**Ngôn ngữ lập trình (programming language)** là dạng ngôn ngữ được chuẩn hóa theo một hệ thống các quy tắc riêng, sao cho qua đó người lập trình có thể mô tả các chương trình làm việc dành cho thiết bị điện tử mà cả con người và các thiết bị đó đều hiểu được.

Có nhiều **ngôn ngữ lập trình**. Các **ngôn ngữ lập trình khác nhau** có các quy tắc riêng khác nhau.

### Python => Python được phát triển để chạy trên nền unix

1. **Python là gì**

**Python là ngôn ngữ lập trình hướng đối tượng đa năng có cấu trúc dữ liệu cấp cao mạnh mẽ và hệ thống thư viện lớn**.  có tốc độ phát triển nhanh .

Python là ngôn ngữ lập trình đa chức năng có thể ứng dụng ở nhiều lĩnh vực khác nhau.

Python là lựa chọn phổ biến khi ta cần viết scripts cho mục đích test và monitor.

Python có khả năng kết hợp với các ngôn ngữ khác .

Không như PHP, Python không được xây dựng chuyên cho môi trường web và không có dủ tính năng web cốt lõi cần có. Bởi vậy, chúng ta phải sử dụng web framework để phát triển ứng dụng web trên Python.

Kể từ khi xuất hiện các framework tuyệt vời như Django, lập trình viên web đã bắt đầu sử dụng Python nhiều hơn.

***Một số điểm cộng cho Python:***

+ Cú pháp súc tích, cấu trúc rành mạch.

+ Tốc độ xử lý nhanh nhẹn.

+ Ứng dụng được trên mọi hệ điều hành (Windows, Linix, Unix, Mac OS, MS – DOS).

+ Rất nhiều người sử dụng (theo thống kê đến nay có hơn 400 triệu người).

+ Nhờ thư viện khổng lồ mà nó cũng tương thích cao với Unix, các phần mềm bên thứ ba, các phần cứng.

=>Lập trình web dùng Django ( youtube, google,…)

=> Data Analysis :

=>Lập trình robot

=> Lập trình game

=> lập trình ứng dụng . chạy trên nhìu hệ đhh

=>machine learning

=>Bâỏ mật moạng và máy tính

**Tại sao lại dùng Python?**

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ,

(**Lập trình hướng đối tượng là phương pháp lập trình lấy đối tượng làm nền tảng để xây dựng chương 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,

Python có nhiều công cụ debug dễ sự dụng.

Công việc quản lý package (module) trong Python cũng vô cùng dễ dàng

Python cũng rất cơ động.

**Lập trình Web bằng Python Web Frameworks**

Python không có sẵn khả năng làm web

Với nhu cầu ngày càng đa dạng, lập trình viên nên chuyển sang một số framework Python như Django hay Flask để có thê cho ra các ứng dụng web phức tạp.

Đa phần những framework này cần phải đi kèm với dịch vụ web như Apache khi tiến đến production, và đây cũng là lúc khuyết điểm xuất hiện. So với PHP, quá trình deploy của Python trên remote server sẽ khá khó khăn

Các dịch vụ nổi tiếng như Instagram, Reddit, Quora và Disqus đều được phát triển trên nền Python

Ưu điểm:

* Là một ngôn ngữ có hình thức sáng sủa, cấu trúc roc ràng, cú pháp ngắn gọn
* 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.
* Tương thích mạnh mẽ với Unix, hardware, với số lượng thư viện khổng lồ
* [*Python*](http://itplus-academy.edu.vn/Khoa-hoc-lap-trinh-Python-PYTHON-FOR-EVERYBODY.html) 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.

Nhược điểm:

* [*Python*](http://itplus-academy.edu.vn/Khoa-hoc-lap-trinh-Python-PYTHON-FOR-EVERYBODY.html)**không có các thuộc tính** như :protected,private hay public, không có vòng lặp **do…while** và **switch….case.**
* Mặc dù tốc độ xử lý của Python nhanh hơn [*PHP*](http://itplus-academy.edu.vn/Lap-trinh-CC-cung-FPT-Software.html) nhưng không bằng[*JAVA*](http://itplus-academy.edu.vn/Khoa-hoc-lap-trinh-java-co-ban.html) và[*C++*](http://itplus-academy.edu.vn/Lap-trinh-CC-cung-FPT-Software.html).

**Tại sao các nhà lập trình web vẫn lựa chọn?**

Không có bất cứ một ngôn ngữ nào là hoàn hảo hết và

cũng không phải các lập trình web lại lựa chọ [*Python*](http://itplus-academy.edu.vn/Khoa-hoc-lap-trinh-Python-PYTHON-FOR-EVERYBODY.html) là ngôn ngữ sử dụng nhiều nhất mà là tất cả lợi ích của Pyton mang lại:

* Dễ học, không quá rườm rà trong việc khai báo. Code ngắn gọn nhằm giúp cho bạn lập trình một cách nhanh chóng và đạt hiệu quả cao.
* Có nhiều trang web lớn sử dụng ngôn ngữ lập trình này như: GOOGLE, Facebook, youtube,…
* Tạo nền tảng vững chắc để bước vào 1 thế giới lập trình.

. **Tính năng chính của Python**

* ***Ngôn ngữ lập trình đơn giản, dễ học:*** Python có cú pháp rất đơn giản, rõ ràng.
* ***Miễn phí, mã nguồn mở:*** Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.
* ***Khả năng di chuyển:*** Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào.

Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

* ***Khả năng mở rộng và có thể nhúng:***  có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác.

Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được. Hệ thống nhúng là một hệ thống được tích hợp cả phần cứng và phần mềm phục vụ cho các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hóa điều khiển, quan trắc và truyền thông.

* ***Ngôn ngữ thông dịch cấp cao:*** Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu.
* ***Thư viện tiêu chuẩn lớn:*** Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó

### *Hướng đối tượng:* Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP([Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming)), bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

**3. Lý do nên học ngôn ngữ lập trình Python**

* ***Là ngôn ngữ lập trình phổ biến nhất:***
* ***Sử dụng linh hoạt trong nhiều mảng lập trình***

Không chỉ nổi tiếng với mảng lập trình web, Python còn là ngôn ngữ được sử dụng rộng rãi trong lập trình robot và các hệ thống nhúng.

* ***Là bước đệm để học thêm các ngôn ngữ lập trình khác***

Python thân thiện với người dùng và hòa đồng với cả những ngôn ngữ lập trình khác. Như bạn đã biết, Python là ngôn ngữ hướng đối tượng có ứng dụng đa dạng. Và nếu bạn nắm rõ Python thì sẽ dễ dàng hơn khi học thêm các ngôn ngữ lập trình khác.

* ***Python được hỗ trợ rất tốt***

Với sự phổ biến của mình nên Python sở hữu cộng đồng hỗ trợ trực tuyến cực tốt. Có rất nhiều các trang web hỗ trợ người dùng Python cực tốt như: The Official Python Tutorial, Stack Overflow...

* ***Python hoàn toàn miễn phí***

Ngôn ngữ lập trình nào cũng miễn phí. Sự khác biệt giữa Python với các ngôn ngữ khác chính là việc nó hoàn toàn tương thích với các phần mềm mã nguồn mở (Open Sourse và GPL) vì Python cũng là mã nguồn mở.

## Các ứng dụng được viết bằng Python là gì?

Python có nhiều ứng dụng và được sử dụng bởi nhiều công ty lớn

* **Web development**: các trang web như Google, Dropbox, Instagram, Reddit… Đều được xây dừng bằng ngôn ngữ Python.
* **Khoa học máy tính, phân tích dữ liệu và machine learning**: Như đã chia sẻ, Python có số lượng thư viện khá lớn, rất nhiều trong số đó cung cấp các thuật toán hoặc cho phép bạn xây dựng các thuật toán để giải quyết các vấn đề phức tạp.
* **Script**: Dù công việc này có thể được thực hiện bởi nhiều ngôn ngữ khác, song Python vẫn là lựa chọn tốt nếu bạn có ý định viết các hàm toán học để phân tích hay nghiên cứu dữ liệu.

**Python** là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay. **Python** mang nhiều ưu điểm của một ngôn ngữ tương lai, cú pháp đơn giản, cấu trúc rõ ràng, số lượng thư viện lớn, mở nguồn mở, miễn phí, hướng đối tượng…

1. **Tại sao nên dùng 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 một framework với đầy đủ các thư viện, module hỗ trợ các web-developer.

Django được so sánh khá nhiều với Ruby on Rails.

**Tại sao nên dùng Django**

Nếu bạn đang có một ý tưởng hay ho, và muốn xây dựng ý tưởng đó trở thành hiện thực một cách nhanh nhất, thì Django luôn là một trong những lựa chọn hàng đầu mà bạn nên cân nhắc.

Django có đầy đủ các yếu tố phục vụ cho một Startup:

Nhanh: Django được thiết kế với triết lý làm sao để các lập trình viên đưa các ý tưởng trở thành một sản phẩm nhanh nhất có thể (trong một vài giờ đồng hồ chẳng hạn)

Có đầy đủ các thư viện/module cần thiết: Django có sẵn các thư viện về user authentication, content admin, site maps, RSS feed v.v…

Đảm bảo về tính bảo mật: Không còn các nỗi lo về các lỗi bảo mật thông thường như SQL Injection, cross-site scripting, cross-site request forgery hay clickjacking nữa. Django cũng cung cấp cả phương pháp để lưu mật khẩu an toàn nữa!

Khả năng mở rộng tốt: Django có thể đáp ứng lượng traffic lớn, nghĩa là bạn không cần phải lo lắng về khả năng scale sản phẩm của mình nữa.

Tính linh hoạt: Xây dựng CMS, hoặc Ecommerce Website, hay kể cả Social Network, Scientific Computing Platforms, tất cả đều có thể với Django.

PostgreSQL (Structured Query Language)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Công ty** | | | **Ngày phát hành phiên bản đầu tiên** | | **Phiên bản ổn định mới nhất** | [**Giấy phép sử dụng**](https://vi.wikipedia.org/w/index.php?title=Gi%E1%BA%A5y_ph%C3%A9p_s%E1%BB%AD_d%E1%BB%A5ng&action=edit&redlink=1) |
| PostgreSQL Global Development Group | | | Tháng 6 năm 1996 | | 12.3-2 | [BSD](https://vi.wikipedia.org/w/index.php?title=BSD_license&action=edit&redlink=1) |
| **Viết bằng** | [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) | |
| [**Hệ điều hành**](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) | [Đa nền](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) | |
| [**Thể loại**](https://vi.wikipedia.org/wiki/Th%E1%BB%83_lo%E1%BA%A1i_ph%E1%BA%A7n_m%E1%BB%81m)  Website: | [ORDBMS](https://vi.wikipedia.org/w/index.php?title=Object-relational_database&action=edit&redlink=1)  postgresql.org | |

**PostgreSQL** là một **hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng** (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay. xây dựng trên mã nguồn ban đầu của đại học Berkeley.

có nhiều đặc điểm hiện đại:

* Câu [truy vấn phức hợp](https://vi.wikipedia.org/w/index.php?title=Truy_v%E1%BA%A5n_ph%E1%BB%A9c_h%E1%BB%A3p&action=edit&redlink=1) (*complex query*)
* Khóa ngoại (*foreign key*)
* Thủ tục sự kiện (*trigger*)
* Các [khung nhìn](https://vi.wikipedia.org/w/index.php?title=Khung_nh%C3%ACn&action=edit&redlink=1) (*view*)
* Tính [toàn vẹn](https://vi.wikipedia.org/w/index.php?title=T%C3%ADnh_to%C3%A0n_v%E1%BA%B9n_(giao_d%E1%BB%8Bch)&action=edit&redlink=1) của các [giao dịch](https://vi.wikipedia.org/wiki/Giao_d%E1%BB%8Bch) (*integrity transactions*)

