

# Những kỹ năng lập trình với Scratch

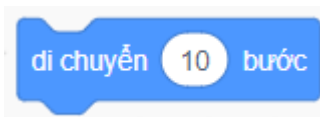
## (phần 2)

## Mục Lục

|  |           |
|--|-----------|
| <b>1. Hướng di chuyển và Di chuyển</b>                       | <b>2</b>  |
| 1.1. Di chuyển   | 2         |
| 1.2. Xoay  | 2         |
| 1.2.1. Xoay phải   | 2         |
| 1.2.2. Xoay trái   | 2         |
| 1.2.3. Giới thiệu tọa độ                                     | 3         |
| a. Tọa độ của nhân vật                                       | 4         |
| b. Tìm tọa độ của nhân vật ở vị trí bất kỳ                   | 5         |
| 1.2.4. Bộ câu lệnh đi tới                                    | 6         |
| 1.2.4.1. Câu lệnh đi tới                                     | 6         |
| 1.2.4.2. Câu lệnh đi tới vị trí xác định                     | 7         |
| 1.2.5. Bộ câu lệnh lướt                                      | 7         |
| 1.2.5.1. Câu lệnh lướt trong ... giây tới ...                | 7         |
| 1.2.5.2. Câu lệnh lướt xác định tọa độ                       | 7         |
| 1.2.6. Bộ câu lệnh hướng                                     | 8         |
| 1.2.6.1. Câu lệnh đặt hướng bằng                             | 8         |
| 1.2.6.2. Câu lệnh hướng về phía đối tượng                    | 9         |
| 1.2.7. Câu lệnh bật lại khi tiếp xúc với cạnh                | 9         |
| 1.2.8. Câu lệnh đặt kiểu xoay                                | 10        |
| 1.2.9. Thao tác trực tiếp lên tọa độ                         | 11        |
| 1.2.9.1. Câu lệnh đặt trực tiếp tọa độ                       | 11        |
| 1.2.9.2. Câu lệnh thay đổi tọa độ                            | 11        |
| 1.2.9.3. Hiển thị vị trí lên màn hình                        | 11        |
| <b>2. Câu lệnh Nếu... Thì...</b>                             | <b>13</b> |
| <b>3. Câu lệnh “Đang chạm ...?” và “Phím .... được bấm?”</b> | <b>15</b> |
| 3.1. Câu lệnh “Đang chạm ...?”                               | 15        |
| 3.2. Câu lệnh “Phím ... được bấm?”                           | 16        |
| <b>4. Khối của tôi</b>                                       | <b>18</b> |
| 4.1. Cách tạo Khối của tôi                                   | 18        |
| 4.2. Cách sử dụng Khối của tôi                               | 19        |

# 1. Hướng di chuyển và Di chuyển

## 1.1. Di chuyển



- Câu lệnh này dùng để làm nhân vật di chuyển về phía trước (Phía trước của nhân vật được quy định bởi yếu tố hướng di chuyển trong mục [Các thông số của nhân vật](#)). Số bước mà nhân vật di chuyển là số ở trong câu lệnh. Thầy/Cô có thể tùy ý chỉnh sửa số này để làm nhân vật di chuyển dài/ngắn theo ý mình mong muốn.
- **Ví dụ:** Như trong hình, câu lệnh này làm nhân vật di chuyển về phía trước một 10 bước.

## 1.2. Xoay

### 1.2.1. Xoay phải



- Câu lệnh này dùng để làm nhân vật quay về bên phải. Nhân vật quay bao nhiêu độ so với vị trí hiện tại phụ thuộc vào số trong câu lệnh. Số này có thể tùy chỉnh để nhân vật quay theo ý mình.
- **Ví dụ:** Khi kích hoạt câu lệnh trên, nhân vật sẽ quay một góc 30 độ sang bên phải như thầy/cô thấy trong hình dưới đây.



### 1.2.2. Xoay trái



- Tương tự như câu lệnh quay phải, câu lệnh quay trái này hoạt động với quy tắc giống hệt như câu lệnh quay phải.
- **Ví dụ:** Khi kích hoạt câu lệnh trên, nhân vật sẽ quay một góc 30 độ sang bên trái như thầy/cô thấy trong hình dưới đây.



