

[Tìm việc IT \(https://itnavi.com.vn/job\)](https://itnavi.com.vn/job)[Kiến Thức Lập Trình \(https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh\)](https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh)▼[Frontend \(https://itnavi.com.vn/blog/category/front-end\)](https://itnavi.com.vn/blog/category/front-end)[Backend \(https://itnavi.com.vn/blog/category/backend\)](https://itnavi.com.vn/blog/category/backend)[Fullstack \(https://itnavi.com.vn/blog/category/fullstack\)](https://itnavi.com.vn/blog/category/fullstack)[Mobile \(https://itnavi.com.vn/blog/category/mobile\)](https://itnavi.com.vn/blog/category/mobile)[Devops \(https://itnavi.com.vn/blog/category/devops\)](https://itnavi.com.vn/blog/category/devops)[Khác \(https://itnavi.com.vn/blog/category/khac\)](https://itnavi.com.vn/blog/category/khac)[Tin Tức \(https://itnavi.com.vn/blog/category/tin-tuc\)](https://itnavi.com.vn/blog/category/tin-tuc)[Sự Kiện \(https://itnavi.com.vn/blog/category/su-kien\)](https://itnavi.com.vn/blog/category/su-kien)[Tin Tức Tuyển Dụng \(https://itnavi.com.vn/blog/category/tin-tuc-tuyen-dung\)](https://itnavi.com.vn/blog/category/tin-tuc-tuyen-dung)[Tài liệu lập trình \(https://itnavi.com.vn/blog/category/tai-lieu-lap-trinh\)](https://itnavi.com.vn/blog/category/tai-lieu-lap-trinh)<https://itnavi.com.vn/blog>

MENU



Tổng hợp các câu lệnh hay dùng khi code Arduino

ITNavi 08 Jul 2021 2101

Nếu là người mới dùng Arduino, bạn có thể quan tâm đến danh sách các lệnh hữu ích cho những hoạt động quan trọng nhất. Bạn cần chúng để điều khiển bo mạch Arduino và xác định các phép toán logic đơn giản.

Điểm hay của Arduino là bạn không thực sự phải học quá nhiều kiến thức lập trình. Chỉ cần đầu tư vào một bộ bo mạch Arduino và cài đặt IDE. Hầu hết các ví dụ lệnh đều có sẵn trong thư viện Examples. Nếu bạn muốn làm theo các ví dụ tham chiếu hiện có, tất cả những gì bạn phải làm là sửa đổi code một chút và tạo các sketch của riêng bạn.



Mục tiêu của bài viết này không phải là để trình bày một danh sách đầy đủ các thủ thuật trên Arduino, mà là để hiểu logic đằng sau các code. Điều quan trọng là phải biết cách áp dụng chúng trong mọi tình huống.

Thanh công cụ Arduino IDE

Verify	Quét code và báo cáo bất kỳ lỗi nào
Upload	Biên dịch code và upload nó lên bo mạch Arduino qua USB
New	Mở một Arduino sketch trống (chương trình viết bởi Arduino IDE)
Open	Mở danh sách các sketch đã lưu trong trình duyệt file
Save	Lưu sketch hiện tại

Serial MonitorMở Serial Monitor trong một cửa sổ mới

Những lệnh Arduino IDE cơ bản

Ngay khi bạn mở Arduino IDE, bạn sẽ được chào đón bởi các lệnh `setup()` và `loop()`. Đây là hai ví dụ duy nhất về sketch mà bạn tìm thấy trong hầu hết tất cả các code Arduino.

- `Setup()`: Mỗi khi sketch bắt đầu, lệnh `setup` sẽ giúp bạn khởi tạo các biến và bắt đầu sử dụng các thư viện. Tham khảo ví dụ về đèn LED kỹ thuật số nhấp nháy không có độ trễ trong bài cài đặt Arduino IDE.

```
void setup() {
  // đặt pin kỹ thuật số làm output:
  pinMode(ledPin, OUTPUT);
}
```

- `Loop()`: Một vòng lặp theo sau `setup` và thực sự là trái tim của chương trình, khiến nó phản hồi vô hạn với bất kỳ logic nào. Ví dụ, đoạn code trên là ví dụ về đèn LED nhấp nháy vô hạn, không có độ trễ.

```
void loop() {
  // đây là nơi bạn đặt code cần chạy mọi lúc.
  // kiểm tra xem đã đến lúc nhấp nháy đèn LED chưa...
}
```