Hơn nữa, PostgreSQL có thể dùng trong nhiều trường hợp khác, chẳng hạn như tạo ra các khả năng mới như:

* Kiểu dữ liệu
* [Hàm](https://vi.wikipedia.org/wiki/H%C3%A0m)
* [Toán tử](https://vi.wikipedia.org/wiki/To%C3%A1n_t%E1%BB%AD)
* Phương pháp liệt kê
* Ngôn ngữ theo thủ tục
* Truy vấn xử lý song song (parallel query)
* Sao chép dữ liệu dạng luồng (Streaming replication)

Ngoài ra, PostgreSQL có thể được người dùng mở rộng bằng nhiều cách, ví dụ bằng cách thêm mới

* Loại dữ liệu
* chức năng
* khai thác
* Chức năng tổng hợp
* phương pháp chỉ mục
* ngôn ngữ thủ tục

**PostgreSQL** được phát triển dựa trên POSTGRES 4.2 tại phòng khoa học máy tính Berkeley, Đại học California.

* **PostgreSQL** được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, **PostgreSQL** sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.
* **PostgreSQL** là một phần mềm mã nguồn mở miễn phí. Theo đó, bạn sẽ được tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức.
* **PostgreSQL** không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên PostgreSQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

## Vì sao sử dụng PostgreSQL?

**PostgreSQL** sở hữu một hệ tính năng đa dạng giúp hỗ trợ các nhà phát triển xây dựng app, các nhà quản trị bảo vệ toàn vẹn dữ liệu, và tạo ra một môi trường chịu lỗi fault-tolerant giúp bạn quản lý dữ liệu bất kể tập dữ liệu lớn hay nhỏ. Bên cạnh hệ thống nguồn mở và miễn phí, **PostgreSQL** cũng có khả năng mở rộng tuyệt vời.

Ví dụ, bạn có thể định nghĩa các kiểu dữ liệu riêng của bạn, xây dựng các hàm tùy chỉnh, hay viết mã từ các ngôn ngữ lập trình khác nhau mà không cần biên dịch lại cơ sở dữ liệu!

**PostgreSQL** tuân theo tiêu chuẩn SQL.

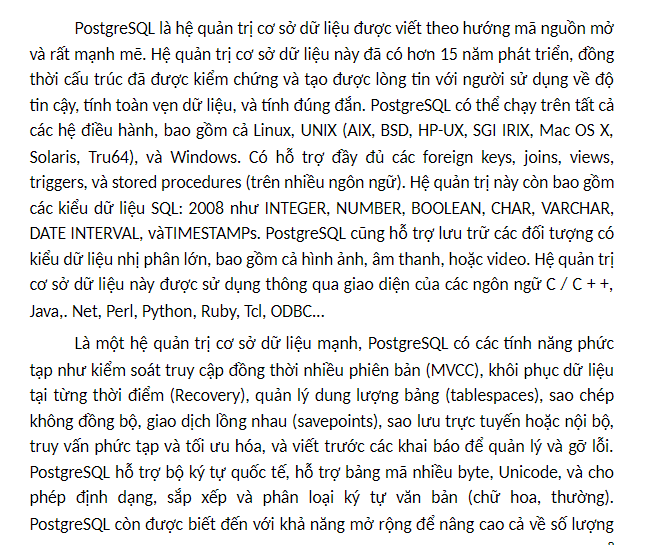
## Các yếu tố giúp PostgreSQL nổi bật

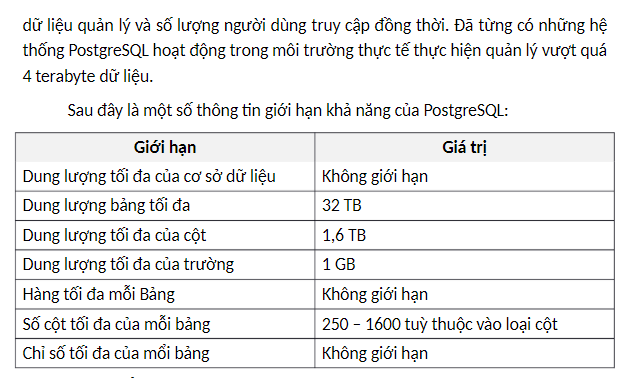
**PostgreSQL là hệ thống quản lý cơ sở dữ liệu** đầu tiên triển **khai tính năng kiểm soát đồng thời** nhiều phiên bản (Multi Version Concurrency Control (MVCC)) trước cả Oracle.

Là hệ thống quản lý cơ sở dữ liệu quan hệ đối tượng,

**PostgreSQL** cho phép thêm vào các tính năng tùy chỉnh được phát triển bằng các ngôn ngữ chương trình khác nhau như C/C , Java,...

Cộgn đồng sử sụng đông đảo





**Vài nét về lịch sử PostgreSQL:**

### Dự án POSTGRES Berkeley => Postgres95=> PostgreSQL

### CÀI ĐẶT

### PostgreSQL có thể được cài đặt bởi bất kỳ người dùng không có đặc quyền nào; không cần truy cập superuser ( root ).

## Kiến trúc cơ bản:

Trong thuật ngữ cơ sở dữ liệu, **PostgreSQL sử dụng mô hình máy khách / máy chủ**. Một phiên PostgreSQL bao gồm các quy trình (chương trình) hợp tác sau đây:

* Một quy trình máy chủ, quản lý các tệp cơ sở dữ liệu, **chấp nhận các kết nối đến cơ sở dữ liệu** từ các **ứng dụng khách** và thực hiện các hành động cơ sở dữ liệu thay mặt cho các máy khách. **Chương trình máy chủ cơ sở dữ liệu được gọi postgres**.
* Ứng dụng máy khách (frontend) của người dùng muốn thực hiện các hoạt động cơ sở dữ liệu. Các ứng dụng khách có thể rất đa dạng về bản chất: **máy khách** có thể là một công cụ định hướng văn bản, ứng dụng đồ họa, máy chủ web truy cập cơ sở dữ liệu để hiển thị các trang web hoặc công cụ bảo trì cơ sở dữ liệu chuyên dụng. Một số ứng dụng khách được cung cấp với bản phân phối PostgreSQL ; hầu hết được phát triển bởi người dùng.

**Máy chủ PostgreSQL có thể xử lý nhiều kết nối đồng thời từ các máy khách**

## Tạo cơ sở dữ liệu:

$ **createdb mydb**

## Xóa cơ sở dữ liệu:

$ **dropdb mydb**

## Tạo bảng:

CREATE TABLE cities (

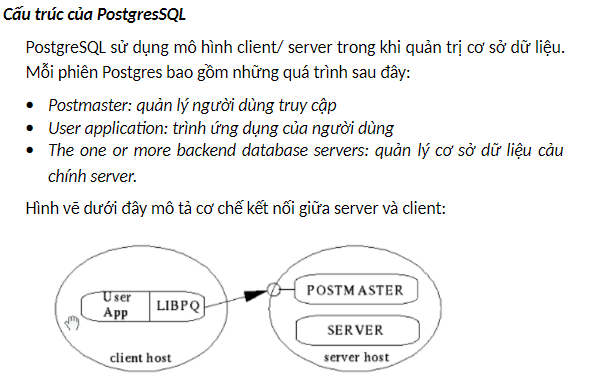
name varchar(80),

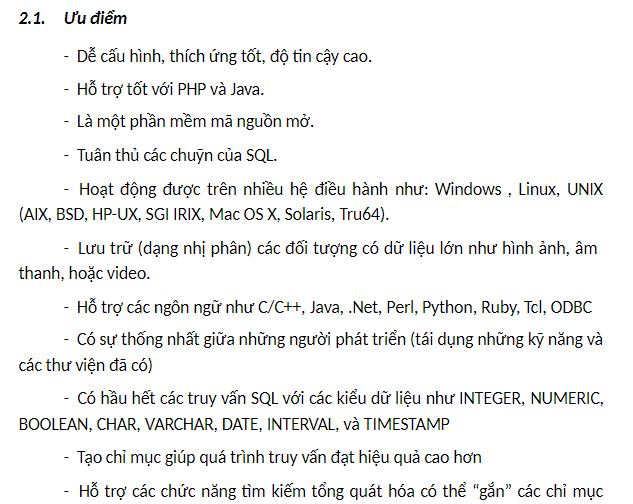
location point

);

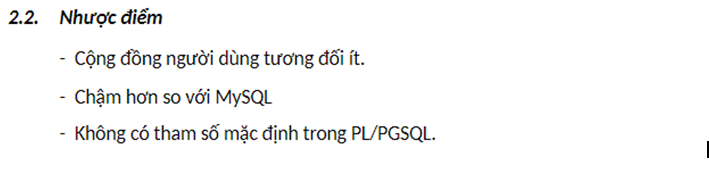
 Xóa **bảng:**

DROP TABLE ***tablename***;Truy vấn bảng:





* **Là một chuẩn SQL phù hợp với hệ quản trị cơ sở dữ liệu quan hệ:** PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, miễn phí và mạnh mẽ
* **Cộng đồng mạnh:** PostgreSQL được hỗ trợ bởi một cộng đồng nhiệt tình và đầy kinh nghiệm.
* **Hỗ trợ mạnh mẽ từ các bên thứ 3:** Bất kể các tính năng tiên tiến thì PostgreSQL đều được hỗ trợ bởi các công cụ tuyệt vời của bên thứ 3 trong việc quản lý cơ sơ dữ liệu
* **Sự mở rộng:** Nó có thể mở rộng lập trình PostgreSQL với các thủ tục lưu trữ, giống như một RDBMS tiên tiến.
* **Đối tượng hóa:** PostgreSQL không chỉ là một hệ quản trị cơ sở dữ liệu quan hệ, mà nó còn đối tượng hóa dữ liệu.



**Hosting:** Do các yếu tố được đề cập bên trên nên nó

## Khi nào nên sử dụng PostgreSQL

* **Toàn vẹn dữ liệu:** Khi độ tin cậy và tính toàn vẹn dữ liệu là tuyệt đối cần thiết thì PostgreSQL là lựa chọn tốt hơn hết.
* **Thủ tục phức tạp và khả năng tùy chỉnh cao:** Nếu bạn cần cơ sở dữ liệu cảu bạn thực hiện các thủ tục có khả năng tùy chỉnh thì PostgreSQL hỗ trợ rất tốt cho việc đó.
* **Sự đồng bộ:** Trong tương lai, nếu có phải chuyển từ hệ quản trị cơ sở dữ liệu quan hệ này sang một hệ quản trị cơ sở dữ liệu khác nhứ Oracle thì PostgreSQl cho việc chuyển đổi trong tương lại đó.
* **Thiết kế phức tạp:** So với sự thực hiện của các hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở miễn phí khác mà với thiết cơ sở dữ liệu phưc tạp thì PostgreSQL cung cấp cho chúng ta những giải pháp tối ưu.

## Đối tượng sử dụng PostgreSQL

Trên thị trường, có nhiều công ty đã xây dựng và cung cấp các sản phẩm, giải pháp sử dụng **PostgreSQL**. Một số công ty nổi bật trong số đó là Apple, Fujitsu, Red Hat, Cisco, Juniper Network, v.v.