### 1.2.3. Giới thiệu tọa độ

The image shows a Scratch stage with a Cat sprite. Below the stage, the 'Nhân vật' (Sprite) and 'Sân khấu' (Stage) panels are visible. The 'Nhân vật' panel has 'Cat' selected. The 'Sân khấu' panel shows 'x' at -125, 'y' at -18, 'Kích thước' at 100, and 'Hướng di chuyển' at 90. The 'Xuất hiện' section is also visible. The 'Hình nền' (Background) is set to 'không nền' (None) and 'Số nền' (Background count) is 3.

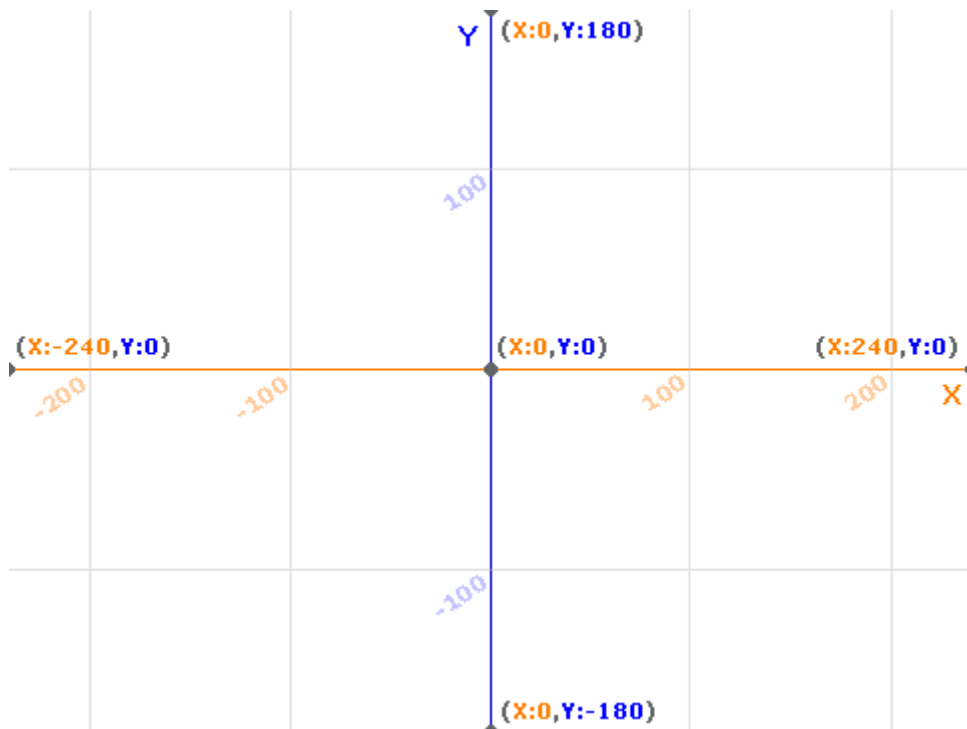
(1) Tên nhân vật.

(2) Hoành độ của nhân vật (so với trung tâm của sân khấu).

- (3) Tung độ của nhân vật (so với trung tâm của sân khấu).
- (4) Ẩn/Hiện nhân vật.
- (5) Kích thước nhân vật.
- (6) Hướng di chuyển (Góc của nhân vật so với trục hoành).
- (7) Nhân vật đang chọn.

### Khái niệm:

- Hệ tọa độ Oxy gồm 2 trục Ox và Oy.



- **Trục Ox** là trục nằm ngang, kéo dài từ bên trái sang bên phải tương ứng với giá trị x từ -240 (bên trái màn hình) đến 240 (bên phải màn hình). Tọa độ x (hay còn gọi là hoành độ) để biểu thị vị trí của nhân vật theo phương ngang.
- Tương tự như vậy, **trục Oy** là trục thẳng đứng, kéo dài từ dưới lên trên tương ứng với giá trị y từ -180 (phía dưới màn hình) đến 180 (phía trên màn hình). Tọa độ y (hay còn gọi là tung độ) để biểu thị vị trí của nhân vật theo phương thẳng đứng.
- Kết hợp tọa độ x và y, ta được một **bộ (x, y)** dùng để biểu diễn chính xác vị trí của nhân vật trên màn hình.
- Ngoài ra, **điểm O** sẽ là điểm nằm chính giữa sân khấu và có tọa độ (0,0).

