức chính quy (Regex – Regular Expression), tham số thứ hai là tên module và tên hàm sẽ trả về nội dung HTML, ở đây là hàm index trong module views (file views.py), tham số thứ 3 là tên biến toàn cục trong toàn bộ ứng dụng, trong đó 2 tham số đầu tiên là bắt buộc phải có.

Regex (biểu thức chính quy) là một chủ đề khá lớn, Regex cho phép chúng ta tạo ra các mẫu định dạng text dùng trong tìm kiếm, xác thực… ở đây Regex giúp tạo các đường dẫn tới các hàm trả về nội dung HTML một cách tự động. Để tạo các chuỗi Regex thì có một số quy luật sau đây:

* ^ – bắt đầu regex
* $ – kết thúc regex
* \d – một kí tự số
* + –  kí tự phía trước có thể lặp lại một hoặc nhiều lần
* / – có một dấu /
* () – gom nhóm một số kí tự nhất định lại với nhau

Sau khi đã có đường dẫn cho riêng polls, thì tiếp theo chúng ta phải tạo đường dẫn đến ứng dụng polls, tạo đường dẫn cho từng ứng dụng trong file mysite/urls.py.

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7 | fromdjango.conf.urls importinclude, url  fromdjango.contrib importadmin    urlpatterns =[      url(r'^polls/', include('polls.urls')),      url(r'^admin/', admin.site.urls),  ] |

Bên trong file mysite/urls.py cũng có một list urlpatterns, list này chứa các đối tượng url dẫn đến các ứng dụng web khác mà chúng ta viết.

## 2.3.5 Django - Model

Trong thư mục mysite chứa một file tên là settings.py, file này chứa các thông tin cấu hình server.

Mặc định thì Django server sử dụng cơ sở dữ liệu SQLite, nhưng nếu bạn muốn sử dụng CSDL khác thì trong file settings.py, bạn tìm đến đối tượng DATABASES và thay đổi các giá trị sau:

* ENGINE: tên module dành cho từng CSDL, mặc định sử dụng SQLite
  + django.db.backends.sqlite3 – cơ sở dữ liệu  SQLite
  + django.db.backends.postgresql – cơ sở dữ liệu PostgreSQL
  + django.db.backends.mysql – cơ sở dữ liệu MySQL
  + django.db.backends.oracle – cơ sở dữ liệu Oracle
* NAME: tên CSDL, mặc định là file db.sqlite3 được tạo ra ở thư mục gốc của server. Nếu bạn không dùng CSDL SQLite thì bạn phải tạo CSDL với tên trùng với NAME trong CSDL mà bạn dùng (bằng câu lệnh CREATE DATABASE <name>)

Ngoài ra nếu không dùng SQLite thì bạn cũng phải cung cấp thêm các thông tin USER, PASSWORD, HOST nữa và user phải có quyền truy cập CSDL cũng như một số quyền như ghi, xem…

Trong file settings.py còn có một list có tên là INSTALLED\_APPS, mặc định khi tạo một project, Django cung cấp cho chúng ta một số ứng dụng thường dùng trong list này.

Mỗi ứng dụng trong list này sẽ giúp Django tìm các model để tạo bảng tương ứng trong file CSDL, nhưng khi tạo project thì các bảng này không tự động được tạo trong file db.sqlite3, để tạo các bảng này chúng ta chạy lệnh sau:

|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py migrate |

Lệnh migrate sẽ tìm các module được liệt kê trong list INSTALLED\_APPS (trong file mysite/settings.py) và tạo các bảng CSDL tương ứng.

2.3.6 Tạo mô hình dữ liệu cho ứng dụng

Mặc định khi tạo một ứng dụng web trong một project Django thì Django đã tạo sẵn cho chúng ta một file có tên là models.py để khai báo các mô hình

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7  8  9  10 | fromdjango.db importmodels    classQuestion(models.Model):      question\_text =models.CharField(max\_length=200)      pub\_date =models.DateTimeField('date published')    classChoice(models.Model):      question =models.ForeignKey(Question, on\_delete=models.CASCADE)      choice\_text =models.CharField(max\_length=200)      votes =models.IntegerField(default=0) |

Chúng ta sẽ tạo 2 mô hình (2 lớp…) là Question và Choice.

|  |  |
| --- | --- |
| 1  2  3 | classQuestion(models.Model):  ...  classChoice(models.Model): |

Mỗi mô hình được tạo ra phải được kế thừa từ lớp django.db.models.Model.

|  |  |
| --- | --- |
| 1  2  3  4 | question\_text =models.CharField(max\_length=200)  pub\_date =models.DateTimeField('date published')  ...  votes =models.IntegerField(default=0) |

Mỗi thuộc tính trong mô hình được kế thừa từ một lớp Field, đây là một lớp ảo trong Django, lớp này lại có nhiều lớp kế thừa khác đại diện cho mỗi kiểu dữ liệu ví dụ như CharField là kiểu text, DateTimeField là kiểu DateTime…

Tham số max\_length là số lượng ký tự tối đa, đây là tham số bắt buộc phải có. Các tham số còn lại như default… là tham số tùy chọn, không có cũng được.

|  |  |
| --- | --- |
| 1 | question =models.ForeignKey(Question, on\_delete=models.CASCADE) |

Ngoài các kiểu dữ liệu thường dùng thì chúng ta còn có kiểu khóa ngoại được định nghĩa trong lớp ForeignKey, tham số đầu tiên là bảng mà khóa ngoại này tham chiếu tới, on\_delete=models.CASCADE tức là khi dữ liệu trong bảng cha có sự thay đổi thì dữ liệu trong bảng con cũng sẽ thay đổi theo, chẳng hạn như bản ghi trong bảng Question bị xóa thì các bản ghi trong bảng Choice có tham chiếu tới bản ghi trong bảng Question này cũng sẽ bị xóa.

## 2.3.7 Tạo bảng trong CSDL từ Model

Sau khi đã tạo các lớp mô hình trong Python, Django sẽ nhìn các thuộc tính trong từng lớp để tạo các bảng tương ứng trong CSDL và tạo thêm các lớp mới trong Python cung cấp các hàm để thao tác với các bảng trong CSDL.

|  |  |
| --- | --- |
| mysite/settings.py | |
| 1  2  3  4  5  6  7  8  9 | INSTALLED\_APPS =[      'polls.apps.PollsConfig',      'django.contrib.admin',      'django.contrib.auth',      'django.contrib.contenttypes',      'django.contrib.sessions',      'django.contrib.messages',      'django.contrib.staticfiles',  ] |

Để Django tạo các bảng mới trong CSDL thì chúng ta phải khai báo trước trong list INSTALLED\_APPS đã bằng cách thêm dòng polls.apps.PollsConfig.

|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py makemigrations polls |

Tiếp theo, chạy lệnh makemigration polls để báo cho Django biết là bạn đã thay đổi một số mô hình, ở đây là thêm 2 lớp mới trong gói polls.

|  |  |
| --- | --- |
| 1  2  3  4  5 | Migrations for'polls':    0001\_initial.py:      -Create model Choice      -Create model Question      -Add field question to choice |

Django sẽ tạo một file python để lưu các thông tin về sự thay đổi này, ở đây là file 0001\_initial.py.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | C:\Project\mysite>python manage.py migrate  Operations to perform:  Applyallmigrations: admin, contenttypes, polls, auth, sessions  Running migrations:  Rendering model states... DONE  Applying polls.0001\_initial... OK |

Chạy lại lệnh migrate để Django cập nhật lại CSDL. Lúc này CSDL chúng ta sẽ có thêm 2 bảng mới là polls\_choice và polls\_question (tên bảng được đặt theo cú pháp <tên package>\_<tên lớp>).

## 2.3.8 Thao tác với các bảng

Sau khi Django đã tạo các bảng bạn có thể bắt đầu thực hiện các công việc thường dùng như thêm-sửa-xóa… bản ghi.

|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py shell |

Có thể sử dụng trình shell mà Django cung cấp sẵn trong file manage.py. Nếu bạn không muốn chạy trực tiếp trong Command Prompt mà muốn chạy trong file .py riêng thì trong file .py đấy bạn phải khai báo các dòng dưới đây:

|  |  |
| --- | --- |
| 1  2  3  4 | importos  importdjango  os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "mysite.settings");  django.setup() |

Các dòng này sẽ thiết lập biến môitrườngDJANGO\_SETTINGS\_MODULE

trỏ đến module mysite.settings, mục đích là để chỉ cho Python biết đường dẫn import. Mặc định khi chạy file manage.py từ cmd thì biến này sẽ được Django thiết lập luôn.

|  |  |
| --- | --- |
| 1  2  3 | >>> frompolls.models importQuestion, Choice  >>> Question.objects.all()  [] |

Import 2 lớp Question và Choice. Phương thức Question.objects.all() liệt kê toàn bộ đối tượng Question đang có trong CSDL, hiện tại là không có bản ghi nào nên kết quả trả về list rỗng.

|  |  |
| --- | --- |
| 1  2 | >>> fromdjango.utils importtimezone  >>> q =Question(question\_text="What's new?", pub\_date=timezone.now()) |

Tạo một đối tượng Question và thiết lập pub\_date là ngày giờ hiện tại của máy tính, chúng ta lấy thông tin này từ phương thức timezone.now() trong module django.utils.timezone.

|  |  |
| --- | --- |
| 1 | >>> q.save() |

Để lưu đối tượng Question mới này vào CSDL thì chỉ cần gọi phương thức save() là được.

|  |  |
| --- | --- |
| 1  2 | >>> q.id  1 |

Khi viết lớp Question và Choice chúng ta không cung cấp thuộc tính id hay bất cứ thuộc tính nào để làm khóa chính, vì vậy khi tạo CSDL Django sẽ tự động thêm 1 thuộc tính id để làm khóa chính và thuộc tính này sẽ tự động tăng.

|  |  |
| --- | --- |
| 1  2  3  4 | >>> q.question\_text  What's new?  >>> q.question\_text ="What's up?"  >>> q.save() |

Bạn có thể thay đổi trực tiếp giá trị trên các đối tượng này sau đó gọi phương thức save() là dữ liệu trên CSDL sẽ được cập nhật.