## SO SÁNH

## Mô hình dữ liệu

PostgreSQL không chỉ là cơ sở dữ liệu quan hệ, nó là quan hệ hướng đối tượng. Điều này cung cấp cho nó một vài lợi thế so với các cơ sở dữ liệu SQL mã nguồn mở khác như MySQL, MariaDB và Firebird.

Một đặc điểm cơ bản của cơ sở dữ liệu quan hệ hướng đối tượng là hỗ trợ các đối tượng người dùng tự định nghĩa và các hành vi của chúng bao gồm các kiểu dữ liệu, các hàm, các thao tác, các tên miền và các chỉ mục. Điều này làm cho PostgreSQL cực kỳ mạnh và linh hoạt. Trong những cái khác, các cấu trúc dữ liệu phức tạp có thể được tạo ra, lưu trữ, và truy xuất.

## Các kiểu dữ liệu và cấu trúc

Có một danh sách các kiểu dữ liệu PostgreSQL hỗ trợ. Bên cạnh kiểu số, floating-point, chuỗi, boolean, và các kiểu dữ liệu mà bạn mong muốn (và nhiều tùy chọn khác), PostgreSQL tự hào với uuid, tiền tệ, liệt kê (enumerated), hình học (geometric), nhị phân (binary), địa chỉ mạng, chuỗi bit, tìm kiếm văn bản, xml, json, mảng, hỗn hợp, và các loại khoảng (range types), cũng như một vài kiểu **internal cho nhận biết đối tượng và vị trí đăng nhập**. Để công bằng, MySQL, MariaDB và Firebird mỗi cái có một vài loại ở mức độ khác nhau, nhưng PostgreSQL hỗ trợ tất cả.

**Địa chỉ mạng**

PostgreSQL cung cấp nhiều kiểu dữ liệu dành cho việc lưu trữ địa chỉ mạng. Kiểu dữ liệu **CIDR** (Classless Internet Domain Routing) theo quy ước cho địa chỉ mạng IPv4 và IPv6. Một vài ví dụ cho CIDR:

* 92.168.100.128/25

**Mảng nhiều chiều**

Bởi vì PostgreSQL là một cơ sở dữ liệu quan hệ hướng đối tượng, mảng các giá trị có thể lưu trữ hầu hết các kiểu dữ liệu có sẵn. Làm điều này bằng cách thêm cặp ngoặc [ ] vào sau kiểu dữ liệu chỉ định cho cột hoặc sử dụng ARRAY. Kích thước của mảng có thể chỉ định cụ thể, nhưng không bắt buộc.

 sự khác nhau giữa PostgreSQL và SQL Server

## ****1. Mô hình****

* **PostgreSQL** : là hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở và miễn phí, được duy trì bởi PostgreSQL Global Development Group và cộng đồng phát triển của nó.
* **SQL Server**: là hệ thống quản lý cơ sở dữ liệu phục vụ cho thương mại, được xây dựng và bảo trì bởi Microsoft.

## ****2. Hệ điều hành máy chủ được hỗ trợ****

* PostgreSQL được sử dụng khá phổ biến. Nó có sẵn rộng rãi trên nhiều hệ điều hành: FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows.
* [SQL Server](https://www.bkns.vn/sql-server-la-gi.html) được giới hạn trong Windows và gần đây là Linux.

## ****3. Thủ tục lưu trữ****

* PostgreSQL sử dụng bằng ngôn ngữ độc quyền **PL / pgSQL** hoặc với các ngôn ngữ phổ biến như Perl, Python, Tcl, v.v.
* SQL Server sử dụng ngôn ngữ Transact SQL và .NET.
* Oracle là phần mềm được cung cấp và hỗ trợ từ hãng Oracle với chi phí khá cao cho các license khi bạn muốn sử dụng nó để quản lý database.
* PostgreSQL là một DBMS hoàn toàn miễn phí vì nó là một phần mềm mã nguồn mở (open-source software). PostgreSQL hiện đang được hỗ trợ phát triển từ cộng đồng developer rất đông đảo. Không chỉ có thế mạnh về chi phí, vì bản thân nó là "mở" nên bạn có thể tối ưu (customize) DBMS này để phục vụ tốt hơn cho hệ thống.

## Định nghĩa cơ bản về Ubuntu

**Ubuntu** là một hệ điều hành do **cộng đồng phát triển** và là hệ điều hành tuyệt vời cho các máy tính xách tay, máy tính để bàn và cả máy chủ. Bất kỳ bạn sử dụng nó ở đâu, Ubuntu đều có tất cả các ứng dụng mà bạn luôn cần, từ các ứng dụng soạn thảo văn bản tới thư điện tử, từ phần mềm máy chủ web tới các công cụ lập trình.

**Ubuntu là hệ điều hành** được cung cấp **miễn phí (free of charge)**. Bạn không phải trả bất kỳ phí bản quyền nào. Bạn có thể tải nó về, sử dụng và chia sẻ Ubuntu với bạn bè, gia đình, nhà trường hoặc doanh nghiệp của bạn mà không vì bất cứ thứ gì một cách tuyệt đối.

Bạn có thể tham khảo và tải Ubuntu mới nhất tại trang chủ: [http://ubuntu.org](http://ubuntu.org/)

[Lên đầu trang ↑](https://topthuthuat.com/ubuntu-la-gi/#kcmenu)

## Hệ điều hành Windows không có an toàn?

Hiện nay, Windows là hệ điều hành được sử dụng phổ biến ở hầu hết người dùng tại Việt Nam vì Windows được hổ trợ nhiều phần mềm và bạn hoàn toàn có thể dễ dàng tìm được cách bẻ khóa phần mềm để sử dụng miễn phí.

Tôi đã từng bị 1 vụ keylog mất đi một vài tài khoản quan trọng mắc dù đã cài phần mềm diệt virus bản quyền. Thời gian đó quả thật là rất khó khăn. Đây là lý do mà tôi muốn thay đổi và [sử dụng Ubuntu](https://topthuthuat.com/ubuntu-la-gi/).

## Tại sao bạn nên sử dụng hệ điều hành Ubuntu

### 1. Hoàn toàn miễn phí

Ubuntu được cung cấp hoàn toàn miễn phí, bạn có thể vào trang chủ để tải ngay phần mềm về và cài đặt vào máy tính. Bạn có thể chỉnh sửa, sao chép hoặc cải tiến phần mềm dưới phép giấy phép GNU GPL.

### 2. Sử dụng it tài nguyên phần cứng

So với Windows thì Ubuntu tỏ ra khá vượt trội trong việc sử dụng ít tài nguyên và tăng hiệu suất hệ thống.

Trước đây tôi có 1 Laptop chay CPU Intel Core 2 Solo, Ram 2GB. Khi chạy Windows XP, chỉ cần mở Chrome và 2 Tab thì hệ thống có dấu hiệu chậm và “Đơ”. Khi chuyển qua Ubuntu, kết quả tôi đã có thể sử dụng bao nhiêu tab tùy thích. Bên cạnh đó, hiệu suất của máy cũng tăng đáng kể.

Chính vì thế, nếu bạn đang sở hữu 1 Laptop hoặc PC với cấu hình yếu, có lẽ Ubuntu là giải pháp tốt nhất dành cho bạn.

### 3. Tính bảo mật cao

Theo đánh giá của các chuyên gia từ các Website nước ngoài, về cơ chế bảo mật Ubuntu có độ bảo mật cao hơn Window. Và ở Ubuntu, khi có bất cứ lỗ hỏng **bảo mật** nào, trong thời gian rất ngắn bạn có thể cập nhật được ngay các bản vá lỗi bởi Ubuntu có 1 cộng động hổ trợ rộng lớn trên toàn thế giới. Các phiên bản Ubuntu mới và cũ đều được hỗ trợ cập nhập bảo mật ứng dụng trong một thời gian dài.

### 4. Khả năng tương thích cao

Khi cài đặt Window, vấn đề hàng đầu là phải tìm đầy đủ các**Drivers** để phần cứng có thể hoạt động ổn định. Vấn đề này sẽ không còn khi **cài đặt Ubuntu**, bạn sẽ nhận được hầu như tất cả các Driver trên máy tính. Nếu Driver gặp vấn đề, thông qua các bạn Update thường xuyên bạn sẽ khắc phục được một cách dễ dàng.

### 5. Kho ứng dụng miễn phí khổng lồ

Ubuntu có một **kho ứng dụng đồ sộ** và hầu hết các ứng dụng là **miễn phí**. Bạn sẽ không mất nhiều thời gian để lên các diễn đàn mô tả về phần mềm mình muốn, tất cả được phân chia theo mục đích sử dụng trong ứng dụng **Ubuntu Software Center**. Ngoài ra, bạn hoàn toàn có thể cài đặt tùy chỉnh các phần mềm khác.

### 6. Chạy được các ứng dụng trên Windows

Sử dụng Photoshop trên Ubuntu

Với **Wine và Steam, Ubuntu** bạn có thể sử dụng phần mềm hoặc chơi hầu hết các game trên Windows từ hệ điều hành Ubuntu. Tuy có vài hạn chế. Điều này cho thấy khả năng mở rộng của Ubuntu là rất lớn và tương lai không xa, Ubuntu có thể chạy được tất cả những thứ mà Windows có.