### Ứng dụng

#### a. Tọa độ của nhân vật

- Để biết được tọa độ của một nhân vật, ta nhấn chọn nhân vật đó và nhìn ở phần thuộc tính nhân vật.

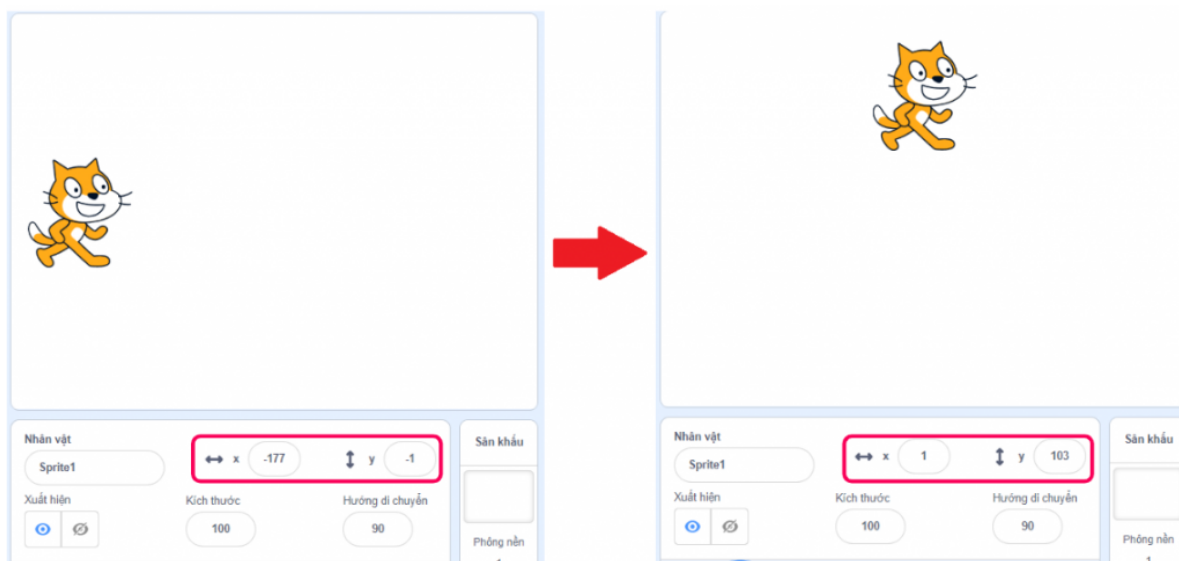
Ví dụ: Nhân vật mèo Miu dưới đây có tọa độ x bằng -125 còn tọa độ y bằng -18.

The image shows the Scratch software interface. At the top, a cat sprite is positioned on a stage. Below the stage, the 'Nhân vật' (Sprite) panel is visible. The 'x' coordinate is set to -125 and the 'y' coordinate is set to -18. The 'Kích thước' (Size) is set to 100. The 'Số khung nền' (Frame number) is set to 3. The 'Hoành độ x' and 'Tung độ y' labels are highlighted in yellow boxes, with arrows pointing to their respective input fields in the 'Nhân vật' panel.

### b. Tìm tọa độ của nhân vật ở vị trí bất kỳ

- Muốn biết tọa độ của nhân vật ở vị trí bất kỳ, bạn hãy nhấn chọn nhân vật và di chuyển nhân vật đến vị trí đó và thả chuột. Phần thuộc tính của nhân vật sẽ tự động cập nhật vị trí tọa độ (x;y) tương ứng.

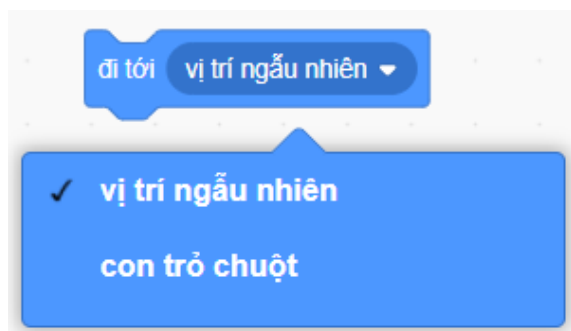
## Ví dụ:



- Vị trí ban đầu của nhân vật là (-177,-1) (hoành độ x có giá trị là -177, tung độ y có giá trị là -1). Sau khi di chuyển nhân vật đến vị trí mới thì tọa độ của nhân vật lúc đó là (1,103) (hoành độ x có giá trị là 1, tung độ y có giá trị là 103).
- Lưu ý: Khoảng giá trị của hoành độ x và hoành độ y tương ứng với màn hình của Scratch.