Sau khi làm quen với các sketch, bạn cần biết những lệnh điều khiển. Những lệnh quan trọng nhất là:

- `Break`: Nếu bạn muốn thoát khỏi một lệnh, bạn cần nhấn `break`.

```
if (sens > threshold) {
  // bail out on sensor detect x = 0; break;
}
```

- `If or else`: Các lệnh logic khởi tạo một hành động mỗi khi một điều kiện được thỏa mãn. Một lần nữa, hãy quay lại ví dụ về đèn LED kỹ thuật số nhấp nháy. Ghi nhớ lệnh `loop()` trong đó code phải chạy vô hạn.

Các lệnh điều khiển hữu ích khác có thể cung cấp logic nhất định. Những lệnh bạn có thể sử dụng bao gồm:

- `return`: Trả về một giá trị nhất định.
- `while`: Một vòng lặp khác diễn ra liên tục trong một điều kiện nhất định. Ví dụ:

```
while (dist < 3 cm) do something repetitive 200 times)
```

- `goto`: Đúng như tên gọi của nó, lệnh này cho phép bạn đi đến một dòng nhất định trong code.

Boolean và các toán tử số học

Bên cạnh sketch và các lệnh kiểm soát, bạn phải biết một số Boolean và toán tử số học để chỉ huy các chương trình.

- Các toán hạng: Bằng (`=`), phép cộng (`+`), phép trừ (`-`), phép nhân (`*`) và phép chia (`/`).
- Các toán hạng nâng cao: Không bằng (`!=`), Nhỏ hơn hoặc bằng (`<=`), lớn hơn hoặc bằng (`>=`), phần dư (`%`).

Các biến quan trọng

Trong một số trường hợp, bạn cần đưa ra một vài biến để xử lý những phép toán logic khác nhau. Những biến quan trọng là:

- `HIGH|LOW`: Điều này mang lại giá trị cuối cùng cao và thấp cho các hằng số.

```
// Các biến sẽ thay đổi:
```

```
int ledState = LOW;
```

- `LED_BUILTIN`: Đưa ra số lượng chân LED (LED pin). Trong ví dụ trên về đèn LED nhấp nháy trên Uno, số chân LED là 13.

Các biến quan trọng khác cần nhớ bao gồm `True/false`, `sizeof()`, `void`, `int` và `string`. Chúng cũng giống như bất kỳ chương trình thông thường nào khác bao gồm Python, C++, v.v...

Các hàm cao cấp

Cuối cùng, bạn cần biết một vài hàm nâng cao để điều khiển bo mạch Arduino. Các hàm nâng cao bao gồm:

Các hàm Arduino tích hợp sẵn

- `digitalRead()`: Đọc giá trị từ một pin kỹ thuật số nhất định. Ngoài ra còn có `digitalWrite()`.
- `pinMode(PIN_NUMBER, INPUT/OUTPUT)`: Thiết lập pin tại vị trí `PIN_NUMBER` thành `INPUT` hoặc `OUTPUT`.
- `pinMode(PIN_NUMBER, INPUT_PULLUP)`: Thiết lập pin tại vị trí `PIN_NUMBER` trở thành đầu vào bằng cách sử dụng điện trở kéo lên (pull-up resistor) tích hợp của bo mạch Arduino.
- `analogRead(PIN_NUMBER)`: Đọc số chân analog `PIN_NUMBER` và trả về một số nguyên từ 0 đến 1023.
- `analogWrite(PIN_NUMBER, VALUE)`: Giả lập analog output `VALUE` bằng cách sử dụng PWM trên `PIN_NUMBER` (Lưu ý: Chỉ khả dụng trên các chân 3, 5, 6, 9, 10 và 11).
- `analogReference(DEFAULT)`: Sử dụng điện áp tham chiếu mặc định (5V hoặc 3.3V tùy thuộc vào điện áp bo mạch).
- `analogReference(INTERNAL)`: Sử dụng điện áp tham chiếu bên trong (1.1V cho ATmega168/328p, 2.56 cho ATmega 32U4/8).
- `analogReference(EXTERNAL)`: Sử dụng điện áp được áp dụng cho chân AREF làm điện áp tham chiếu (Lưu ý: chỉ 0-5V).