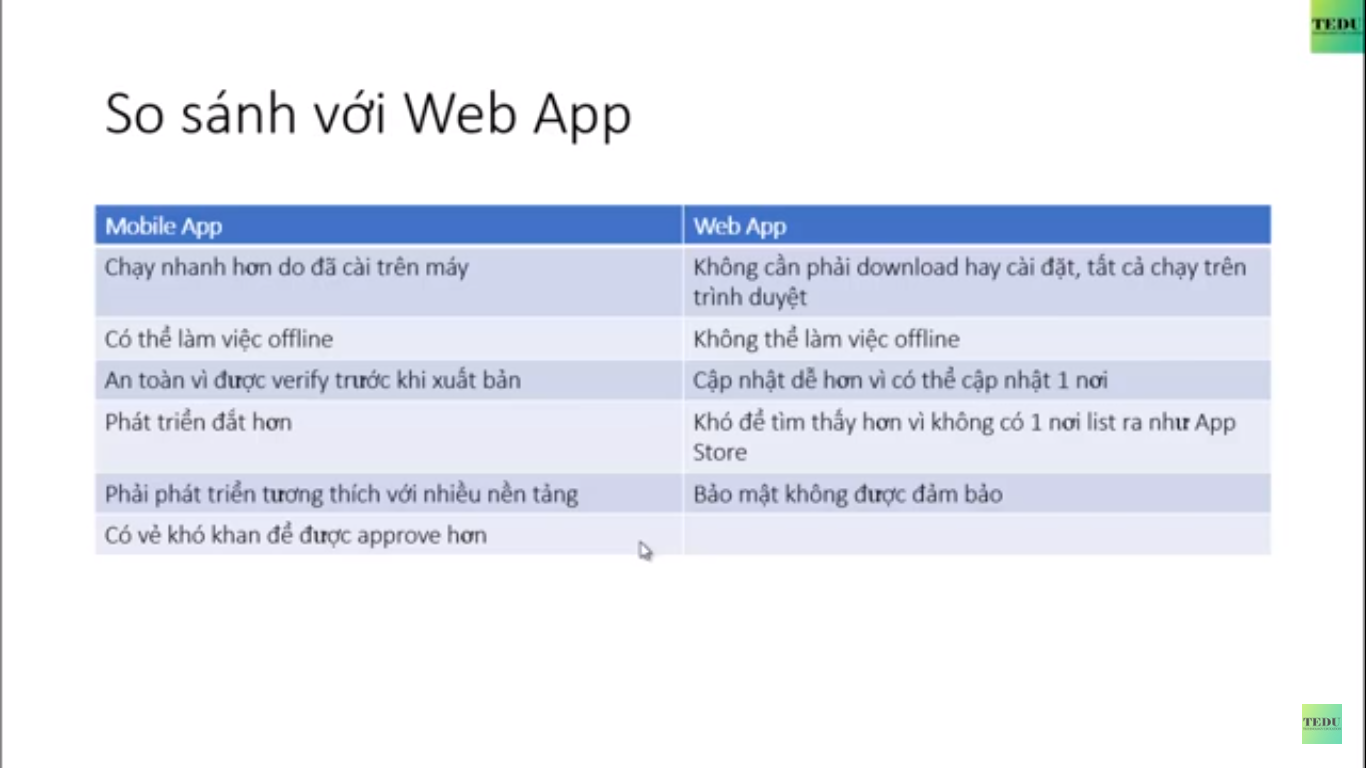
### 7. Cài đặt Ubuntu song song Windows

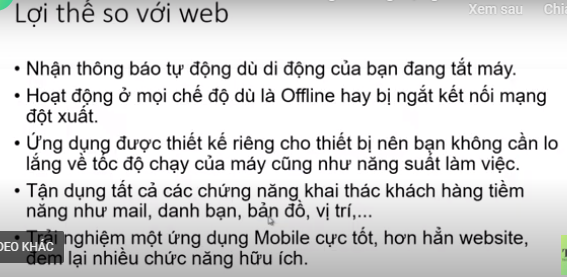
Bạn hoàn toàn có thể **cài đặt Ubuntu song song với Windows** trên cùng một ổ cứng. Bạn có thể chuyển đổi qua lại 2 hệ điều hành này thông qua menu boot. Điều này rất tuyệt vời, bởi bạn có thể sử dụng Windows để chạy các chương trình phức tạp hoặc dùng Ubuntu để giải trí hoặc khám phá những trải nghiệm mới mẻ trên Ubuntu.

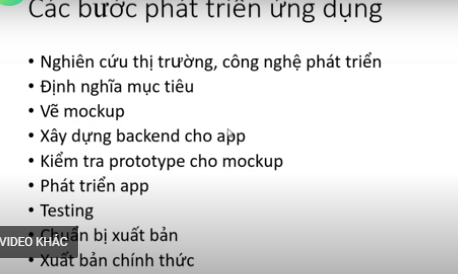
Để cài đặt Ubuntu song song với Windows 7 hoặc Windows 8, bạn có thể tham khảo bài viết hướng dẫn [tại đây](https://topthuthuat.com/cach-cai-dat-song-song-ubuntu-va-windows/).

1. **MOBIE APP**

App mobile hay ứng dụng này sẽ được các lập trình viên phát triển bằng nhiều ngôn ngữ khác nhau và cũng có thể phục vụ trên nhiều nền tảng thiết bị khác nhau, nhưng chủ yếu là chạy trên các thiết bị di động nên thường được gọi là ứng dụng di động.







**Mockup** chính là những mô hình thiết kế có kích thước thu nhỏ, hoặc bằng kích thước thật, dùng để mô phỏng ý tưởng thiết kế tới đối tượng khách hàng là doanh nghiệp (trong vai trò bạn là người phải trình bày ý tưởng thiết kế tới họ). Mockup có thể là file pdf, photoshop, hoặc là một mô hình có thật. **Mockup càng phức tạp và chi tiết, khách hàng của bạn càng nắm vững thông điệp và ý tưởng thiết kế mà bạn muốn truyền tải.**

**Phân loại các Mobile App phổ biến hiện nay**

**1. Native Mobile App**

Thì đây là loại ứng dụng di động mà người dùng khi muốn sử dụng sẽ phải download nội dung, cài đặt thiết bị, sau khi đã download và cài đặt sẽ có thể truy cập và sử dụng. Với loại Mobile App chúng ta vẫn thường gặp với những App như:

- Các ứng dụng game Offline

- Các ứng dụng tra cứu, phiên dịch

 . . .

Đối với các ứng dụng game, dường như tất cả những hình ảnh, âm thanh, chi tiết trong game đều đã được tải xuống trước đó. Và khi những nội dung này đã được tải xuống, người dùng họ có thể chơi game mà thiết bị không cần yêu cầu phải được kết nối internet. Riêng đối với những ứng dụng yêu cầu khi truy cập người dùng cần phải truy cập internet cho thiết bị thì những ứng dụng này đa số đều yêu cầu người dùng phải đăng nhập tài khoản, thực hiện các thao tác chuyển đổi giữa các tài khoản với nhau, chẳng hạn như mua bán vật phẩm bên trong game hay chuyển tiền giữa các tài khoản ứng dụng với nhau. Để kể ra một số cái tên thuộc nhóm ứng dụng Navive Mobile App chúng ta có Facebook, có Vimo, có Zalo . . .

**Những đặc trưng cơ bản của Native Mobile App:**

- Ứng dụng được phát triển phục vụ các thiết bị di động

- Các ứng dụng được phát triển phục vụ một hoặc cùng lúc nhiều hệ điều hành khác nhau

- Chi phí dành cho những Native Mobile App này là tương đối tốn kém

- Với các phiên bản được cập nhập, nội dung sẽ được làm mới

- Các ứng dụng này có thể được sử dụng mà không yêu cầu kết nối internet, chúng cũng thường mang lại những trải nghiệm khá tốt cho người dùng

- Các app càng chứa nhiều nội dung thì dung lượng tải về sẽ càng nặng và thường tập trung vào các ứng dụng game

**2. Hybrid Mobile Apps**

Các ứng dụng thuộc nhóm Hybrid Mobile Apps chính là những web app. Và web app ở đây sẽ là những những ứng dụng cũng được các lập trình việc tạo ra dựa trên các ngôn ngữ phổ biến hiện nay và nền tảng để các ứng dụng này được tạo ra đó là browser. Với các web app, người dùng họ cũng có thể truy cập, sử dụng và thực hiện các tương tác ngay trên app này

Hybrid Mobile Apps là sự kết hợp giữa hoàn hảo giữa có web app và các mobile app dựa trên những đặc trưng cơ bản của 2 nhóm ứng dụng này. Những ngôn ngữ thường được lựa chọn để xây dựng nên các Hybrid Mobile Apps này đó chính là :

- HTML5

- Javascript

- CSS3

Mặc dù được viết bởi các ngôn ngữ trên nền web nhưng các app này được bao bọc vỏ ngoài một cách khá kỹ lưỡng và khi đó chúng có thể hiển thị trên các kho ứng dụng di động mà không gặp phải bất kỳ trục trặc nào

Các web app đang nhận được sự quan tâm rất lớn từ phía các doanh nghiệp, cửa hàng

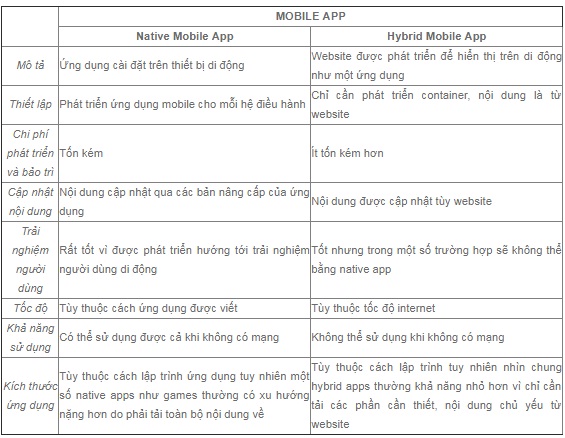
**Đặc trưng cơ bản của Hybrid Mobile Apps**

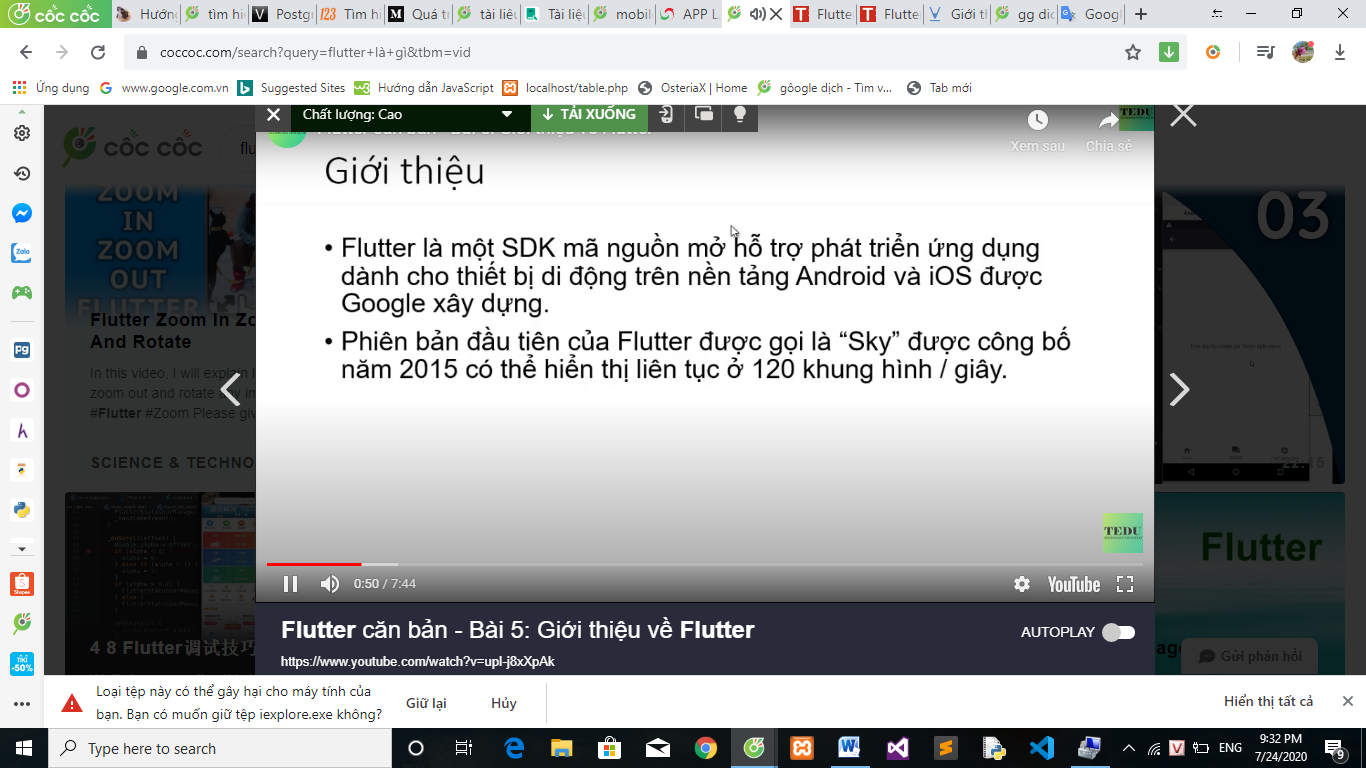
- Đó là các website được thiết kế như các ứng dụng di động

- Nội dung của các app này là lấy từ các website, chúng chỉ được phát triển vỏ ngoài để trở thành các ứng dụng

- Chi phí phát triển không yêu cầu quá cao, về phần nội dung thì phát triển như trên web

- Tốc độ load ứng dụng sẽ phụ thuộc vào tốc độ mạng và người dùng họ sẽ không thể truy cập ứng dụng nếu như thiết bi không có kết nối internet