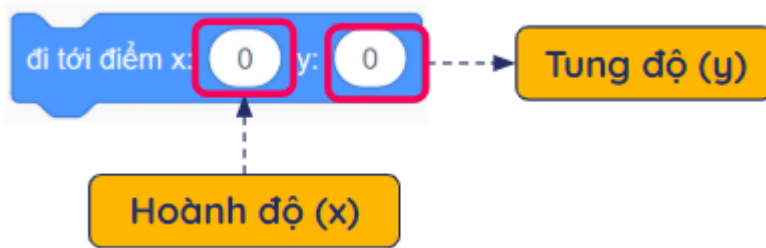
### 1.2.4. Bộ câu lệnh đi tới

#### 1.2.4.1. Câu lệnh đi tới



- Câu lệnh **đi tới** có tác dụng làm cho nhân vật lập tức di chuyển đến vị trí nào đó dựa vào lựa chọn của câu lệnh:
  - **vị trí ngẫu nhiên**: khi kích hoạt, nhân vật đi tới vị trí ngẫu nhiên.
  - **con trỏ chuột**: khi kích hoạt, nhân vật đi tới vị trí của con trỏ chuột.

### 1.2.4.2. Câu lệnh đi tới vị trí xác định



- Câu lệnh **đi tới điểm x: ... y: ...** có tác dụng làm cho nhân vật lập tức di chuyển đến vị trí tương ứng với hoành độ (x) và tung độ (y) đã xác định trên sân khấu.

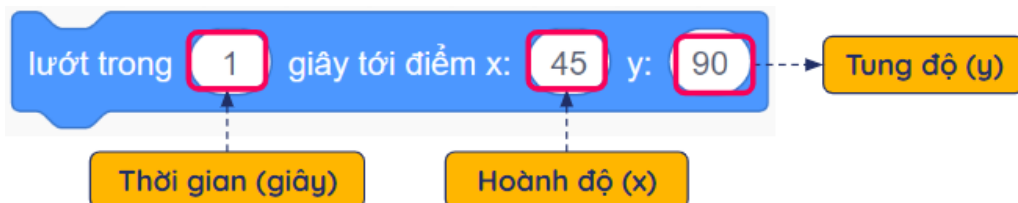
### 1.2.5. Bộ câu lệnh lướt

#### 1.2.5.1. Câu lệnh lướt trong ... giây tới ...



- Nếu như câu lệnh **đi tới** làm nhân vật **LẬP TỨC** nhảy đến vị trí nào đó, thì câu lệnh **lướt trong ... giây tới ...** làm cho nhân vật **DI CHUYỂN TỪ TỪ** đến một vị trí trong số giây được nhập vào.
- **Số giây** để nhân vật di chuyển đến nơi chỉ định được nhập trong ô số, số giây được **đặt mặc định** là **1 giây**.
- Khi kích hoạt câu lệnh:
  - **vị trí ngẫu nhiên:** nhân vật di chuyển đến vị trí bất kì.
  - **con trỏ chuột:** nhân vật di chuyển đến vị trí con trỏ chuột.

#### 1.2.5.2. Câu lệnh lướt xác định tọa độ

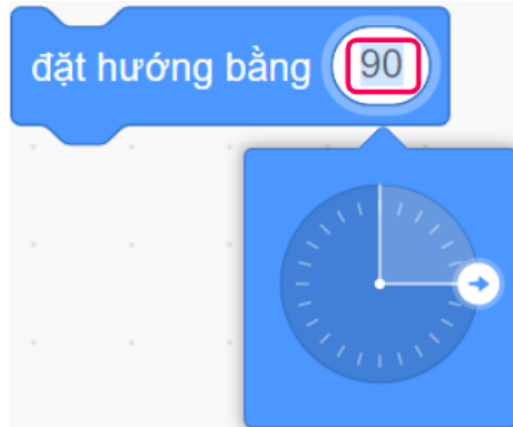


- Sau khi nhập số giây và tọa độ (hoành độ x, tung độ y) của vị trí mà bạn mong muốn và thực hiện câu lệnh này, nhân vật sẽ di chuyển đến vị trí có tọa độ (hoành độ x, tung độ y) tương ứng trên phong nền trong thời gian mà bạn nhập.