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7  8  9  10  11 | fromdjango.db importmodels  classQuestion(models.Model):      question\_text =models.CharField(max\_length=200)      pub\_date =models.DateTimeField('date published')      def\_\_str\_\_(self):          returnself.question\_text  classChoice(models.Model):      question =models.ForeignKey(Question, on\_delete=models.CASCADE)      choice\_text =models.CharField(max\_length=200)      def\_\_str\_\_(self):          returnself.choice\_text |

Sau đó chúng ta tắt và mở lại shell, rồi chạy thử đoạn lệnh sau sẽ được kết quả khác:

|  |  |
| --- | --- |
| 1  2  3 | >>> frompolls.models importQuestion, Choice  >>> Question.objects.all()  [<Question: What's up?>] |

Phương thức objects.all() sẽ gọi đến phương thức \_\_str\_\_().

|  |  |
| --- | --- |
| 1  2  3  4 | >>> Question.objects.filter(id=1)  [<Question: What's up?>]  >>> Question.objects.filter(question\_text\_\_startswith='What')  [<Question: What's up?>] |

Bạn có thể lọc các bản ghi theo thuộc tính bằng phương thức filter().

|  |  |
| --- | --- |
| 1 | >>> q =Question.objects.get(pk=1) |

Bạn cũng có thể lấy từng bản ghi đơn lẻ bằng phương thức get(), ở trên chúng ta lấy theo khóa chính pk (Primary Key).

|  |  |
| --- | --- |
| 1  2 | >>> q.choice\_set.all()  [] |

Bất cứ bảng nào có một bảng khác chứa khóa ngoại tham chiếu đến khóa chính của nó đều sẽ có một thuộc tính được tạo tự động là một tập hợp các đối tượng của bảng kia. Ở đây bảng Choice chứa khóa ngoại tham chiếu đến bảng Question, do đó bảng Question sẽ có một danh sách các đối tượng Choice, chúng ta có thể lấy danh sách này qua choice\_set.all() (tên danh sách được đặt theo <tên bảng>\_set). Ở đây chúng ta vẫn chưa tạo đối tượng Choice nào nên danh sách trả về rỗng.

|  |  |
| --- | --- |
| 1  2  3  4  5 | >>> q.choice\_set.create(choice\_text='Not much', votes=0)  <Choice: Not much>  >>> q.choice\_set.create(choice\_text='The sky', votes=0)  <Choice: The sky>  >>> c =q.choice\_set.create(choice\_text='Just hacking again', votes=0) |

Tạo 3 đối tượng Choice bằng phương thức choice\_set.create().

|  |  |
| --- | --- |
| 1  2  3  4 | >>> q.choice\_set.all()  [<Choice: Not much>, <Choice: The sky>, <Choice: Just hacking again>]  >>> q.choice\_set.count()  3 |

Lấy số lượng các bản ghi trong bảng bằng phương thức count().

|  |  |
| --- | --- |
| 1  2 | >>> c =q.choice\_set.filter(choice\_text\_\_startswith='Just hacking')  >>> c.delete() |

Để xóa một bản ghi trong bảng thì chúng ta dùng phương thức delete().

## 2.3.9 Hệ thống Admin

### 2.3.9.1 Tạo user

Để có thể đăng nhập vào admin thì trước hết chúng ta phải tạo một tài khoản admin đã vì Django không tạo sẵn cho chúng ta khi tạo project.

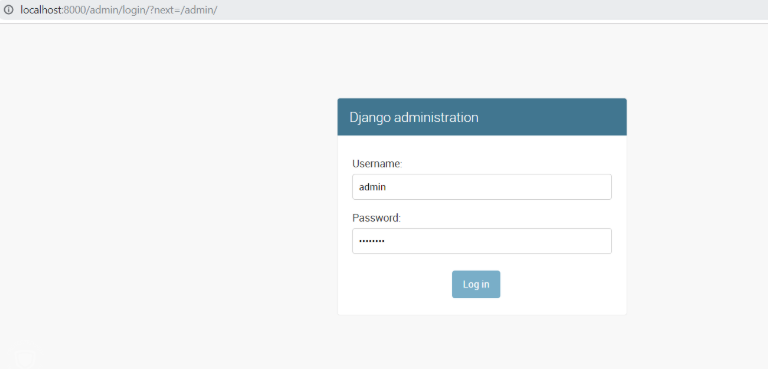
|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py createsuperuser |

Để tạo tài khoản thì chúng ta chạy file manage.py với tham số createsuperuser.

|  |  |
| --- | --- |
| 1  2  3  4  5 | Username: admin  Email address: admin@example.com  Password:  Password (again):  Superuser created successfully. |

Tiếp theo bạn cung cấp username, password và email.

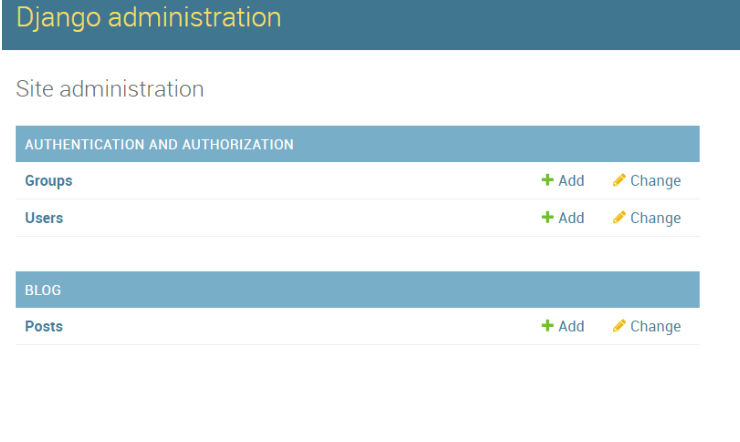
### 2.3.9.2 Truy cập trang admin



Hình 2.1 Trang đăng nhập admin

Để truy cập vào trang admin thì bạn chỉ cần thêm /admin vào đường dẫn trang chủ.

Sau khi đăng nhập thành công, **Admin**mặc định sẽ quản lý 2 bảng là **User**(bảng lưu các user trong hệ thống), và bảng **Group**nhằm tạo các nhóm trong user với những quyền có thể thực thi trong hệ thống.



Hình 2.2: Giao diện bảng lưu các User và Group

Sau đó bạn đăng nhập bằng username và password mà chúng ta đã tạo hồi nãy là sẽ được chuyển đến giao diện admin.

Mặc định Django đã bật chức năng dịch nên có thể trang admin sẽ được hiển thị bằng ngôn ngữ mà bạn dùng trên trình duyệt.

Tại trang admin chúng ta có thể thao tác với 2 bảng là User và Group. Các bảng Question và Choice mà chúng ta đã tạo ra không được hiển thị ở đây là vì chúng ta chưa đăng ký các bảng đó với trang admin.

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7 | fromdjango.contrib importadmin    # Register your models here.  from.models importQuestion, Choice    admin.site.register(Question)  admin.site.register(Choice) |

Để đăng ký các bảng (hay các mô hình) với admin thì chúng ta chỉ cần dùng phương thức admin.site.register() trong file admin.py mà Django đã tạo cho chúng ta.

## 2.3.10 Django – View và Template

Trong Django thì một View là một hàm/phương thức làm một công việc cụ thể nào đó, một View thường đi kèm với một Template. Ví dụ, một ứng dụng Blog có một số View sau:

* Home – hiển thị các bài viết mới nhất.
* Entry – hiển thị bài viết nào đó.
* Archive – Lưu trữ các bài viết theo năm/tháng.
* Comment – xử lý việc đăng bình luận của độc giả.

Trong ứng dụng “Poll” mà chúng ta đã viết từ đầu series tới bây giờ, chúng ta sẽ xây dựng các View sau:

* Index – Hiển thị các câu hỏi mới.
* Detail – Hiển thị một câu hỏi nhất định nào đó và đưa các câu trả lời để người dùng chọn.
* Result – Hiển thị kết quả bầu chọn của người dùng.
* Vote – Xử lý việc trả lời của người dùng.

Trong Django, một trang web được tạo ra bởi các hàm View, Django sẽ chọn View nào tùy thuộc vào URL mà chúng ta đã thiết lập.

Để từ một đường dẫn URL đến một View thì Django sử dụng khái niệm URLConf, đây là một module Python của Django làm nhiệm vụ phân tích đường dẫn và chuyển đến một hàm View nhất định.

### 2.3.10.1 Tạo View

Chúng ta sẽ viết thêm một số hàm View mới.

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7  8  9 | def detail(request, question\_id):      return HttpResponse("You're looking at question %s." % question\_id)    def results(request, question\_id):      response = "You're looking at the results of question %s."      return HttpResponse(response % question\_id)    def vote(request, question\_id):      return HttpResponse("You're voting on question %s." % question\_id) |

Chúng ta viết thêm 3 view mới là detail(), results() và vote().

Tiếp theo chúng ta tạo thêm các URL trỏ đến từng view này.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | from django.conf.urls import url    from . import views    urlpatterns = [      # ex: /polls/      url(r'^$', views.index, name='index'),      # ex: /polls/5/      url(r'^(?P<question\_id>[0-9]+)/$', views.detail, name='detail'),      # ex: /polls/5/results/      url(r'^(?P<question\_id>[0-9]+)/results/$', views.results, name='results'),      # ex: /polls/5/vote/      url(r'^(?P<question\_id>[0-9]+)/vote/$', views.vote, name='vote'),  ] |

### 2.3.10.2 Template

Các hàm View trả vềmột trong hai thứ: hoặc là trả về một đốitượng HttpReponse chứa nội dung HTML để hiển thị lên trình duyệt, hoặc là một lỗi exception 404.