**3.FLUTER**

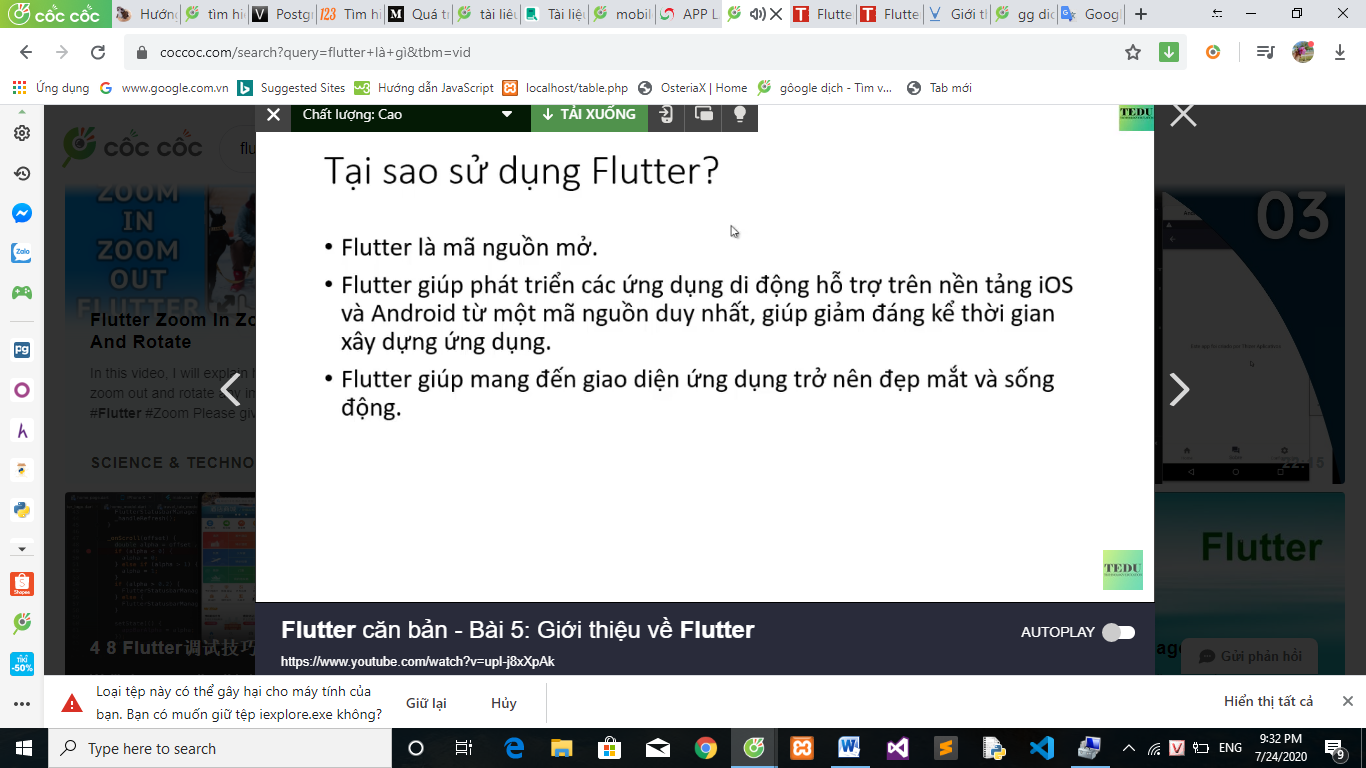


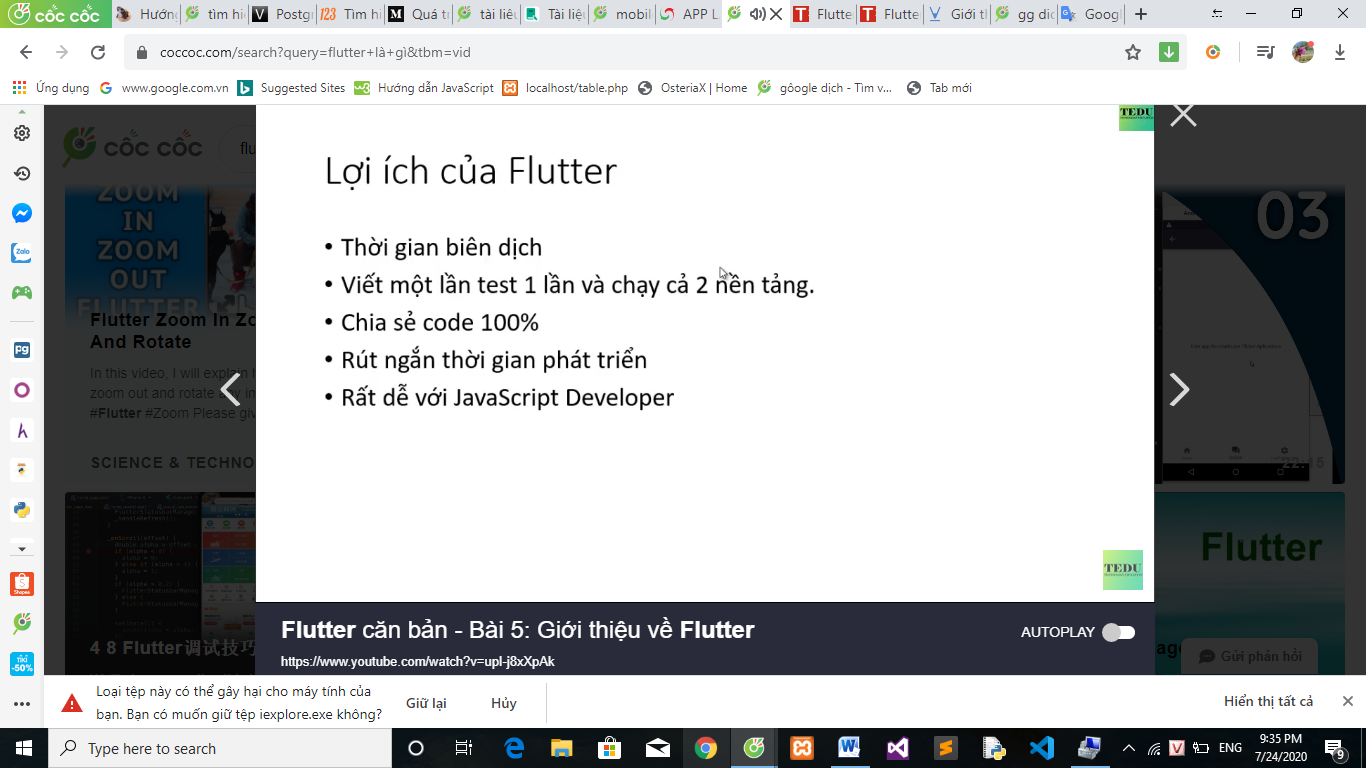
Flutter là bộ công cụ UI của Google để xây dựng các ứng dụng đẹp, được biên dịch nguyên bản cho [thiết bị di động](https://flutter.dev/docs),  từ một cơ sở mã duy nhất.

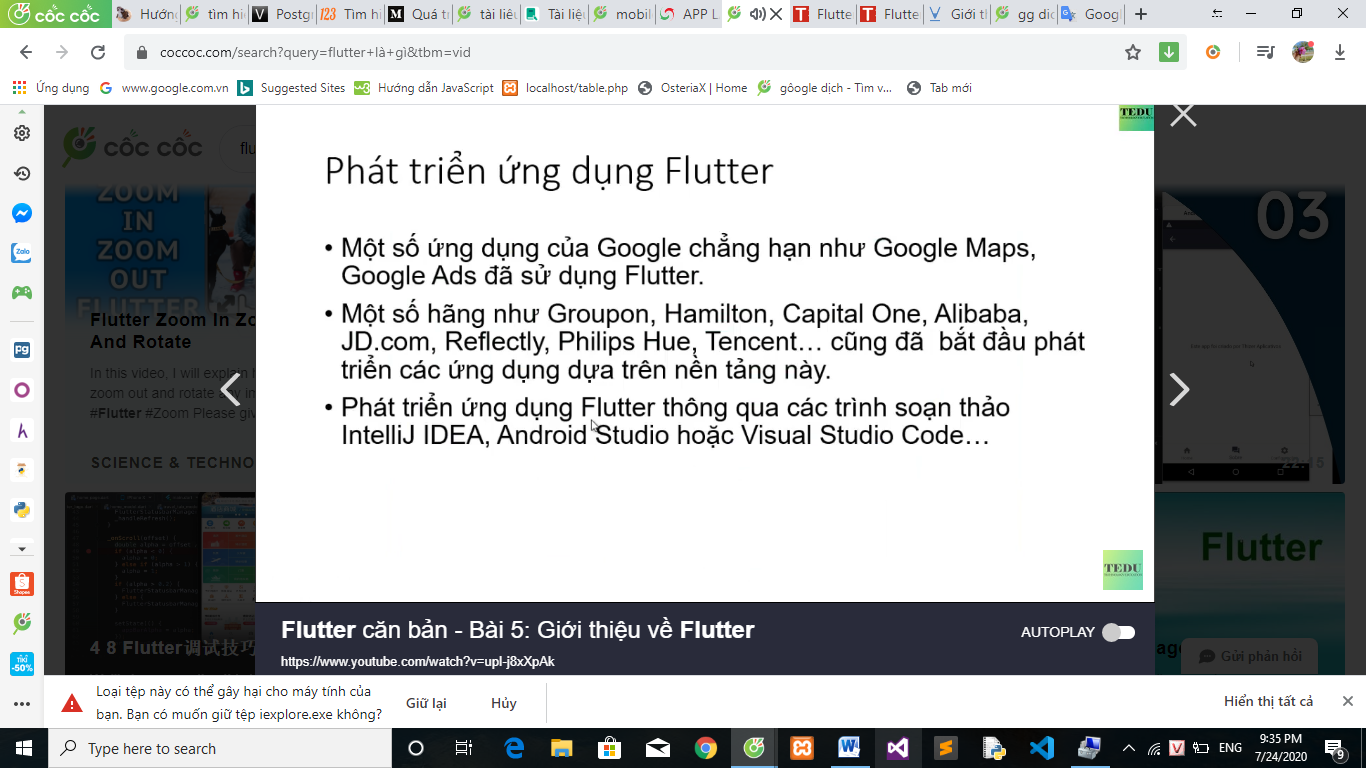
**Flutter sử dụng ngôn ngữ DART cũng do Google phát triển**

SDK: Software Development Kit **là** các công cụ và phần mềm dùng để phát triển ứng dụng thông qua một nền tảng nhất định

Phát triển Flutter có thể được thực hiện trên macOS, Linux, hoặc Window







#### Flutter: Ưu điểm

* Mạnh về hiệu ứng, hiệu suất ứng dụng rất cao.
* Giao tiếp gần như trực tiếp với hệ thống
* Ngôn ngữ kiểu tĩnh nhưng với cú pháp hiện đại (tương tự JS, Python, Java), compiler linh động khi dùng AOT (cho sản phẩm cuối) và JIT (cho quá trình phát triển với hot reload)
* Có thể chạy được giả lập mobile ngay trên web, tiện cho việc phát triển. Các bộ đo lường chỉ số hiệu suất được hỗ trợ sẵn giúp lập trình viên kiểm soát tốt hiệu suất của ứng dụng.  
  – Có thể dùng để xây dựng các nền tảng gắn vào ứng dụng native để tăng hiệu suất.