Các hàm liên quan đến thời gian

- `Delay()`: Hãy nhớ lại ví dụ về đèn LED nhấp nháy. Bạn có thể thêm độ trễ vào đó.
- `millis()`: Trả về thời gian tính bằng mili giây kể từ khi Arduino sketch bắt đầu chạy dưới dạng một số nguyên dài không dấu.
- `micros()`: Trả về thời gian tính bằng micro giây kể từ khi Arduino sketch bắt đầu chạy dưới dạng một số nguyên dài không dấu.
- `delayMicroseconds` (số nguyên): Trì hoãn thực thi chương trình trong (số nguyên) micro giây.

Các hàm toán học

- `min(i, j)`: Trả về giá trị thấp nhất trong hai giá trị `i` và `j`
- `max(i, j)`: Trả về giá trị cao nhất trong hai giá trị `i` và `j`
- `abs(i)`: Trả về giá trị tuyệt đối của `i`
- `sin(angle)`: Trả về sin của một góc tính bằng radian
- `cos(angle)`: Trả về cosin của một góc tính bằng radian
- `tan(angle)`: Trả về tang của một góc tính bằng radian
- `sqrt(i)`: Trả về căn bậc hai của `i`
- `pow(ơ số, số mũ)`: Tính lũy thừa bậc (số mũ) của (ơ số). Ví dụ: `pow(2, 3) == 8`
- `constrain(i, minval, maxval)`: Ràng buộc giá trị `i` giữa `minval` (giá trị nhỏ nhất) và `maxval` (giá trị lớn nhất)
- `map(val, fromL, fromH, toL, toH)`: Chuyển đổi `val` từ phạm vi này sang phạm vi khác
- `random(i)`: Trả về một số nguyên dài ngẫu nhiên nhỏ hơn `i`
- `random(i, j)`: Trả về một số nguyên dài ngẫu nhiên giữa `i` và `j`
- `randomSeed(k)`: Sử dụng giá trị `k` để seed hàm `random()`

Giao tiếp nối tiếp (serial communication)

- `Serial.begin(speed)`: Bắt đầu giao tiếp nối tiếp ở tốc độ xác định
- `Serial.end()`: Kết nối giao tiếp nối tiếp
- `Serial.print(DATA)`: Xuất DATA vào cổng nối tiếp. DATA có thể là ký tự, chuỗi, số nguyên và số thực dấu phẩy động (floating point number)
- `Serial.available()`: Trả về số ký tự có sẵn để đọc trong bộ đệm nối tiếp
- `Serial.read()`: Đọc ký tự đầu tiên trong bộ đệm nối tiếp (trả về -1 nếu không có dữ liệu)
- `Serial.write(DATA)`: Ghi DATA vào bộ đệm nối tiếp. DATA có thể là một ký tự, số nguyên hoặc array (mảng)
- `Serial.flush()`: Xóa bộ đệm nối tiếp sau khi giao tiếp chiều đi hoàn tất

Servo

- `Servo myServo`: Tạo biến `myServo` của loại Servo
- `myServo.attach(PIN_NUMBER)`: `myServo` được liên kết với pin tại vị trí `PIN_NUMBER`
- `myServo.write(angle)`: Ghi một góc từ 0 đến 180 cho servo gắn với `myServo`
- `myServo.writeMicroseconds(uS)`: Ghi một giá trị tính bằng micro giây vào servo được gắn với `myServo` (thường từ 1000 đến 2000 với 1500 là điểm giữa)
- `myServo.read()`: Trả về một số nguyên chứa góc hiện tại của servo trong khoảng từ 0 – 180
- `myServo.attached()`: Trả về true nếu servo được gắn vào pin
- `myServo.detach()`: Tách `myServo` bằng một pin đính kèm

Ngoài ra còn có:

- Các lệnh trên là một số lệnh hữu ích nhất trong việc xử lý các bo mạch Arduino trên IDE. Danh sách này rất hạn chế và chưa đầy đủ, nhưng nó có thể giúp bạn bắt đầu các dự án của mình. Để biết chi tiết, bạn cần nghiên cứu các sketch của những người đam mê Arduino khác và tìm hiểu xem bạn có thể học hỏi được gì từ họ.