Django cung cấp cho chúng ta hệ thống Template, chúng ta có thể viết phần giao diện ở Template và dùng nó để hiển thị dữ liệu của các View cho đẹp mắt hơn, vậy mục đích chính của View chính là kết nối giữa Template và Model.

Đầu tiên chúng ta tạo một thư mục có tên là templates trong thư mục polls, Django sẽ tự động tìm các file template trong thư mục này. Ở đây bạn tạo thêm một thư mục khác nữa tên là polls, trong thư mục này bạn tạo một file tên index.html.

|  |
| --- |
|  |

Trình duyệt chỉ hiểu code HTML chứ không hiểu code Python, để có thể sử dụng code Python thì Django cung cấp cho chúng ta các thẻ template, thẻ template bắt đầu và kết thúc bằng cặp kí tự {% %} hoặc {{ }}, các câu lệnh Python nằm trong cặp dấu {% %}, còn các biến thì nằm trong cặp {{ }}.

|  |  |
| --- | --- |
| 1 | from django.template import loader |

Để có thể dùng template thì trước hết chúng ta import module loader trong gói django.template.

|  |  |
| --- | --- |
| 1 | template = loader.get\_template('polls/index.html') |

Để dùng template thì chúng ta dùng phương thức get\_template

|  |  |
| --- | --- |
| 1 | return HttpResponse(template.render(context, request)) |

Sau đó chúng ta gọi đến phương thức template.render() để tạo nội dung HTML có sử dụng template. Tham số đầu tiên là nội dung HTML trả về, tham số thứ 2 là đối tượng request được gửi đến.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | from django.shortcuts import render    def index(request):      latest\_question\_list = Question.objects.order\_by('-pub\_date')[:5]      context = {          'latest\_question\_list': latest\_question\_list,      }      return render(request, 'polls/index.html', context) |

Ngoài ra bạn cũng có thể dùng hàm render() để trả về một đốitượng HttpResponse một cách trực tiếp luôn, với tham số thứ nhất là đối tượng request, tham số thứ 2 là đường dẫn đến file template, tham số thứ 3 là nội dung HTML trả về.

## 2.3.11 Django - Upload file

Tạo một app mới với tên là file\_uploader:

|  |  |
| --- | --- |
| 1 | C:\Project\mysite>python manage.py startapp file\_uploader |

Khai báo app cho project.

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5 | INSTALLED\_APPS =[      #...      'file\_uploader',      #...  ] |

Tạo file forms.py bên trong thư mục file\_uploader:

|  |  |
| --- | --- |
| file\_uploader/forms.py | |
| 1  2  3  4  5 | fromdjango importforms    classUploadFileForm(forms.Form):      title =forms.CharField(max\_length=50)      file=forms.FileField() |

Để hiển thị thẻ element chọn file thì chúng ta dùng lớp FileField().

Tiếp theo chúng ta tạo thư mục templates và file template:

|  |  |
| --- | --- |
| file\_uploader/templates/fileUploaderTemplate.html | |
| 1  2  3  4  5  6  7 | <formaction=""method="POST"enctype="multipart/form-data">      {% csrf\_token %}      <table>          {{ form.as\_table }}      </table>      <inputtype="submit"value="Submit"/>  </form> |

Để gửi file lên thì trong thẻ <form> chúng ta phải khai báo thuộctính enctype="multipart/form-data".

Sau khi đã có template và form để hiển thị thì chúng ta tiến hành tạo view để kết nối chúng với nhau.

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | fromdjango.db importmodels    # Create your models here.  fromdjango.http importHttpResponse  from.forms importUploadFileForm    deffileUploaderView(request):      ifrequest.method =='POST':          form =UploadFileForm(request.POST, request.FILES)          ifform.is\_valid():              upload(request.FILES['file'])              returnHttpResponse("<h2>File uploaded successful!</h2>")          else:              returnHttpResponse("<h2>File uploaded not successful!</h2>")        form =UploadFileForm()      returnrender(request, 'fileUploaderTemplate.html', {'form':form})    defupload(f):      file=open(f.name, 'wb+')      forchunk inf.chunks():          file.write(chunk) |

Viết hàm view fileUploaderView() và hàm xử lý việc ghi file là upload().

|  |  |
| --- | --- |
| 1 | ifform.is\_valid(): |

Phương thức is\_valid() có trong lớp django.forms.Form sẽ kiểm tra tính hợp lệ của dữ liệu được gửi lên. Mục đích chính của lớp Form trong Django chính là hỗ trợ chúng ta kiểm tra sự đúng đắn của dữ liệu được gửi lên.

Phương thức is\_valid() chỉ có chức năng đơn giản là kiểm tra xem ô text title có dữ liệu hay không vì mặc định các field của Form trong Django bắt buộc phải có dữ liệu mới được nhận.

|  |  |
| --- | --- |
| 1  2 | upload(request.FILES['file'])  returnHttpResponse("<h2>File uploaded successful!</h2>") |

Nếu dữ liệu là hợp lệ thì chúng ta tiến hành lưu file vào thư mục gốc của server và trả lời thông báo gửi thành công về cho người dùng.

|  |  |
| --- | --- |
| 1  2  3  4 | defupload(f):      file=open(f.name, 'wb+')      forchunk inf.chunks():          file.write(chunk) |

Hàm upload() sẽ thực hiện copy file vào thư mục gốc của server (thư mục có file manage.py)

Tạo url trỏ tới hàm view trong app.

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7 | fromdjango.conf.urls importurl    from. importviews    urlpatterns =[      url(r'^$', views.fileUploaderView),  ] |

Tiếp theo là url từ địa chỉ server tới app:

|  |  |
| --- | --- |
|  | |
| 1  2  3  4  5  6  7 | fromdjango.conf.urls importurl, include  fromdjango.contrib importadmin    urlpatterns =[      url(r'^admin/', admin.site.urls),      url(r'^file-upload/', include('file\_uploader.urls')) |

# CHƯƠNG 3: XÂY DỰNG CƠ SỞ DỮ LIỆU VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

## 3.1 Xây dựng cơ sở dữ liệu

### **3.1.1 Bảng User: Lưu thông tin sinh viên**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | NOTNULL | ID sinh viên | Khóa chính |
| 2 | Password | VARCHAR(218) | NULL | Mật khẩu |  |
| 3 | Last\_login | DATETIME(6) | NULL | Thời gian đăng nhập cuối cùng |  |
| 4 | Is\_superuser | TINYINT(1) | NULL | Cấp quyền admin |  |
| 5 | Username | VARCHAR(150) | NOTNULL | Tên tài khoản |  |
| 6 | FirstName | VARCHAR(30) | NOTNULL | Tên sinh viên |  |
| 7 | LastName | VARCHAR(30) | NOTNULL | Họ sinh viên |  |

Bảng 3.1:Bảng User

### **3.1.2 Bảng Category: Lưu thông tin chuyên ngành**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | NOTNULL | ID sinh viên | Khóa chính |
| 2 | Tên chuyên ngàng | VARCHAR(250) | NOTNULL |  |  |

Bảng 3. 2:Bảng Categogy

## 3.1.3Bảng Question: Lưu thông tin câu hỏi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | Not null | ID câu hỏi | Khóa chính |
| 2 | Figure | VARCHAR(100) | Null | Ảnh minh họa |  |
| 3 | Content | VARCHAR(100) | Null | Câu trả lời |  |
| 4 | Explaination | Text | Null | Giải thích |  |
| 5 | Category\_id | INT(11) | Null | ID Chuyên ngành | Khóa ngoài |

Bảng 3. 3: Bảng Question

## 3.1.4 Bảng Quiz: Lưu thông tin môn kiểm tra

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | Not null | ID bộ câu hỏi | Khóa chính |
| 2 | Title | VARCHAR(60) | Null | Tên môn kiểm tra |  |
| 3 | Description | Longtext | Null | Mô tả |  |
| 4 | url | VARCHAR(60) | NOTNULL |  |  |
| 5 | Random\_order | TINYINT(1) | Null | Lựa chọn có đảo câu hỏi hay không |  |
| 6 | Max\_question | INT(10) | NOTNULL | Số câu hỏi tối đa cho 1 đề |  |
| 7 | Passmark | INT(6) | NULL | Số điểm để hoàn thành bài thi |  |
| 8 | Category\_id | Int(11) | Not null | Mã chuyên ngành | Khóa ngoài |

Bảng 3. 4: Bảng Quiz

## 3.1.5 Bảng Question\_quiz: Lưu thông tin các câu hỏi có trong 1 môn kiểm tra

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | Not null | ID bộ câu hỏi | Khóa chính |
| 2 | Question\_id | Int(11) | Null | Mã câu hỏi | Khóa ngoài |
| 3 | Quiz\_id | Int(11) | Null | Mã đề kiểm tra | Khóa ngoại |

Bảng 3. 5: Bảng Question\_quiz

## 3.1.6 Bảng Answer: Lưu thông tin câu trả lời cho các câu hỏi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | Not null | ID đề | Khóa chính |
| 2 | Content | Varchar(50) | NotNull | Câu trả lời |  |
| 3 | Correct | TINYINT(1) | NotNull | Đáp án |  |
| 4 | Question\_id | Int(11) | Not null | Mã câu hỏi | Khóa ngoài |

Bảng 3. 6:Bảng Answer

## 3.1.7 Bảng Progress: Lưu thông tin tiến trình thực hiện bài làm của sinh viên

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | Not null | ID đề | Khóa chính |
| 2 | Score | Varchar(50) | Null | Điểm số |  |
| 3 | Correct\_answer | Varchar(50) | Null | Số câu đúng |  |
| 4 | Wrong\_answer | Varchar(50) | Null | Số câu sai |  |
| 5 | User\_id | Int(11) | NotNull | Mã tài khoản sinh viên | Khóa ngoại |