#### Flutter: Nhược điểm

* Bộ render UI được nhóm phát triển gần như viết lại, không liên quan tới UI có sẵn của Framework native, dẫn đến memory sử dụng khá nhiều. Hơn nữa, các UI không đi chung với OS, mà được phát triển riêng, nghĩa là cùng 1 phiên bản Flutter khi tạo ra ứng dụng cho iOS thì iOS 8.x -> 12.x đều y chang nhau, tương tự như với Android. Nhưng UI của Android thì tất nhiên khác với iOS.
* Phải học thêm ngôn ngữ DART: lập trình viên biết về DART không nhiều, cũng có rủi ro là học xong DART sẽ dính liền luôn với DART ở mảng phát triển ứng dụng mobile. Chứ không uyển chuyển như JS hay Python có thể nhảy qua lại giữa front, back hay AI…

**ĐỒ HỌA**

Graphic Design được xem là loại hình nghệ thuật ứng dụng có sự kết hợp hình ảnh, chữ viết và ý tưởng một cách đầy sáng tạo để nhằm mục đích truyền đạt thông tin một cách hiệu quả và thú vị thông qua các hình thức ấn phẩm in ấn và trực tuyến.

**Một trong những vấn đề cơ bản trong đồ họa trên thiết bị Android là màn hình có rất nhiều loại kích cỡ và hình dạng.**

**OpenGL (Open Graphics Library)**

**OpenGL là bộ thư viện đồ họa có khoảng 150 hàm giúp xây dựng các đối tượng và thao tác cần thiết trong các ứng dụng tương tác 3D.**

OpenGL được thiết kế như một giao diện độc lập với phần cứng được sắp xếp hợp lý để được triển khai trên nhiều nền tảng phần cứng khác nhau. Để đạt được những phẩm chất này, không có lệnh nào để thực hiện các tác vụ cửa sổ hoặc lấy đầu vào của người dùng được đưa vào OpenGL; thay vào đó, bạn phải làm việc thông qua bất kỳ hệ thống cửa sổ nào kiểm soát phần cứng cụ thể mà bạn đang sử dụng. Tương tự, OpenGL không cung cấp các lệnh cấp cao để mô tả các mô hình của các đối tượng ba chiều. Các lệnh như vậy có thể cho phép bạn chỉ định các hình dạng tương đối phức tạp như ô tô, các bộ phận của cơ thể, máy bay hoặc phân tử. Với OpenGL, bạn phải xây dựng mô hình mong muốn của mình từ một nhóm nhỏ *nguyên thủy hình học* - điểm, đường và đa giác.

Những thứ **OpenGL không hỗ trợ**

• bản thân OpenGL không có sẵn các hàm nhập xuất hay thao tác trên window, • OpenGL không có sẵn các hàm cấp cao để xây dựng các mô hình đối tượng, thay vào đó, người dùng phải tự xây dựng từ các thành phần hình học cơ bản ( điểm, đoạn thẳng, đa giác).

Rất may là một số thư viện cung cấp sẵn một số hàm cấp cao được xây dựng nên từ OpenGL. GLUT (OpenGL Utility Toolkit) là một trong số đó và được sử dụng rộng rãi. Trong tài liệu này, chúng ta sẽ sử dụng chủ yếu là OpenGL và GLUT.

OpenGL được thiết kế nhằm thỏa mãn mục đích chính sau:

- Đơn giản hóa việc tương tác giữa các mô hình không gian 3 chiều bằng một giao diện lập trình thống nhất.

- Hỗ trợ tối đa các chức năng của giao diện OpenGL bằng cách ép buộc các phần cứng 3 chiều khác nhau phải tương thích. Ngay cả khi không thể ép phần cứng hỗ trợ hoàn toàn, OpenGL có thể yêu cầu hệ thống sử dụng thêm sức mạnh phần mềm để xử lý.

- Tiêu chuẩn OpenGL nhận các nguyên hàm hình học như điểm, đường thẳng và đa giác rồi chuyển thành các điểm đồ họa (pixel) trên màn hình. Quá trình này được thực hiện thông qua luồng ống dẫn đồ họa (thuật ngữ graphics pipeline). Một tên gọi khác của OpenGL cũng được giới kỹ thuật chia sẻ đó là bộ máy trạng thái OpenGL.

Những thứ OpenGL hỗ trợ là các hàm đồ họa

• xây dựng các đối tượng phức tạp từ các thành phần hình học cơ bản (điểm, đoạn, đa giác, ảnh, bitmap),

• sắp xếp đối tượng trong 3D và chọn điểm thuận lợi để quan sát,

• tính toán màu sắc của các đối tượng (màu sắc của đối tượng được quy định bởi điều kiện chiếu sáng, texture của đối tượng, mô hình được xây dựng hoặc là kết hợp của cả 3 yếu tố đó),

• biến đổi những mô tả toán học của đối tượng và thông tin màu sắc thành các pixel trên màn hình (quá trình này được gọi là resterization). 2. Cấu trúc lệnh trong OpenGL OpenGL sử dụng tiền tố gl và tiếp theo đó là những từ được viết hoa ở chữ cái đầu để tạo nên tên của một lệnh, ví dụ glClearColor(). Tương tự, OpenGL đặt tên các hằng số bắt đầu bằng GL\_ và các từ tiếp sau đều được viết hoa và cách nhau bởi dấu ‘\_’, ví dụ: GL\_COLOR\_BUFFER\_BIT.

**OpenGL Utility Toolkit (GLUT)** Để khắc phục một số **nhược điểm** của OpenGL, GLUT được tạo ra với với nhiều hàm hỗ trợ

• quản lý window

• display callback

• nhập xuất (bàn phím, chuột,…)

• vẽ một số đối tượng 3D phức tạp (mặt cầu, khối hộp,…)

**Opengl ES chủ yếu được sử dụng trên điện thoại di động và web (webgl). Theo thông số kỹ thuật, trình điều khiển opengl máy tính để bàn của bạn có thể hỗ trợ tất cả opengl es apis.**

**OpenGL ES (và OpenGL)**[**không được hỗ trợ**](https://en.wikipedia.org/wiki/Deprecation)**trong các hệ điều hành của Apple, nhưng vẫn hoạt động tối thiểu trên iOS 12.**

**OpenGL ES**  (**Open Graphics Library for Embedded System**)

### OpenGL® ES là API ([Application Programming Interface](https://thuthuat.taimienphi.vn/api-la-gi-30030n.aspx) )đa nền tảng, miễn phí bản quyền để hiển thị đồ họa 2D và 3D tiên tiến trên các hệ thống nhúng (Hệ thống nhúng là một hệ thống được tích hợp cả phần cứng và phần mềm phục vụ cho các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hóa điều khiển, quan trắc và truyền thông.) và di động

### - bao gồm bảng điều khiển, điện thoại, thiết bị và phương tiện. Nó bao gồm một tập hợp con OpenGL được xác định rõ ràng phù hợp với các thiết bị có công suất thấp và cung cấp giao diện linh hoạt và mạnh mẽ giữa phần mềm và phần cứng tăng tốc đồ họa.

**Cơ bản OpenGL ES trong Android**

Có 2 classes cơ bản trong Android framework cho phép bạn tạo và điều khiển đồ họa với OpenGL ES

API: GLSurfaceView và GLSurfaceView.Renderer.

Nếu muốn sử dụng được OpenGL trong ứng dụng của bạn thì việc đầu tiên bạn cần làm đó là hiểu được cách triển khai thực hiện 2 classes này.

* GlSurfaceView: là một View mà bạn sẽ vẽ, thao tác các đối tượng trên đó.
* Renderer: là một interface định nghĩa các methods(phương pháp) được yêu cầu để vẽ đồ họa trên GlSurfaceVew

### OpenGL ES 3.2 - Chức năng OpenGL bổ sung

Phiên bản mới nhất trong sê-ri, OpenGL ES 3.2 đã bổ sung chức năng bổ sung dựa trên [Gói mở rộng Android](https://www.khronos.org/registry/OpenGL/extensions/ANDROID/ANDROID_extension_pack_es31a.txt) cho OpenGL ES 3.1, giúp mang chức năng của API di động gần hơn đáng kể so với đối tác trên máy tính để bàn - OpenGL.

### OpenGL ES 3.1 - Đưa tính toán vào đồ họa di động

Mặc dù chỉ là một cú hích trong bản sửa đổi nhỏ của API, OpenGL ES 3.1 là một cột mốc lớn đối với API, vì nó đã thêm khả năng tính toán mục đích chung trong API, mang lại tính toán cho đồ họa di động.

### OpenGL ES 3.0 - Đồ họa nâng cao

OpenGL ES 3.0 là một bước tiến hóa khác của OpenGL ES, đáng chú ý là bao gồm nhiều mục tiêu kết xuất, khả năng kết cấu bổ sung, bộ đệm đồng nhất, phản hồi và chuyển đổi phản hồi.

### OpenGL ES 2.0 - Tạo bóng lập trình

OpenGL ES 2.0 là API đồ họa di động đầu tiên để hiển thị các trình đổ bóng lập trình trong thế hệ phần cứng đồ họa mới nhất. Nó vẫn là một API phổ biến hiện nay và vẫn là API đồ họa 3D có sẵn rộng rãi nhất và vẫn là một lựa chọn vững chắc để nhắm mục tiêu vào phạm vi rộng nhất của các thiết bị trên thị trường.

### OpenGL ES 1.X - Đồ họa chức năng cố định

OpenGL ES 1.0 và 1.1 là các API đồ họa di động đầu tiên, được xác định liên quan đến thông số kỹ thuật OpenGL 1.5, cung cấp khả năng tăng tốc đồ họa chức năng cố định

**SO SÁNH OPENGL VÀ OPENGL ES**

Khi bạn sử dụng **opengl** bình thường, bạn có thể sử dụng glBegin và glEnd để bao quanh các nguyên hàm hình học bạn cần vẽ, nhưng khi sử dụng **Opengl ES**, bạn sẽ phải sử dụng **bộ đệm đỉnh**.

OpenGL ES 1.0 được phát hành công khai vào ngày 28 tháng 7 năm 2003. OpenGL ES 1.0 dựa trên API OpenGL 1.3 ban đầu, với nhiều chức năng được loại bỏ và thêm một chút. Một điểm khác biệt đáng kể giữa OpenGL và OpenGL ES là OpenGL ES đã loại bỏ nhu cầu đóng khung các cuộc gọi thư viện OpenGL với glBeginvà glEnd.

Sự khác biệt đáng kể khác là ngữ nghĩa gọi cho các hàm kết xuất nguyên thủy đã được thay đổi theo hướng có các mảng đỉnh và [các](https://en.wikipedia.org/wiki/Fixed-point_arithmetic) kiểu dữ liệu [điểm cố định](https://en.wikipedia.org/wiki/Fixed-point_arithmetic) được giới thiệu cho tọa độ đỉnh. Các thuộc tính cũng được thêm vào để hỗ trợ tốt hơn cho khả năng tính toán của các bộ xử lý nhúng, thường thiếu [đơn vị dấu phẩy động](https://en.wikipedia.org/wiki/Floating_point_unit) (FPU). Nhiều chức năng và kết xuất nguyên thủy khác đã bị xóa trong phiên bản 1.0 để tạo giao diện nhẹ

**WEBGL**

**Giới thiệu cơ bản về WebGL**

**1.WebGL là gì?**

WebGL (Web Graphics Library) là một thư viện đồ họa dành cho web, nó được dẫn xuất từ OpenGL ES (thư viện đồ họa 2D và 3D trên hệ thống nhúng: điện thoại, đồ điện tử, xe cơ giới). WebGL cung cấp các chức năng cơ bản tương tự OpenGL ES và hoạt động tốt trên các phân cứng đồ họa 3D hiện đại.