## 1.2.6. Bộ câu lệnh hướng

- Hướng của nhân vật được đo bằng góc giữa phía mà nhân vật đó đang hướng về và trục hoành (trục Ox).

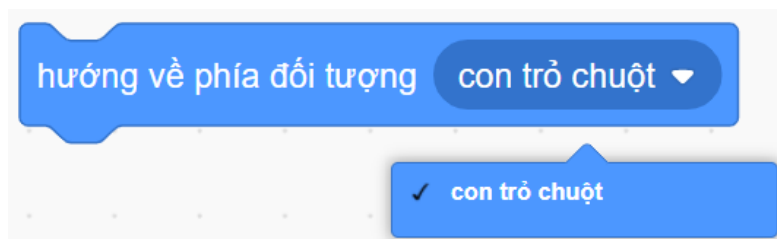
### 1.2.6.1. Câu lệnh đặt hướng bằng



- Câu lệnh **đặt hướng bằng ...** có tác dụng làm cho nhân vật quay về hướng thầy/cô mong muốn bằng cách nhập số vào ô.
- Thầy/Cô có thể nhập trực tiếp số hoặc quay mũi tên về hướng mình muốn.



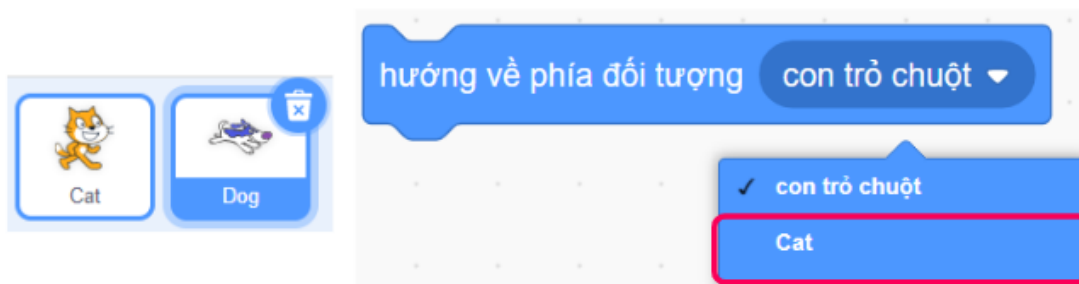
### 1.2.6.2. Câu lệnh hướng về phía đối tượng



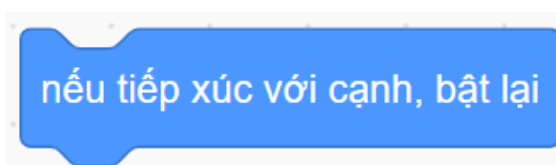
- Khi chạy câu lệnh này, nhân vật sẽ **hướng về phía vị trí của con trỏ chuột** trên phong nền.



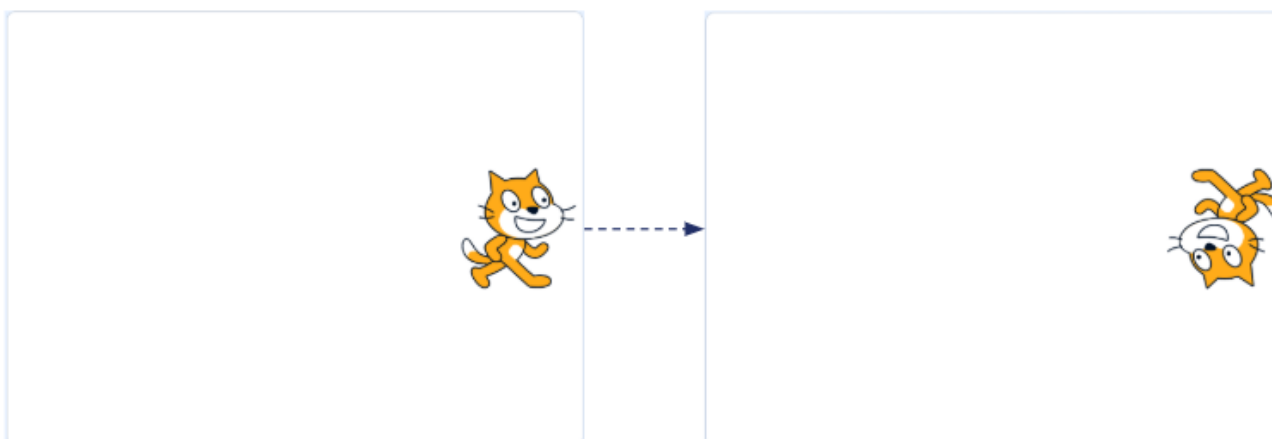
- Trong trường hợp dự án (Project) có nhiều hơn 1 nhân vật (sprite), ta có thể lập trình để nhân vật này hướng về nhân vật khác.
- Ví dụ trong hình dưới đây, có 2 nhân vật: Cat (mèo) và Dog (chó). Để lập trình cho Dog hướng về phía Cat, ta chọn câu lệnh *hướng về phía đối tượng Cat*.