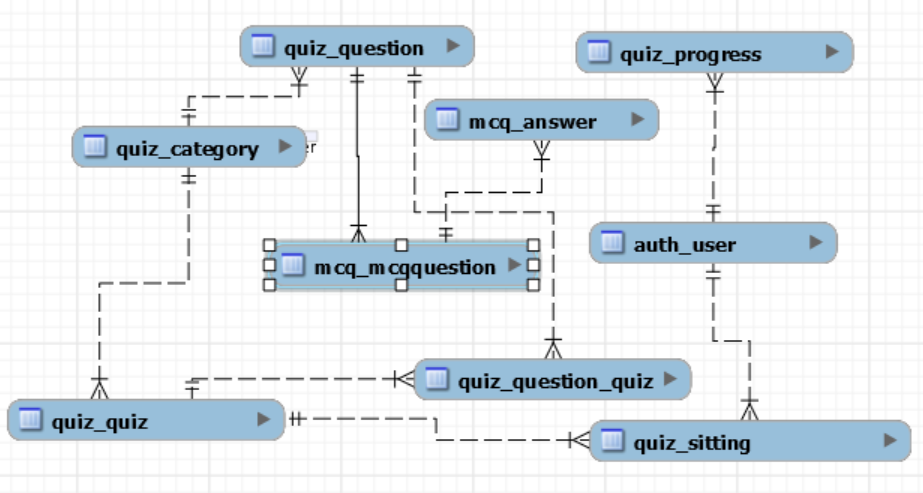
Bảng 3. 7: Bảng Progress

## 3.1.8Bảng Sitting: Lưu lại thông tin của bài kiểm tra khi sinh viên làm bài

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả | Ghi chú |
| 1 | ID | Int(11) | Not null | ID bài làm | Khóa chính |
| 2 | Question\_list | VARCHAR(1024) | Null | Lưu lại danh sách mã các câu hỏi có trong bài kiểm tra | Khóa ngoại |
| 3 | Incorrect\_question | VARCHAR(1024) | Null |  |  |
| 4 | Current\_score | Int(11) | Null | Số điểm của bài thi |  |
| 5 | Complete | TINYINT(1) | Null | Bài kiểm tra hoàn thành hay chưa |  |
| 6 | User\_answer | LongText | Null | Câu trả lời |  |
| 7 | Quiz\_id | Int(11) | NotNull | Mã môn kiểm tra | Khóa ngoài |
| 8 | User\_id | Int(11) | NotNull | Id sinh viên | Khóa ngoài |

Bảng 3. 8: Bảng Sitting

## 3.1.9: Biểu đồ mô tả cơ sở dữ liệu



Hình 3. 1: Biểu đồ mô tả cơ sở dữ liệu

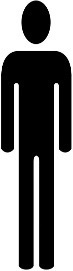
## 3.1.10 Kết nối CSDL

Dự án chúng em thực hiện sử dụng SQLite, là Hệ quản trị CSDL mặc định trong các dự án Django. Để thêm CSDL mới cho dự án, chúng ta vào phần Settings.py của dự án:

DATABASES = {  
'default': {  
}  
'ENGINE': 'django.db.backends.sqlite3',  
'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),  
}

# 3.2 Phân tích và thiết kế hệ thống thông tin

## 3.2.1 Biểu đồ Usecase Quản trị tổng quan



Hình 3. 2: Biểu đồ Usecae Quản trị tổng quan

## 3.2.2 Biểu đồ Usecase Sinh viên tổng quan



Hình 3. 4: Biểu đồ Usecase Sinh viên tổng quan

## 3.2.3 Usecase miêu tả đăng nhập admin



Hình 3. 5: Usecase miêu tả đăng nhập admin

Mô tả:

Tác nhân: người quản trị

Cho phép người quản trị đăng nhập để thao tác với hệ thống

Dòng sự kiện:

Các dòng sự kiện chính:

* Chức năng này bắt đầu khi người quản trị muốn đăng nhập để sử dụng hệ thống
* Hệ thống yêu cầu người quản trị nhập tên đăng nhập, mật khẩu.
* Hệ thống kiểm tra tính hợp lệ của tên đăng nhập, mật khẩu.

Các dòng sự kiện phụ:

* Nếu trong dòng sự kiện chính người quản trị cung cấp 1 tên đăng nhập, mật khẩu hoặc nhóm người quản trị không hợp lệ hệ thống sẽ xuất hiện 1 thông báo lỗi trên màn hình. Người quản trị có thể chọn tiếp tục đăng nhập lại theo dòng sự kiện chính hoặc hủy bỏ việc đăng nhập để kết thúc chức năng đăng nhập. Nếu đăng nhập thành công, người quản trị sẽ bắt đầu sử dụng những chức năng tiếp theo với vai trò đã đăng nhập. Nếu không trạng thái của hệ thống không thay đổi.

## 3.1.4 Usecase thay đổi mật khẩu admin



Hình 3. 6: Usecase thay đổi mật khẩu admin

Mô tả:

Tác nhân: người quản trị

Dòng sự kiện:

Các dòng sự kiện chính:

* Các chức năng này bắt đầu khi người quản trị đã đăng nhập được vào hệ thống.
* Hệ thống yêu cầu người quản trị nhập tên, mật khẩu mới.
* Hệ thống kiểm tra tính hợp lệ của mật khẩu mới qua đó cho phép người quản trị có thể sử dụng hệ thống với mật khẩu mới.

Các dòng sự kiện phụ:

* Nếu dòng sự kiện chính người quản trị cung cấp mật khẩu mới không hợp lệ hệ thống sẽ xuất hiện một thông báo lỗi trên màn hình.
* Nếu thay đổi mật khẩu thành công, hệ thống sẽ cho phép người quản trị sử dụng mật khẩu đó cho đến khi nào người quản trị muốn thay đổi mật khẩu khác.

## 3.2.5 Usecase mô tả Quản lí đề thi



Hình 3. 7: Usecase mô tả Quản lí đề thi

Mô tả: cho phép người quản trị sau khi đăng nhập hệ thống có thể quản lí (nhập, sửa, xóa) đề thi.

Dòng sự kiện:

Các dòng sự kiện chính:

* Chức năng này bắt đầu khi người quản trị lựa chọn việc quản lí đề thi
* Người quản trị thực hiện các thao tác thêm mới hay cập nhật và có lưu hoặc hủy thay đổi.
* Hệ thống kiểm tra tính hợp lệ của dữ liệu và thực hiện lưu thay đổi

Các dòng sự kiện phụ:

* Nếu trong dòng sự kiện chính người quản trị cung cấp những dữ liệu không hợp lệ (thêm mới, cập nhật) hệ thống sẽ xuất hiện thông báo lỗi trên màn hình.
* Người quản trị có thể lựa chọn tiếp tục thực hiện các thao tác theo dòng sự kiện chính hoặc hủy bỏ chức năng này.
* Các thay đổi về đề thi được cập nhật.

## 3.2.6 Usecase mô tả đăng nhập của sinh viên



Hình 3. 8: Usecase mô tả đăng nhập của sinh viên

Mô tả:

Tác nhân: sinh viên

Dòng sự kiện:

Các dòng sự kiện chính:

* Chức năng này bắt đầu khi thành viên muốn đăng nhập vào sử dụng hệ thống
* Hệ thống yêu cầu sinh viên nhập tên đăng nhập, mật khẩu
* Hệ thống kiểm tra tính hợp lệ của tên đăng nhập, mật khẩu đăng nhập

Các dòng sự kiện phụ:

* Nếu trong dòng sự kiện chính sinh viên cung cấp 1 tên đăng nhập, mật khẩu hoặc nhóm sinh viên không hợp lệ thì hệ thống sẽ hiện thông báo lỗi trên màn hình.
* Sinh viên có thể chọn tiếp tục đăng nhập lại theo dòng sự kiện chính hoặc hủy bỏ việc đăng nhập để kết thúc chức năng đăng nhập.
* Nếu đăng nhập thành công sinh viên sẽ bắt đầu sử dụng các chức năng tiếp theo với vai trò đã đăng nhập. Nếu không, trạng thái của hệ thống không đổi

## 3.2.7 Usecase thực hiện bài kiểm tra



Hình 3. 9: Usecase thực hiện bài kiểm tra

Mô tả: cho phép sinh viên làm bài kiểm tra

Dòng sự kiện:

Các dòng sự kiện chính:

* Hệ thống hiển thị danh sách câu hỏi trong nội dung đề.
* Nếu sinh viên không thực hiện kiểm tra, điểm của bài kiểm tra sẽ là 0 điểm.
* Sinh viên làm đúng đến đâu được cộng điểm đến đó.

## 3.2.8 Sơ đồ phân rã chức năng

### 3.2.8.1 Sơ đồ phân rã chức năng tổng quát

Sinh viên

Admin

Xây dựng phần mềm kiểm tra trắc nghiệm

Hình 3. 10: Sơ đồ phân rã chức năng tổng quát

### 3.2.8.2 Sơ đồ phân rã chức năng Admin

Lịch sử làm bài

Sinh viên làm bài

Cập nhật

Xóa

Thêm

Xóa

Cập nhật

Thêm

Xóa

Cập nhật

Thêm

Thống kê

QL thời gian đóng/mở bài thi

QL đề kiểm tra

QL tài khoản GV, SV

Admin

Hình 3. 11: Sơ đồ phân rã chức năng Admin

## 3.2.9 Sơ đồ luồng dữ liệu

### 3.2.9.1 Sơ đồ luồng dữ liệu Admin

Cập nhật đề

Đề

Cập nhật câu hỏi

Kết quả

Sinh viên

Chấm điểm

Thông tin tk

Làm bài

Admin

Hình 3. 12: Sơ đồ luồng dữ liệu Admin

### 3.2.9.2 Sơ đồ luồng dữ liệu Sinh viên

Yêu cầu làm bài

Thông tin tk

Nộp bài

Sinh viên

Yêu cầu kq

Điểm thi

Hình 3. 13: Sơ đồ luồng dữ liệu Sinh viên

## 3.2.10 Mô hình quan hệ thực thể

(n,n)

(n,n)

Lớp

Giảng viên

Bài thi

Bài làm

Môn

(1,n)

(1,n)

(1,n)