Nguồn: Tổng hợp các câu lệnh hay dùng khi code Arduino (<https://itnavi.com.vn/blog/tong-hop-cac-cau-lenh-hay-dung-khi-code-arduino>)

BÌNH LUẬN

0 bình luận

Sắp xếp theo **Mó**

Viết bình luận...

Plugin bình luận trên Facebook

CHIA SẺ BÀI VIẾT



(<https://twitter.com/deathwisher>) (<https://facebook.com/deathwisher?er.php?>

u=https://twitter.com/vanblog/tong

hop- hop-

cac- cac-

cau- cau-

lenh- lenh-

hay- hay-

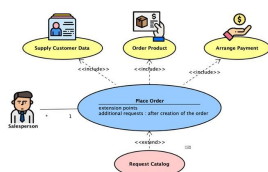
dung- dung-

khi- khi-

code- code-

```
arduino)
arduino)
```

BÀI VIẾT LIÊN QUAN



(<https://itnavi.com.vn/blog/use-case-la-gi>)

KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://itnavi.com.vn/blog/category/kiem-thuc-lap-trinh))

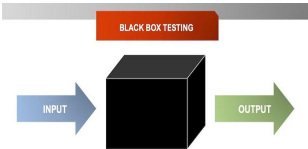
Use Case là gì? Bí quyết để xây dựng 1 sơ đồ Use Case hoàn hảo (<https://itnavi.com.vn/blog/use-case-la-gi>)



(<https://itnavi.com.vn/blog/dev-la-nghe-gi>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh))

Dev là nghề gì? Tổng hợp kiến thức về nghề lập trình viên (<https://itnavi.com.vn/blog/dev-la-nghe-gi>)

ITNavi 29 Sep 2020



(<https://itnavi.com.vn/blog/testcase-la-gi>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh))

Testcase là gì? Làm thế nào để tạo nên những biểu mẫu test case chất lượng? (<https://itnavi.com.vn/blog/testcase-la-gi>)

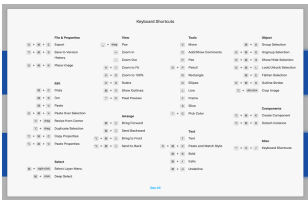
ITNavi 15 Mar 2021



(<https://itnavi.com.vn/blog/lap-trinh-nhung>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh))

Lập trình nhúng là gì? Kỹ thuật lập trình nhúng căn bản (<https://itnavi.com.vn/blog/lap-trinh-nhung>)

ITNavi 29 Jan 2021



(<https://itnavi.com.vn/blog/figma-la-gi>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh))

Figma là gì? Làm thế nào để cài đặt và sử dụng thành thạo Figma? (<https://itnavi.com.vn/blog/figma-la-gi>)

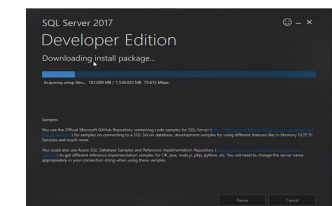
ITNavi 22 Apr 2021



(<https://itnavi.com.vn/blog/srs-la-gi>)

~~SRS là gì và tầm quan trọng của tài liệu này trong quy trình sản xuất phần mềm~~ (<https://itnavi.com.vn/blog/srs-la-gi>)

ITNavi 22 Apr 2021



(<https://itnavi.com.vn/blog/sql-server>)
KIẾN THỨC LẬP TRÌNH (<https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh>)

SQL Server là gì? Mục đích của việc sử dụng SQL Server (<https://itnavi.com.vn/blog/sql-server>)

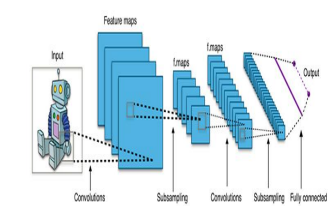
ITNavi 20 Oct 2020



(<https://itnavi.com.vn/blog/rest-api-la-gi>)
KIẾN THỨC LẬP TRÌNH (<https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh>)