WebGL là javascript API có thể sử dụng được trong HTML5, WebGL được code trong tag <canvas> của HTML 5, điều này cho phép trình duyệt có thể truy cập và sử dụng GPU để xuất ra các đồ họa. WebGL được hỗ trợ bởi đa số các trình duyệt hiện đại: Chrome, FireFox, IE, Opera...

**2.Các ưu điểm của WebGL**

* Ứng dụng WebGL được viết bằng javascript nên các ứng dụng này có thể tương tác trực tiếp với các phần tử HTML, ngoài ra ta có thể sử dụng thêm các thư viện javascript và các công nghệ HTML để hỗ trợ cho ứng dụng.
* WebGL cũng hỗ trợ cho các nền tảng di động.
* WebGL mã nguồn mở.
* WebGL sử dụng javascript để code vì vậy nó được hỗ trợ tự động quản lý bộ nhớ.
* WebGL không cần thiết phải biên dịch để chạy.
* Dễ dàng thiết lập và chạy, chỉ cần một text editor (soạn thảo văn bản )và trình duyệt.

**4.GIT HUB**

GitHub là một **nền tảng lưu trữ mã để kiểm soát phiên bản** và cộng tác. Nó cho phép bạn và những người khác làm việc cùng nhau trên các dự án từ bất cứ đâu.

Đây cũng là nơi giúp các lập trình viên post các code, kế hoạch của mình lên để các thành viên khác trong nhóm dự án theo dõi, copy các đoạn code đó về và tiếp tục phần việc của mình.

#### Không cần mã hóa

Để hoàn thành hướng dẫn này, bạn cần có [tài khoản GitHub.com](http://github.com/) và truy cập Internet. Bạn không cần biết cách viết mã, sử dụng dòng lệnh hoặc cài đặt Git (phần mềm kiểm soát phiên bản GitHub được tích hợp sẵn).

Với khả năng lưu trữ và độ bảo mật cao của mình, **GitHub**được xem là nguồn lưu dữ liệu mở phổ biến nhất hiện nay. ***GitHub*** là sự kết hợp giữa 2 từ:

* ***Git – hệ thống quản lý dự án và phiên bản code***
* ***Hub – nơi biến những dòng lệnh trên Git thành mạng xã hội cho lập trình viên***

Nhưng cách sử dụng GitHub như thế nào?

Nó sử dụng để hợp tác nhiều người lại với nhau, từ mọi nơi trên thế giới, lên kế hoạch, theo dõi và làm chung một dự án.

[**GitHub**](https://github.com/) cũng là một nền tảng lưu trữ online lớn nhất trên thế giới về các dự án nhiều người làm.

## Bước 1. Tạo một kho lưu trữ

Một **kho lưu trữ** thường được sử dụng để tổ chức một dự án duy nhất. Các kho lưu trữ có thể chứa các thư mục và tệp, hình ảnh, video, bảng tính và bộ dữ liệu - bất cứ thứ gì mà dự án của bạn cần. Chúng tôi khuyên bạn nên bao gồm README hoặc tệp có thông tin về dự án của bạn. GitHub giúp bạn dễ dàng thêm một cái cùng lúc bạn tạo kho lưu trữ mới của mình. Nó cũng cung cấp các tùy chọn phổ biến khác như tệp giấy phép.

hello-worldKho lưu trữ của bạn có thể là nơi bạn lưu trữ ý tưởng, tài nguyên hoặc thậm chí chia sẻ và thảo luận mọi thứ với người khác.

### Để tạo một kho lưu trữ mới

1. Ở góc trên bên phải, bên cạnh hình đại diện hoặc nhận dạng của bạn, nhấp vào và sau đó chọn **Kho lưu trữ mới** .
2. Đặt tên cho kho lưu trữ của bạn hello-world.
3. Viết một mô tả ngắn.
4. Chọn **Khởi tạo kho lưu trữ này với README** .

## Bước 2. Tạo một chi nhánh

**Phân nhánh** là cách để làm việc trên các phiên bản khác nhau của một kho lưu trữ cùng một lúc.

**Để tạo một chi nhánh mới**

1. Chuyển đến kho lưu trữ mới của bạn hello-world.
2. Nhấp vào thả xuống ở đầu danh sách tập tin có **nhánh: chính** .
3. Nhập tên chi nhánh readme-edits, vào hộp văn bản chi nhánh mới.
4. Chọn hộp **Tạo nhánh** màu xanh hoặc nhấn vào Enter Enter trên bàn phím của bạn.

**5. VPS**

***VPS hay Virtual Private Server là máy chủ ảo, được tạo ra bằng cách chia server vật lý thành nhiều server khác***. **Các máy chủ ảo** này có tính năng tương tự như máy chủ riêng. VPS hoạt động dưới dạng chia sẻ source từ server vật lý ban đầu

**Khi nào thì sử dụng VPS?**

Ngày nay VPS được sử dụng để phục vụ các mục đích như:  
Lưu trữ các dịch vụ của website  
Làm một máy chủ game  
Tạo các môi trường ảo để lập trình, nghiên cứu…  
Xây dựng các hệ thống mail server, web server…  
Chạy các chương trình truyền thông

**Máy chủ ảo có thể cài được những hệ điều hành nào?**

Máy chủ ảo có thể cài được các hệ điều hành windows (bao gồm windows Server, windows dành cho cá nhân như windows 7, 8.1, 10...) tất cả hệ sinh thái linux đều có thể cài đặt: CentOS, ubuntu, fedora...) Ngoài ra Máy chủ có thể cắm thêm card dời (card màn hình) => có thể chạy các ứng dụng nặng, yêu cầu card màn hình. Hoặc đáp ứng nhu cầu IoT, machine learning...

**Ứng dụng chủ yếu của máy chủ ảo:**

Máy chủ ảo ra đời chủ yếu nhằm tiết kiệm ngân sách cho Tổ chức / cá nhân có nhu cầu sử dụng máy chủ với cấu hình thấp. thay vì phải trả vài triệu 1 tháng chi phí cho máy chủ, ngày nay chỉ cần trả vài chục ngàn (vài đô la) cho mỗi máy chủ ảo / 1 tháng.

Đối với máy chủ ảo chạy hệ điều hành windows, chủ yếu phục vụ webserver, File Server, email server hoặc các ứng dụng local / ứng dụng đặc thù cho doanh nghiệp. Mà bộ cài đặt yêu cầu sử dụng nền tảng windows, windows server hoặc các ứng dụng liên quan.

Đối với máy chủ ảo cài đặt hệ điều hành linux, trên 90% ứng dụng cho webserver. còn lại là ứng dụng email server hoặc các ứng dụng khác chạy mã nguồn mở PHP. Phổ biến nhất là website chạy wordpress rất phù hợp chạy trên máy chủ ảo linux.

**VPS** được tạo ra nhờ **công nghệ ảo hóa** thay vì sử dụng phần mềm quản lý (**hosting control panel**) thông thường để quản lý. Số lượng **VPS** luôn thấp hơn nhiều lần so với số lượng hosting nếu cài đặt trên cùng một hệ thống server, do đó tính ổn định và hiệu suất sử dụng tài nguyên của **VPS** luôn vượt trội so với hosting. Một **VPS** có thể chứa hàng trăm hosting khác nhau.”

Có thể so sánh giữa máy chủ ảo **VPS** và **máy chủ riêng** (**Dedicated Server**). Thay vì một server vật lý phục vụ cho một website thì **server ảo** có thể đáp ứng được nhiều website và lượng truy cập lớn hơn. Hình thức này sẽ tiết kiệm chi phí cho doanh nghiệp một cách hiệu quả hơn bao giờ hết.

Trong khi máy chủ ảo **VPS** dựa vào nền tảng vật lý thì **Cloud Server** được xây dựng bằng điện toán đám mây. Do đó khi chọn mua hoặc [**thuê VPS**](https://www.matbao.net/cloud-server-linux.html), khách hàng cũng cần lưu ý đến một số các thông số sau:

* RAM của máy chủ
* SWAP máy chủ ảo VPS
* Ổ cứng (disk)
* CPU Core
* Băng thông (Band Width)
* Thời gian Up-time của máy chủ ảo VPS
* Hệ điều hành máy chủ ảo VPS

***RAM của máy chủ***

Để giúp server VPS hoạt động mượt và ổn định, nên chọn sản phẩm có càng nhiều RAM càng tốt. Hầu hết các loại RAM đều có điểm tương đồng về mặt hiệu năng. Khi mua hay thuê VPS, nên hỏi kĩ nhà cung cấp rằng RAM cho server vật lý hay **server ảo**.

Ngoài ra, bạn cũng nên tuỳ thuộc vào lượng truy cập website mà chọn RAM nhiều hay ít. Với WordPress, lượng truy cập 5000/ngày và 100 user online cùng lúc, bạn cần tối thiểu 1GB RAM.

***SWAP máy chủ ảo VPS được hiểu là gì?***

SWAP là bộ nhớ ảo lưu lại các hành động xử lý cũ nếu như bộ nhớ RAM bị đầy. Bản thân SWAP là một không gian lưu trữ trên ổ cứng chứ không phải là một bộ nhớ độc lập. Không phải VPS nào cũng hỗ trợ bộ nhớ SWAP mà chỉ có các XEN VPS mới hỗ trợ SWAP.

***Ổ cứng (disk)***

Là không lưu trữ sẽ được sử dụng để lưu các file cài đặt của hệ điều hành và các file của mã nguồn website bạn lưu trên đó. Ổ đĩa hiện nay được chia làm 2 loại: [**HDD và SSD**](https://wiki.matbao.net/kb/ssd-la-gi-nhung-luu-y-khi-mua-o-cung-ssd/). Ổ cứng loại SSD thường giá sẽ đắt hơn loại ổ HDD.

***CPU Core***

Đối với CPU của Server, bạn cần quan tâm đến 2 chỉ tiêu quan trọng: số Core, tốc độ xung nhịp. Thường thì số core càng cao khả năng xử lý dữ liệu càng tốt. Ở các gói VPS, trung bình bạn sẽ được chọn từ 1 core đến 3 core.

***Băng thông (Band Width)***

Hãy ví độ lớn của băng thông cũng như độ rộng của đường phố. Phố càng rộng, giao thông càng thuận lợi và ngược lại. Các nhà cung cấp băng thông thường cung cấp lượng băng thông tối thiểu 10 – 20 Mbps.

***Thời gian Up-time của máy chủ ảo VPS là gì?***

Một lưu ý nữa bạn cần biết khi chọn mua hoặc **thuê server** đó là thời gian Up-time của nó. Thời gian up-time của **VPS** thường được ước lượng từ thời gian hoạt động của nó. Thời gian hoạt động của VPS từ 99.95 đến 99.9% thì bạn đều có thể chấp nhận mua được.