(n,n)

(n,n)

(1,n)

Sinh viên

(n,n)

(n,n)

Hình 3. 14: Mô hình quan hệ thực thể

# CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG

# 4.1 Tổng quan giao diện

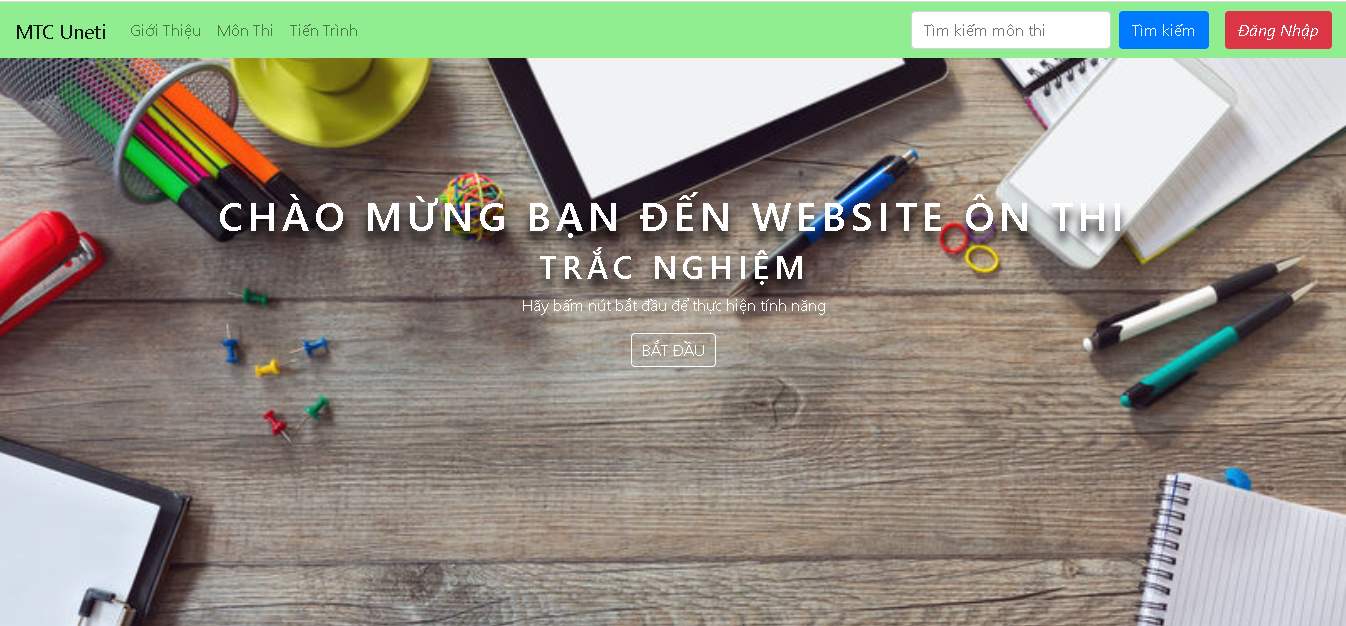
Giao diện được thiết kế phù hợp với nhu cầu người dùng. Các chức năng được hiển thị ở menu giúp người dùng có thể thao tác nhanh chóng giữa các trang. Nội dung đề thi sát với đề thi thực tế góp phần cho người dùng làm quen với đề thi trước khi bước vào kì thi chính thức. Trang web có chức năng tự động tính điểm thi người dùng hoàn thành bài thi của mình và điểm thi sẽ được lưu vào CSDL khi người dùng là thành viên của Website.

Website có các trang chính:

* Đăng nhập, đăng kí thông tin cá nhân
* Danh sách các môn thi
* Kiểm tra
* Kết quả thi

# 4.2 Giao diện

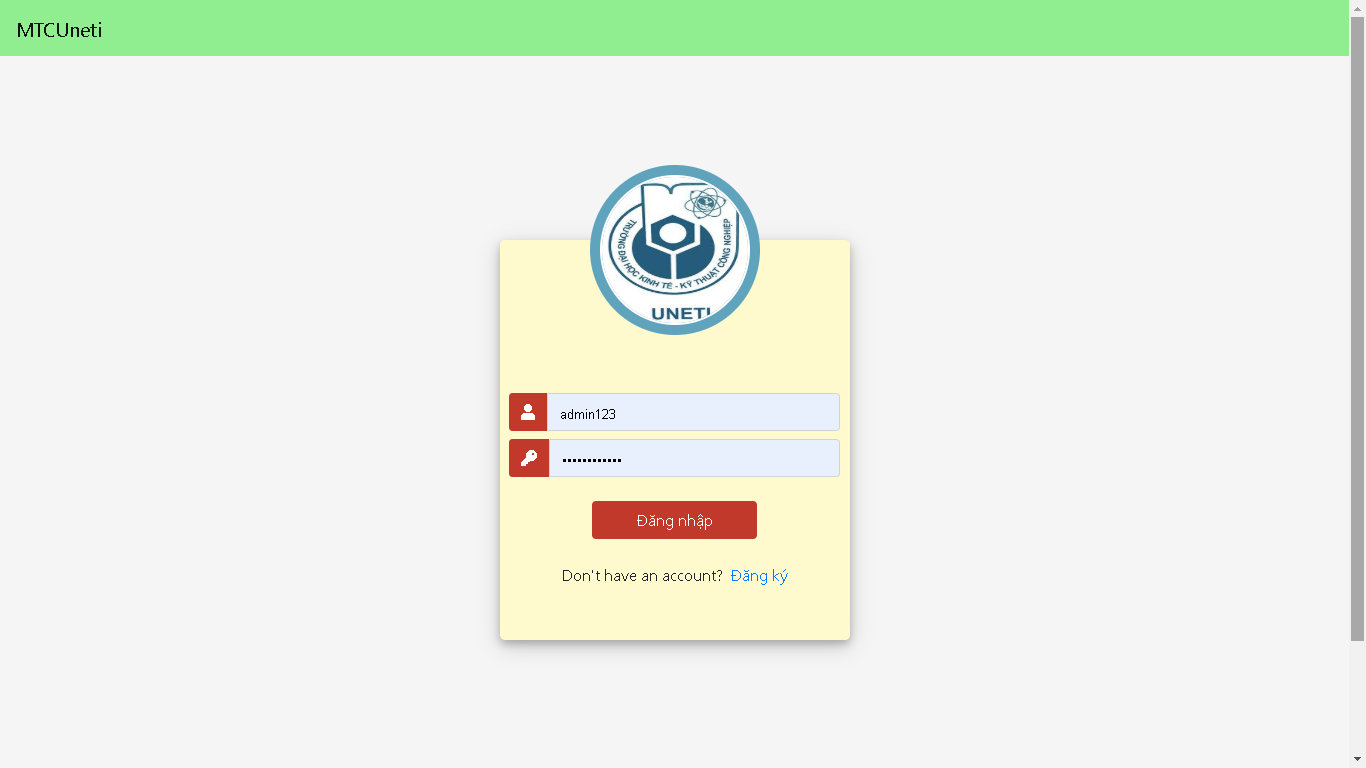
## 4.2.1 Giao diện trang chủ Website



Hình 4. 1: Giao diện trang chủ Website

## 4.2.2 Giao diện trang đăng nhập

Nếu người dùng đã có tài khoản đăng kí, có thể đăng nhập trực tiếp với tên tài khoản và mật khẩu đã đăng kí.