Rest API là gì? Tìm hiểu về cách thức hoạt động của Restful API (<https://itnavi.com.vn/blog/rest-api-la-gi>)

ITNavi 18 Apr 2021



(<https://itnavi.com.vn/blog/neural-network-la-gi>)
KIẾN THỨC LẬP TRÌNH (<https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh>)

Tổng quan về Neural Network(mạng Nơ Ron nhân tạo) là gì? (<https://itnavi.com.vn/blog/neural-network-la-gi>)

ITNavi 13 May 2021

```
18 // Update is called once per frame
19 void Update () {
20     _offset.y += speed * Time.deltaTime;
21     _mat.SetTextureOffset ("MainTex", _offset);
22 }
23 }
```

(<https://itnavi.com.vn/blog/lap-trinh-unity>)
KIẾN THỨC LẬP TRÌNH (<https://itnavi.com.vn/blog/category/kien-thuc-lap-trinh>)

Lập trình Unity là gì và những kiến thức tổng quan cần nắm vững (<https://itnavi.com.vn/blog/lap-trinh-unity>)

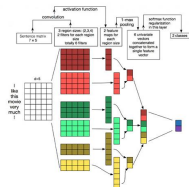
ITNavi 14 Oct 2020



(<https://itnavi.com.vn/blog/json-la-gi>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH))

JSON là gì? Hướng dẫn cách dùng cho người mới bắt đầu (<https://itnavi.com.vn/blog/json-la-gi>)

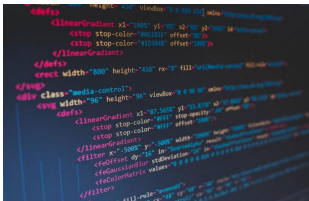
ITNavi 02 Feb 2021



(<https://itnavi.com.vn/blog/cnn-la-gi>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH))

Những Thông Tin Về Cấu Trúc Mạng CNN Là Gì? (<https://itnavi.com.vn/blog/cnn-la-gi>)

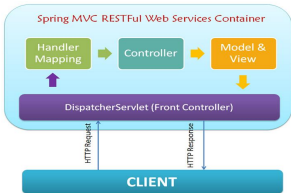
ITNavi 04 May 2021



(<https://itnavi.com.vn/blog/css-la-gi>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH))

CSS là gì? Vai trò và ưu điểm của CSS trên website (<https://itnavi.com.vn/blog/css-la-gi>)

ITNavi 02 Apr 2021



(<https://itnavi.com.vn/blog/spring-mvc-la-gi>)
KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH))

Tìm hiểu những lợi ích mà Spring MVC mang lại (<https://itnavi.com.vn/blog/spring-mvc-la-gi>)

ITNavi 17 May 2021




(<https://itnavi.com.vn/blog/quality-control-la-gi>)


ITNavi

06 Jul 2021


- MỚI NHẤT
- PHỔ BIẾN
- NGẪU NHIÊN




(https://itnavi.com.vn/blog/svg-la-gi)
SVG là gì? Tìm hiểu chi tiết về cách dùng và ứng dụng của SVG (https://itnavi.com.vn/blog/svg-la-gi)
ITNavi 16 Mar 2022




(https://itnavi.com.vn/blog/cach-rut-gon-link)
Top 7 các cách rút gọn link đơn giản dễ dàng nhất 2022 (https://itnavi.com.vn/blog/cach-rut-gon-link)
ITNavi 17 Feb 2022



(https://itnavi.com.vn/blog/stateless-la-gi)
Stateless là gì? Stateful là gì? Sự khác biệt giữa 2 loại (https://itnavi.com.vn/blog/stateless-la-gi)
ITNavi 15 Feb 2022



(https://itnavi.com.vn/blog/hybrid-app-la-gi)
Hybrid App là gì? Ưu, nhược điểm của ứng dụng lai Hybrid App (https://itnavi.com.vn/blog/hybrid-app-la-gi)
ITNavi 15 Feb 2022



(https://itnavi.com.vn/blog/entity-relationship-diagram-la-gi)
Entity Relationship Diagram là gì? Những thông tin hữu ích (https://itnavi.com.vn/blog/entity-relationship-diagram-la-gi)
ITNavi 10 Feb 2022

NEWSLETTER

Nhập địa chỉ email của bạn dưới đây để đăng ký nhận tin mới nhất

Your email address...