### 1.2.7. Câu lệnh bật lại khi tiếp xúc với cạnh



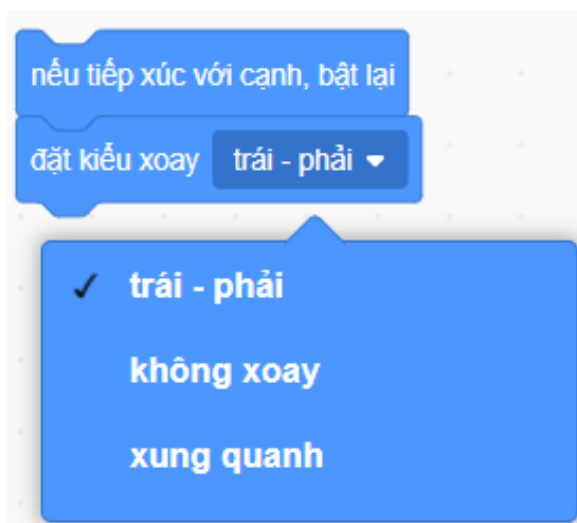
- Câu lệnh này là một câu điều kiện, nó sẽ được thực thi khi nhân vật của thầy/cô di chuyển đến cạnh của màn hình sân khấu.
- Cụ thể, khi nhân vật của thầy/cô di chuyển và chạm vào cạnh màn hình, nhân vật sẽ bị bật ngược ra, như ví dụ bên dưới.



### 1.2.8. Câu lệnh đặt kiểu xoay

- Khi nhân vật của thầy/cô di chuyển và chạm vào cạnh của màn hình, sử dụng câu lệnh **nếu tiếp xúc với cạnh, bật lại** sẽ làm nó bật ngược lại, nhưng với hướng ngược lại như thầy/cô thấy trong ví dụ trên. Đây là lúc thầy/cô nên sử dụng câu lệnh **đặt kiểu xoay**.
- Câu lệnh đặt kiểu xoay có 3 lựa chọn:
  - Trái-phải: có nghĩa là hình ảnh của nhân vật chỉ quay sang trái và quay sang phải
  - Không xoay: có nghĩa là hình ảnh của nhân vật luôn hướng mặt sang một phía dù thay đổi thông số Hướng di chuyển trong phần Thuộc tính của nhân vật

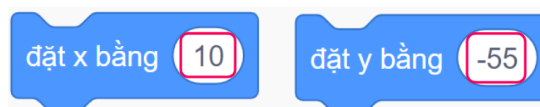
- Xung quanh: có nghĩa là hình ảnh nhân vật xoay 360 độ
- Khi sử dụng câu lệnh này, nhân vật của mình sẽ bật ra nhưng theo hướng mà thầy/cô chọn, chứ không chỉ bật ngược như ví dụ trên



- Thầy/Cô nên ghép câu lệnh **đặt kiểu xoay** với câu lệnh **nếu tiếp xúc với cạnh, bật lại** như thế này.
- Lưu ý: Câu lệnh **đặt kiểu xoay** chỉ ảnh hưởng đến hướng của hình ảnh nhân vật. Nó không thể hiện hướng chuyển động của nhân vật.