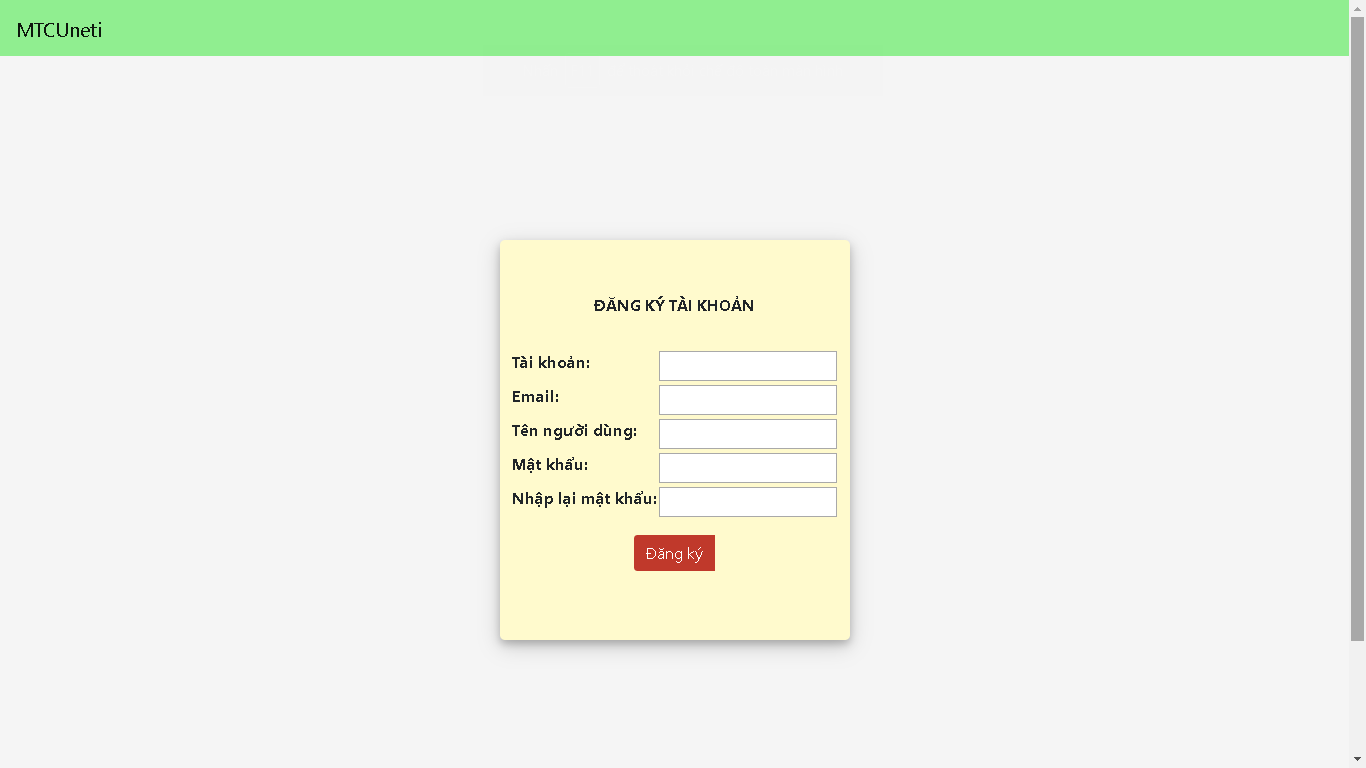


Hình 4. 2: Giao diện trang đăng nhập

## 4.2.3 Giao diện trang đăng kí

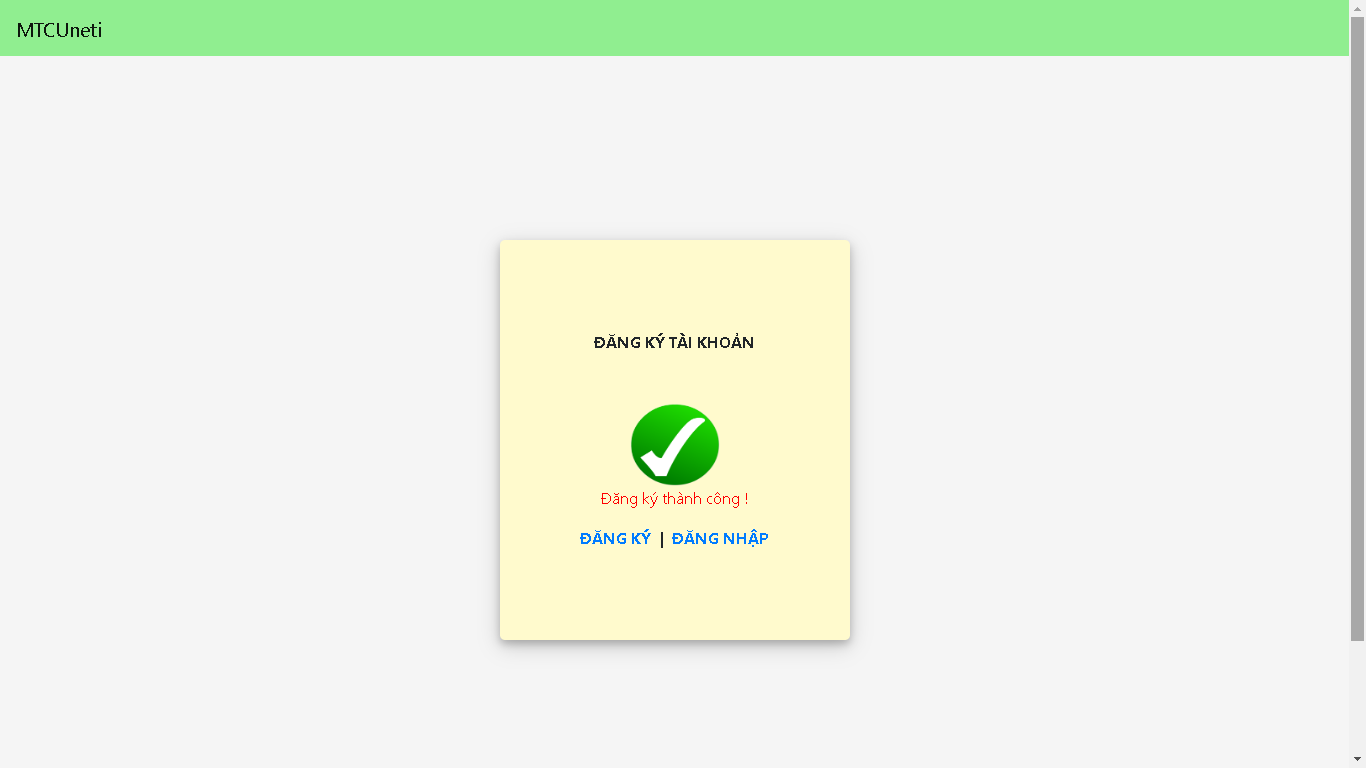
Nếu người dùng chưa có tài khoản, có thể đăng kí tài khoản. Người dùng nhập thông tin cá nhân của mình vào các ô tương ứng.

Sau khi nhập xong, click vào ô đăng kí, người dùng đã tạo 1 tài khoản và được lưu thông tin vào CSDL. Lần đăng nhập tiếp theo, người dùng chỉ cần đăng nhập với tên tài khoản và mật khẩu đúng như đã đăng kí trước đó.



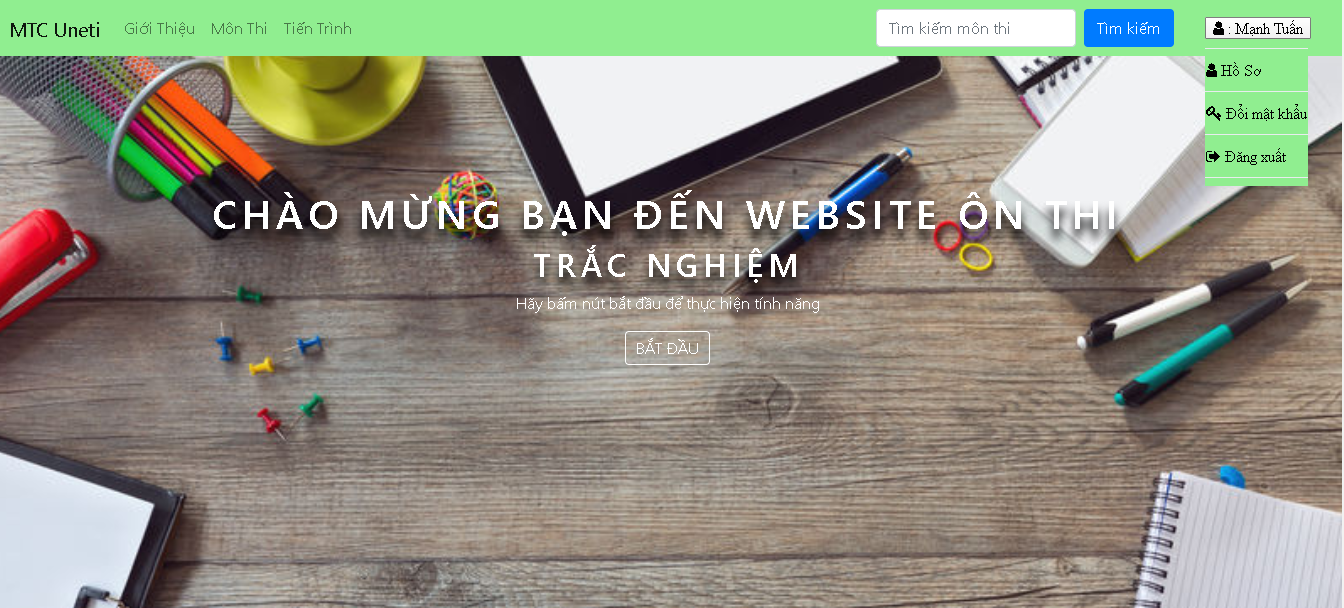
Hình 4. 3: Giao diện trang đăng kí

Sau khi đăng ký thành công



Hình 4. 4: Giao diện đăng kí thành công

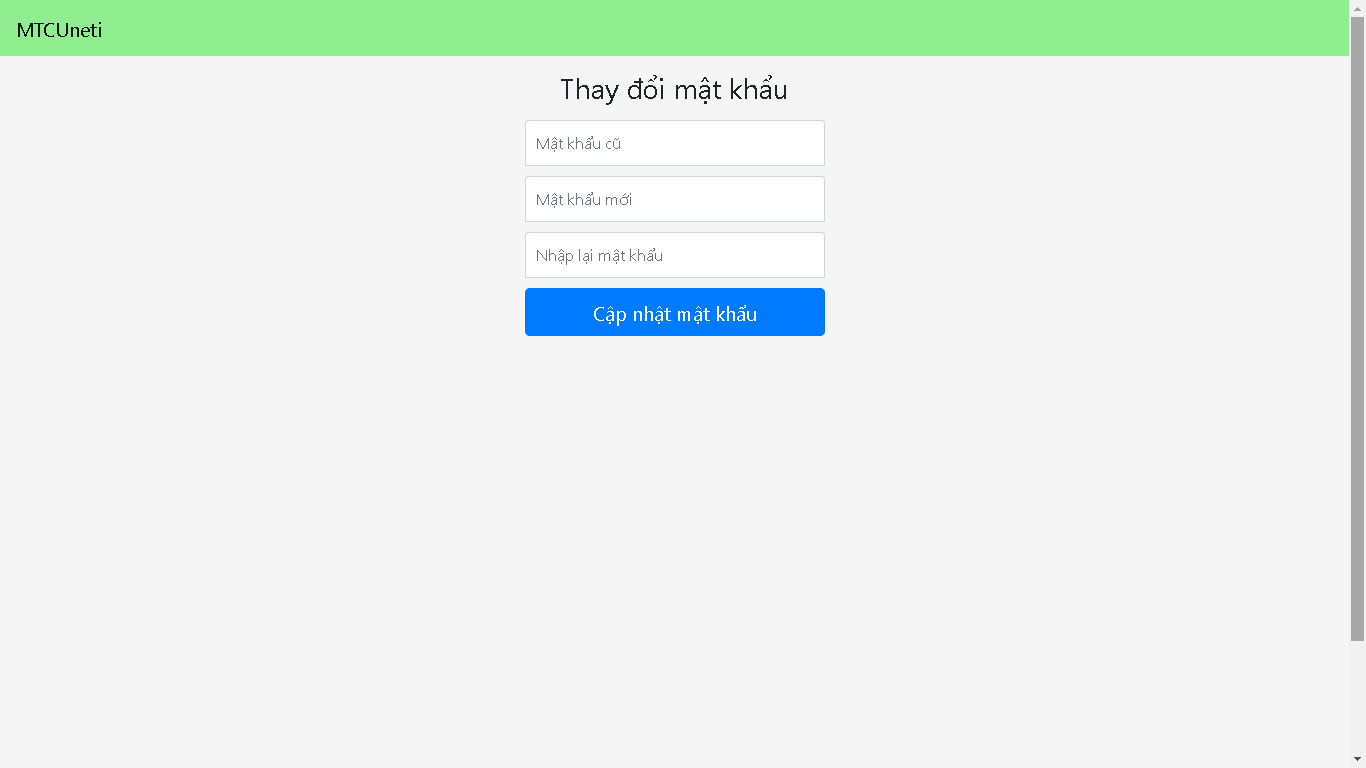
Tài khoản được đăng nhập



Hình 4. 5: Giao diện tài khoản được đăng nhập

## 4.2.4 Giao diện trang thay đổi mật khẩu

Trong quá trình sử dụng, người dùng có thể quên mật khẩu đã đăng kí, hoặc muốn thay đổi mật khẩu để đảm bảo tính bảo mật. Chức năng thay đổi mật khẩu giúp người dùng thực hiện điều đó. Người dùng chỉ việc nhập vào các ô tương ứng và click vào “Cập nhật mật khẩu”.