ĐĂNG KÝ

VIỆC LÀM IT LƯƠNG CAO

- Việc làm PHP (https://itnavi.com.vn/job/PHP)
- Việc làm JAVA (https://itnavi.com.vn/job/JAVA)
- Việc làm JAVASCRIPT (https://itnavi.com.vn/job/JAVASCRIPT)
- Việc làm C# (https://itnavi.com.vn/job/C#)
- Việc làm PYTHON (https://itnavi.com.vn/job/PYTHON)
- Việc làm RUBY (https://itnavi.com.vn/job/RUBY)
- Việc làm .NET (https://itnavi.com.vn/job/.NET)
- Việc làm NODEJS (https://itnavi.com.vn/job/NODEJS)

KẾT NỐI VÀ THEO DÕI



(https://www.facebook.com/ITNaviVN) (https://twitter.com/ITNaviVN) (https://www.youtube.com/channel/UCQSpIGXkODovl_TV4P8Q5w)

TIN TỨC ĐẶC BIỆT



(<https://itnavi.com.vn/blog/li-xi-trao-tay-nhan-ngay-qua-khung-khi-dang-tuyen-it-tai-itnavi>)

qua-khung-khi-dang-tuyen-it-tai-itnavi)

LÌ XÌ TRAO TAY - NHẬN NGAY QUÀ KHỦNG khi đăng tuyển IT tại ITNavi (<https://itnavi.com.vn/blog/li-xi-trao-tay-nhan-ngay-qua-khung-khi-dang-tuyen-it-tai-itnavi>)

ITNavi 06 Jan 2022



Chia sẻ những kiến thức mới nhất chính xác về công nghệ thông tin giúp bạn có những định hướng chính xác lĩnh vực công nghệ thông tin

DANH MỤC NỔI BẬT

- UNCATEGORIZED ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/UNCATEGORIZED](https://itnavi.com.vn/blog/category/uncategorized))
- TIN TỨC ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/TIN-TUC](https://itnavi.com.vn/blog/category/tin-tuc))
- TIN TỨC TUYỂN DỤNG ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/TIN-TUC-TUYEN-DUNG](https://itnavi.com.vn/blog/category/tin-tuc-tuyen-dung))
- KIẾN THỨC LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/KIEN-THUC-LAP-TRINH](https://itnavi.com.vn/blog/category/kiem-thuc-lap-trinh))
- SỰ KIỆN ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/SU-KIEN](https://itnavi.com.vn/blog/category/su-kiem))
- TÀI LIỆU LẬP TRÌNH ([HTTPS://ITNAVI.COM.VN/BLOG/CATEGORY/TAI-LIEU-LAP-TRINH](https://itnavi.com.vn/blog/category/tai-lieu-lap-trinh))

HOT TAGS

- [ITNavi \(https://itnavi.com.vn/blog/search?ITNAVI\)](https://itnavi.com.vn/blog/search?ITNAVI)
[PHP \(https://itnavi.com.vn/blog/search?PHP\)](https://itnavi.com.vn/blog/search?PHP)
[JAVA \(https://itnavi.com.vn/blog/search?JAVA\)](https://itnavi.com.vn/blog/search?JAVA)
- [JAVASCRIPT \(https://itnavi.com.vn/blog/search?JAVASCRIPT\)](https://itnavi.com.vn/blog/search?JAVASCRIPT)
[C# \(https://itnavi.com.vn/blog/search?C#\)](https://itnavi.com.vn/blog/search?C#)
[PYTHON \(https://itnavi.com.vn/blog/search?PYTHON\)](https://itnavi.com.vn/blog/search?PYTHON)
- [RUBY \(https://itnavi.com.vn/blog/search?RUBY\)](https://itnavi.com.vn/blog/search?RUBY)
[.NET \(https://itnavi.com.vn/blog/search?.NET\)](https://itnavi.com.vn/blog/search?.NET)
[NODEJS \(https://itnavi.com.vn/blog/search?NODEJS\)](https://itnavi.com.vn/blog/search?NODEJS)