***Hệ điều hành máy chủ ảo VPS là gì?***

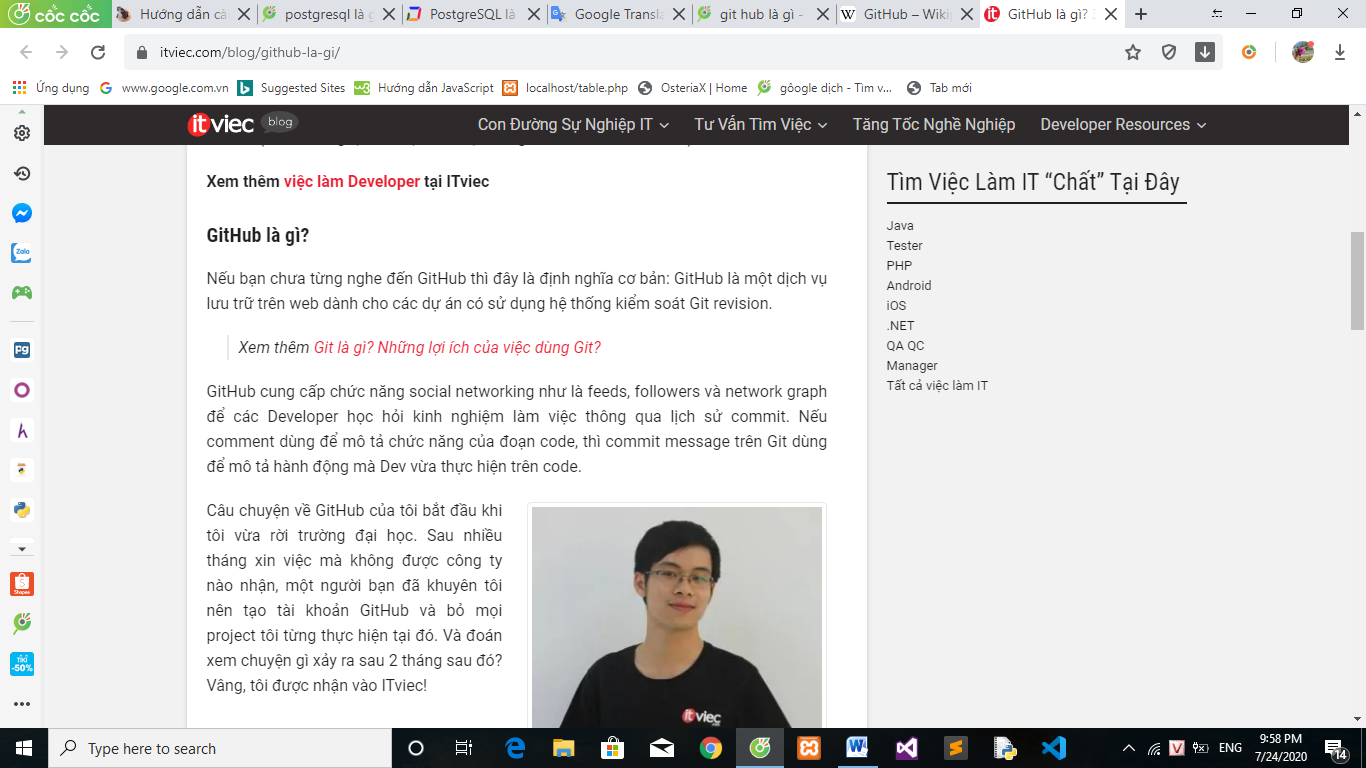
**Máy chủ ảo VPS** có 2 hệ điều hành phổ biến gồm: Linux và Window. Linux thân thiện người dùng, hỗ trợ ứng dụng nhiều hơn với chi phí ít hơn so với Window. Tuy nhiên, làm việc với Dot Net hoặc Visual Studio, bạn nên chọn **VPS Window**.

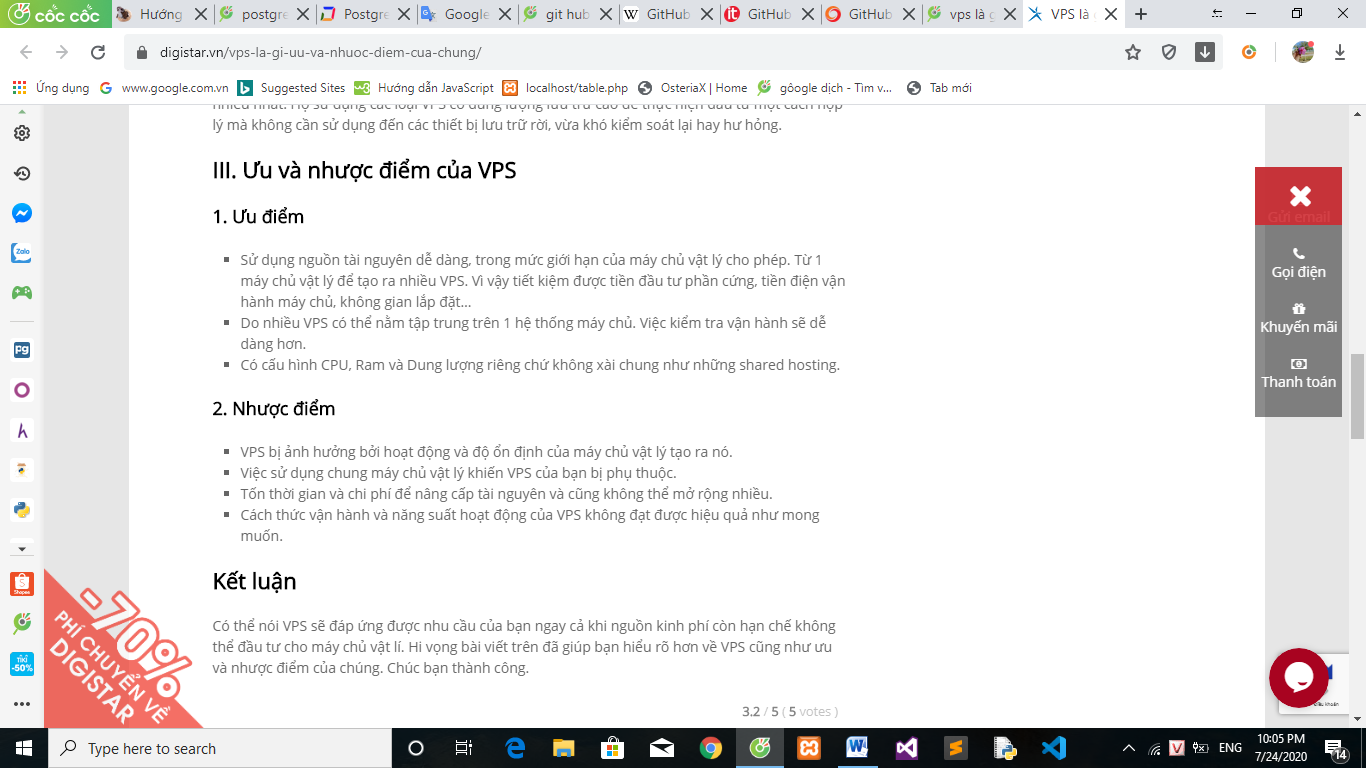
**Thông số VPS** là một trong các yếu tố quan trọng nhất mà bạn nên cân nhắc khi quyết định **mua VPS**.

**Chọn hệ điều hành nào cho VPS?**

Dưới đây là một số hệ điều hành Linux phổ biến nhất. Các bạn có thể cân nhắc sử dụng cho VPS:

* Hệ điều hành Debian
* Hệ điều hành Ubuntu
* Hệ điều hành CentOS
* Hệ điều hành Fedora





## Tìm hiểu VPS linux là gì?

**Máy chủ ảo VPS** - Virtual Private Server được phân chia từ một máy chủ vật lý, có tính chất như một máy chủ riêng biệt. Máy chủ ảo được tạo ra nhờ vào công nghệ ảo hóa, nó sẽ có những tính chất chia sẻ tài nguyên khác nhau tùy vào công nghệ ảo hóa được sử dụng. Cũng như một máy chủ vật lý, máy chủ ảo được chạy trên các nền tảng hệ điều hành khác nhau. Trong đó, VPS Linux được sử dụng rất phổ biến. Vậy, bài này chúng ta sẽ cùng **tìm hiểu VPS Linux** nhé.

Tìm hiểu VPS linux là gì?

**Có rất nhiều loại VPS khác nhau:**

* VPS Windows và VPS Linux là hai hệ điều hành phổ biến nhất  hiện nay
* Open VZ VPS, Xen VPS, KVM, VMWare… là công nghệ ảo hóa được sử dụng để phân chia máy chủ vật lý thành các máy chủ ảo khác nhau

**VPS Linux** là một máy chủ ảo được xây dựng trên nền tảng Linux, hiện nay Linux đang thống lĩnh thị trường máy chủ, có tới hơn 60% các máy chủ chạy hệ điều hành Linux.

**VPS Linux thường được sử dụng để làm:**

* Máy chủ [web hosting](https://vinahost.vn/hosting.html)
* Máy chủ hệ thống tên miền – DNS
* Máy chủ [thư điện tử](https://vinahost.vn/email-hosting.html)
* Tạo các môi trường ảo để lập trình, nghiên cứu và phân tích virus
* Các dịch vụ Web khác

Linux là phần mềm mã nguồn mở rẻ hơn Window, máy chủ hoạt động trên nền tảng Linux linh hoạt và hỗ trợ cho nhiều loại ứng dụng hơn, vì vậy mà hệ điều hành Linux được lựa chọn ưu tiên hơn. Tuy nhiên, vẫn có một số ứng dụng hỗ trợ hơn khi hoạt động trên Windows. Vì thế, khi bạn cần cài đặt sử dụng ứng dụng đó thì nên lựa chọn VPS Windows

OpenGL được thiết kế nhằm thỏa mãn mục đích chính sau:

- Đơn giản hóa việc tương tác giữa các mô hình không gian 3 chiều bằng một giao diện lập trình thống nhất.

- Hỗ trợ tối đa các chức năng của giao diện OpenGL bằng cách ép buộc các phần cứng 3 chiều khác nhau phải tương thích. Ngay cả khi không thể ép phần cứng hỗ trợ hoàn toàn, OpenGL có thể yêu cầu hệ thống sử dụng thêm sức mạnh phần mềm để xử lý.

- Tiêu chuẩn OpenGL nhận các nguyên hàm hình học như điểm, đường thẳng và đa giác rồi chuyển thành các điểm đồ họa (pixel) trên màn hình. Quá trình này được thực hiện thông qua luồng ống dẫn đồ họa (thuật ngữ graphics pipeline). Một tên gọi khác của OpenGL cũng được giới kỹ thuật chia sẻ đó là bộ máy trạng thái OpenGL.

**WEBGL**

**Giới thiệu cơ bản về WebGL**

**1.WebGL là gì?**

WebGL (Web Graphics Library) là một thư viện đồ họa dành cho web, nó được dẫn xuất từ OpenGL ES (thư viện đồ họa 2D và 3D trên hệ thống nhúng: điện thoại, đồ điện tử, xe cơ giới). WebGL cung cấp các chức năng cơ bản tương tự OpenGL ES và hoạt động tốt trên các phân cứng đồ họa 3D hiện đại. WebGL là javascript API có thể sử dụng được trong HTML5, WebGL được code trong tag <canvas> của HTML 5, điều này cho phép trình duyệt có thể truy cập và sử dụng GPU để xuất ra các đồ họa. WebGL được hỗ trợ bởi đa số các trình duyệt hiện đại: Chrome, FireFox, IE, Opera...

**2.Các ưu điểm của WebGL**

* Ứng dụng WebGL được viết bằng javascript nên các ứng dụng này có thể tương tác trực tiếp với các phần tử HTML, ngoài ra ta có thể sử dụng thêm các thư viện javascript và các công nghệ HTML để hỗ trợ cho ứng dụng.
* WebGL cũng hỗ trợ cho các nền tảng di động.
* WebGL mã nguồn mở.
* WebGL sử dụng javascript để code vì vậy nó được hỗ trợ tự động quản lý bộ nhớ.
* WebGL không cần thiết phải biên dịch để chạy.
* Dễ dàng thiết lập và chạy, chỉ cần một text editor và trình duyệt.