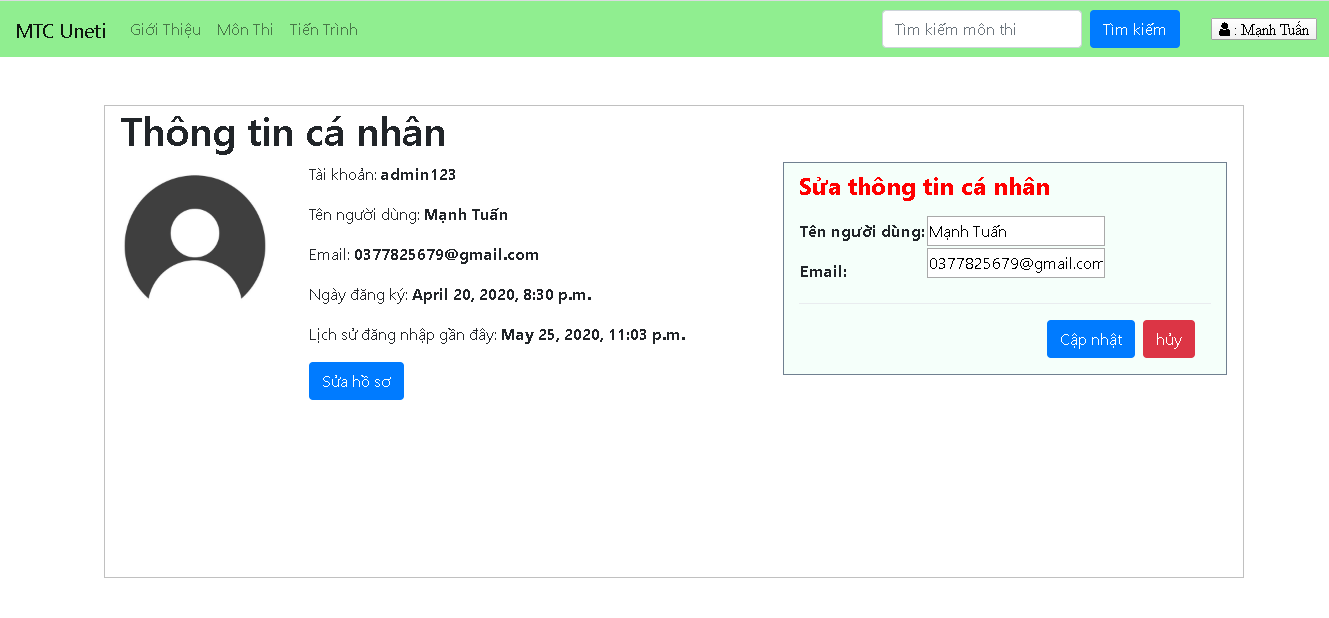


Hình 4. 6: Giao diện trang thay đổi mật khẩu

## 4.2.5 Giao diện trang hồ sơ cá nhân

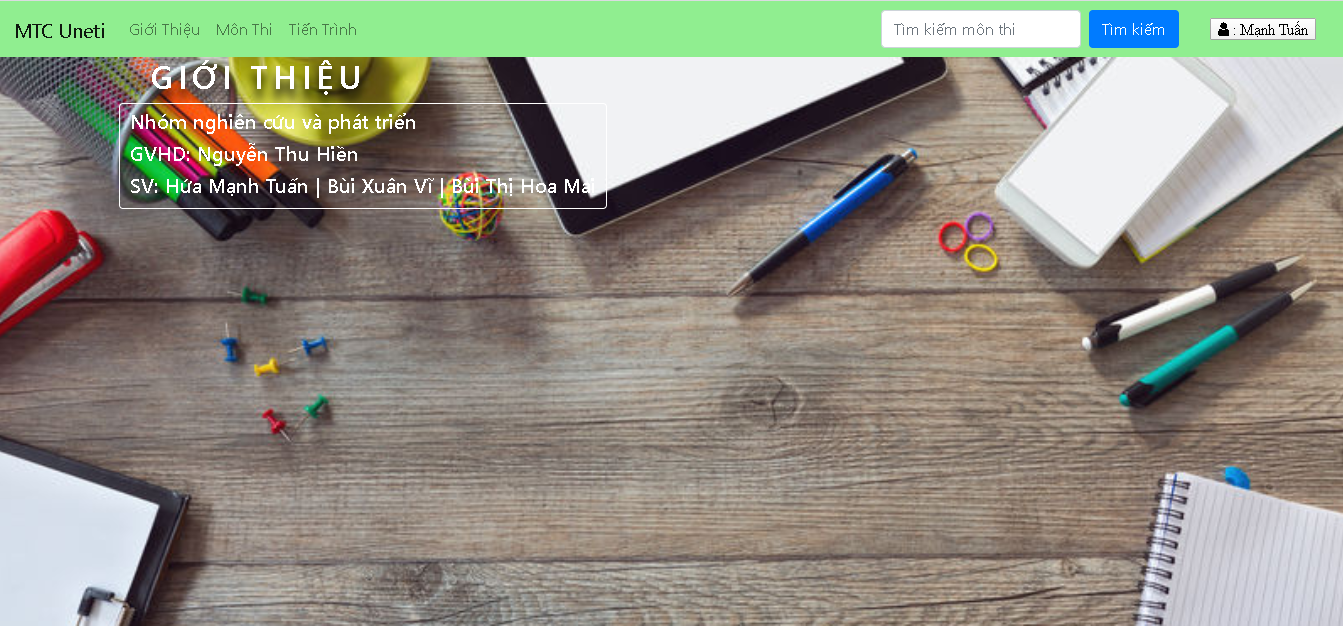
Trang chứa các thông tin cá nhân của người dùng bao gồm: tài khoản, tên người dùng, email, ngày đăng kí, lịch sử đăng nhập gần đây.

Người dùng cũng có thể chỉnh sửa thông tin của mình nếu muốn.



Hình 4. 7: Giao diện trang hồ sơ cá nhân

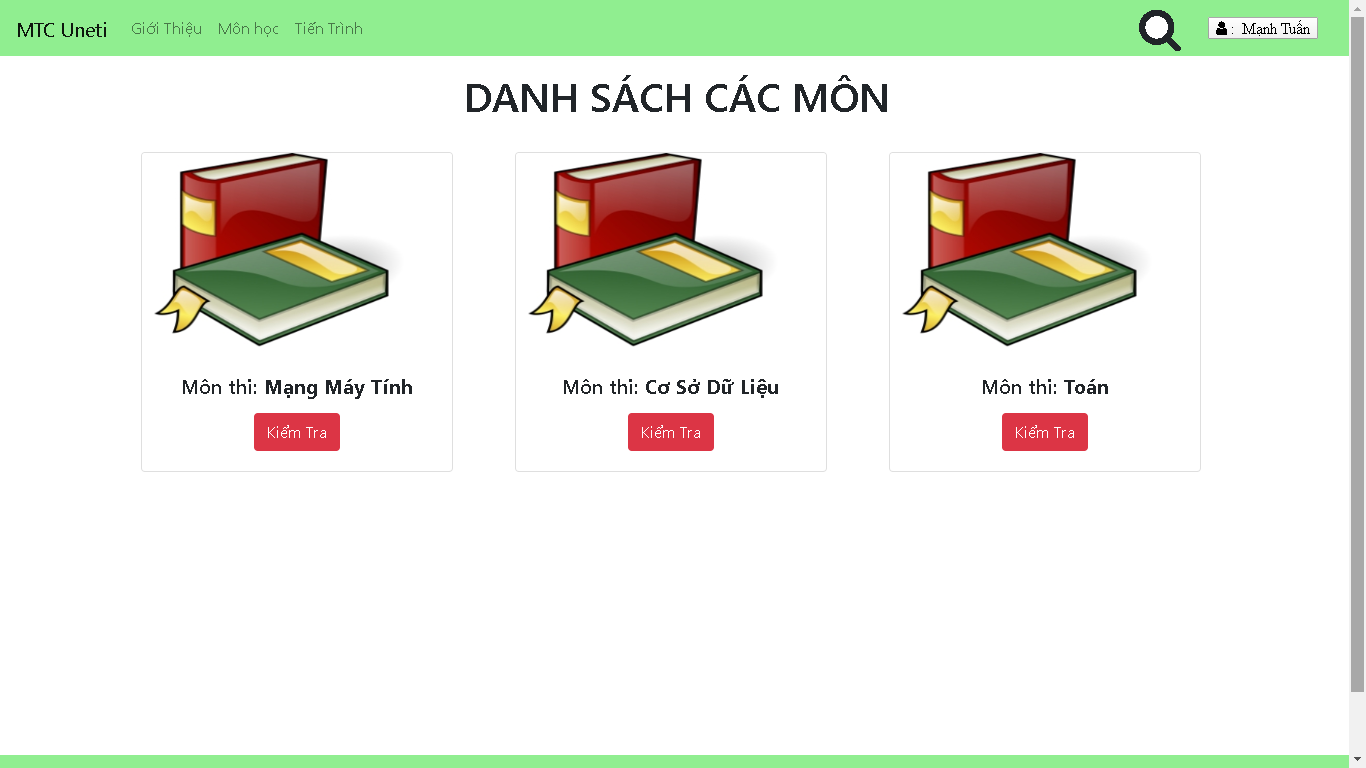
## 4.2.6 Giao diện trang giới thiệu



Hình 4. 8: Giao diện trang giới thiệu

## 4.2.7 Giao diện trang môn thi

Người dùng có thể lựa chọn môn thi và click “Kiểm tra” để bắt đầu tiến hành thi

****

Hình 4. 9: Giao diện trang môn thi

## 4.2.8 Giao diện trang kiểm tra

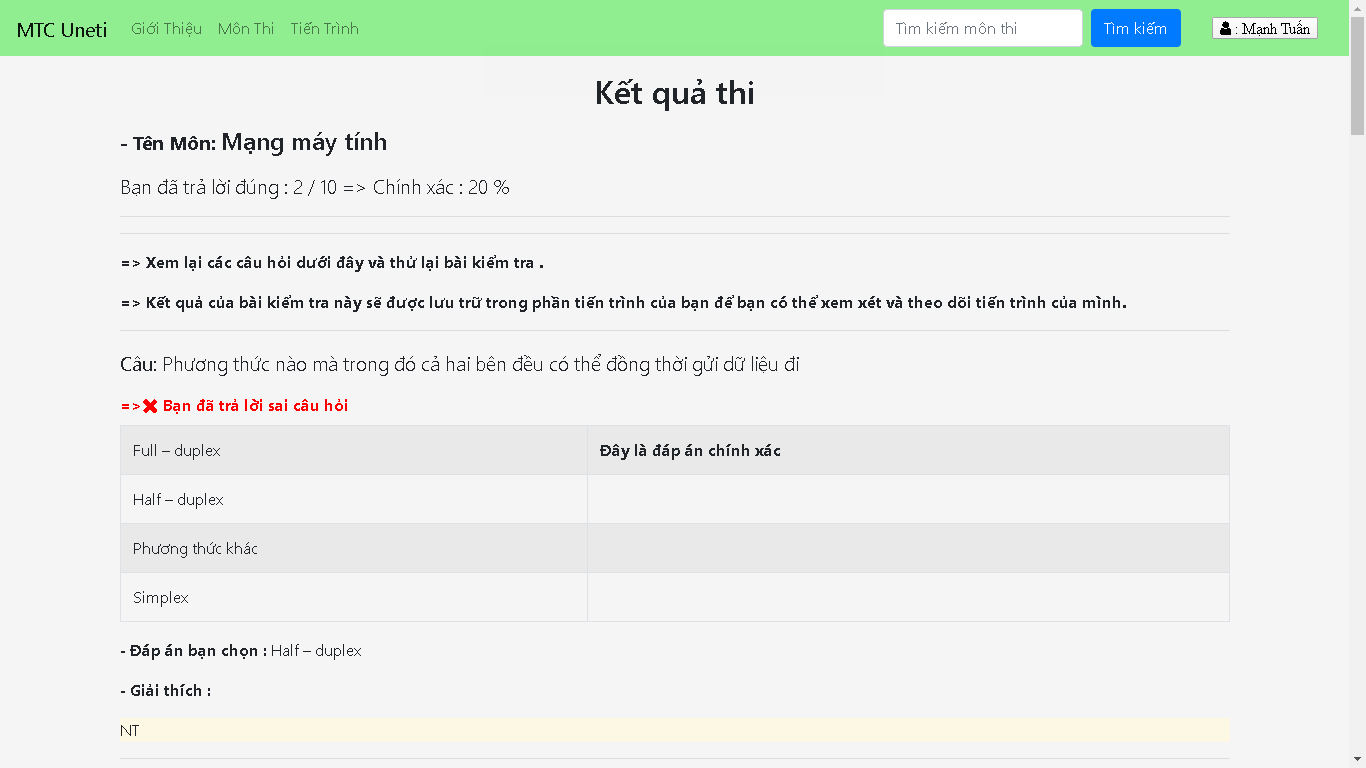
Trang sẽ hiển thị lần lượt các câu hỏi trắc nghiệm với 4 đáp án tương ứng. Người dùng chọn đáp án phù hợp và click “Tiếp” để di chuyển sang câu tiếp theo.



Hình 4. 10: Giao diện trang kiểm tra

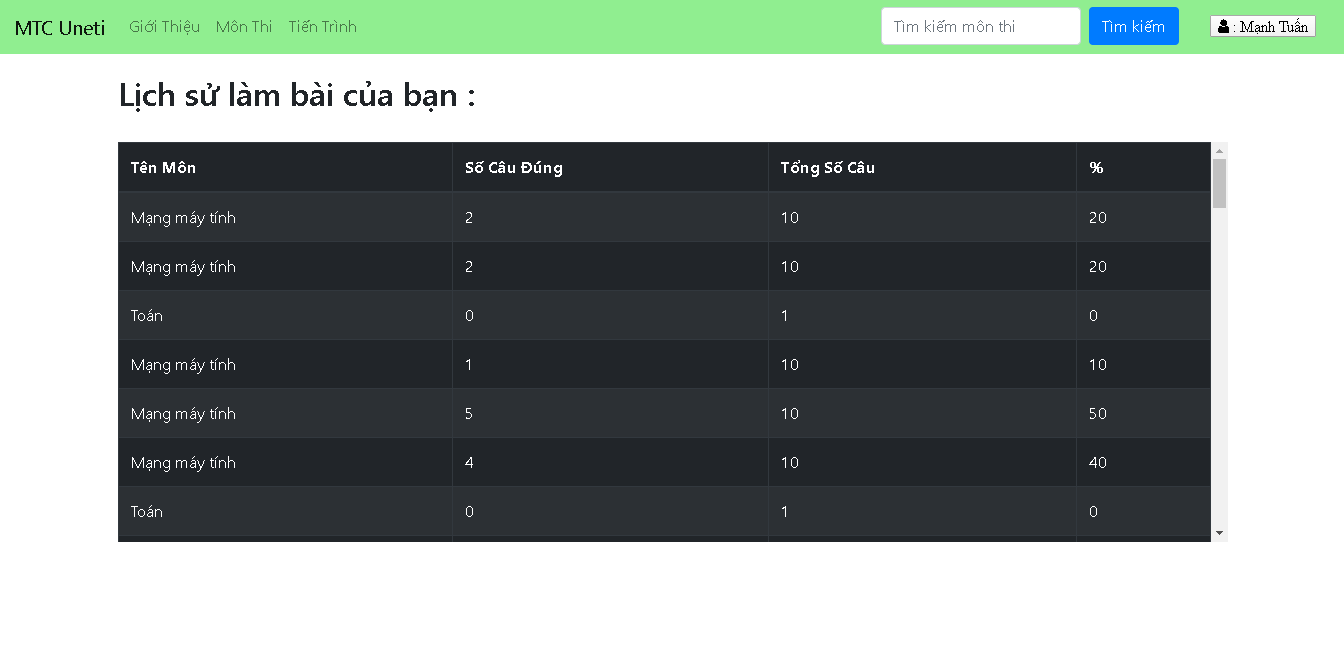
## 4.2.9 Giao diện trang kết quả

Trang kết quả thi chứa các thông tin: tên môn, số câu trả lời đúng, kết quả bài thi sẽ được lưu trữ trong phần tiến trình để người dùng có thể xem xét.



Hình 4. 11: Giao diện trang kết quả

## 4.2.10 Giao diện trang lịch sử làm bài



Hình 4. 12:Giao diện trang lịch sử làm bài

# KẾT LUẬN

Sau một thời gian làm việc nghiêm túc và hiệu quả, đến nay đề tài “Xây dựng ứng dụng kiểm tra trắc nghiệm trực tuyến” đã hoàn thành về cơ bản theo đúng những yêu cầu về nội dung và thời gian như đã định. Trong quá trình nghiên cứu và thực hiện đề tài với quyết tâm cao nhưng do hạn chế về kinh nghiệm và kiến thức nên đề tài chắc chắn sẽ không thể tránh khỏi những thiếu sót. Rất mong nhận được ý kiến đóng góp của thầy cô để đề tài của chúng em được hoàn thiện hơn.

Kết quả đạt được:

* Xây dựng được website có thể ứng dụng được thực tế.
* Hiểu và lập trình thành công ngôn ngữ Python.
* Sử dụng thành công Django (một web framework của Python).

Hạn chế của đề tài:

* Chưa tạo được chức năng tìm kiếm nâng cao.
* Quản lí điểm chưa được chặt chẽ.
* Nội dung website còn 1 vài hạn chế.

Hướng phát triển:

* Thêm một số chức năng cần thiết cho Website.
* Tạo nguồn dữ liệu đa dạng và phong phú.

# TÀI LIỆU THAM KHẢO

[1]. Sách Python cơ bản, tác giả Võ Duy Tuấn.

[2]. Sách Ngôn ngữ lập trình Python, tác giả Lê Văn Tâm.

[3]. Sách Begin to code with python, tác giả Rob Miles.

[4]. Sách Deep learning with python, tác giả Francois Chollet.

[5]. Một số trang web tham khảo:

<https://www.hocpython.com/>

<https://www.python.org/>

<https://www.djangoproject.com/>

<https://phocode.com/python/django/ebook-django/>

<https://en.wikipedia.org/wiki/Django>