## 1.2.9. Thao tác trực tiếp lên tọa độ

### 1.2.9.1. Câu lệnh đặt trực tiếp tọa độ



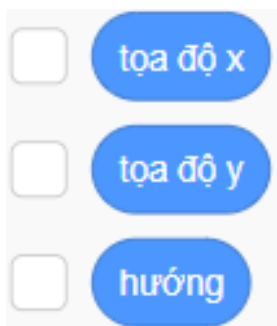
- Có 2 câu lệnh đặt tọa độ, áp dụng cho hoành độ x và tung độ y
- Sau khi nhập tọa độ mình mong muốn và chạy câu lệnh, nhân vật của thầy/cô sẽ được đưa đến vị trí có tọa độ tương ứng.

### 1.2.9.2. Câu lệnh thay đổi tọa độ



- Có 2 câu lệnh thay đổi tọa độ (hoành độ x, tung độ y)
- Câu lệnh này có tác dụng tăng/giảm tọa độ x hoặc y lên số đơn vị mà thầy/cô nhập.
- **Ví dụ:** Trong hình trên:
  - Câu lệnh **thay đổi x một lượng 10** có tác dụng tăng tọa độ x hiện tại lên 10 đơn vị.
  - Câu lệnh **thay đổi y một lượng -10** có tác dụng giảm tọa độ y hiện tại xuống 10 đơn vị.

### 1.2.9.3. Hiển thị vị trí lên màn hình

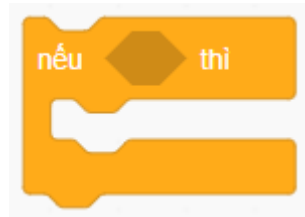


- Các mục có thể tích như trên để hiển thị tọa độ (hoành độ x, tung độ y) và hướng của nhân vật lên màn hình.
- **Ví dụ:** Khi thầy/cô tích chọn các thông số, trên màn hình sẽ hiện

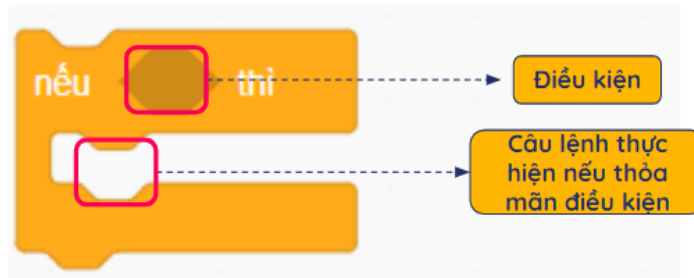


## 2. Câu lệnh Nếu... Thì...

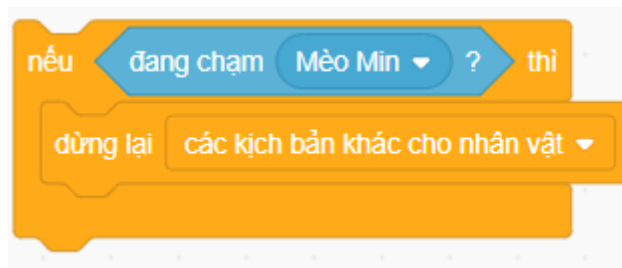
- Một trong câu lệnh cơ bản của tất cả các ngôn ngữ lập trình, bao gồm cả Scratch, chính là câu lệnh **Nếu ... thì ...**



- **Câu lệnh Nếu** được gọi là **câu lệnh điều kiện**. Câu lệnh này dùng để kiểm tra một điều kiện nào đó, **nếu thỏa mãn điều kiện thì thực hiện câu lệnh bên trong nó**.

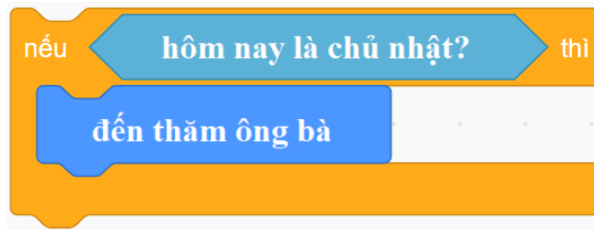


### Ví dụ:



- Trong trường hợp này, **nếu nhân vật đang chạm Mèo Min** thì chương trình sẽ dừng lại các kịch bản khác cho nhân vật.
- Ở đây **“Dừng lại các kịch bản khác cho nhân vật”** được hiểu đơn giản là các nhân vật khác sẽ dừng ở trạng thái hiện tại và không thực hiện các hành động sau nữa.
- **Nếu điều kiện sai** (đang không chạm Mèo Min) thì sẽ không làm gì cả.

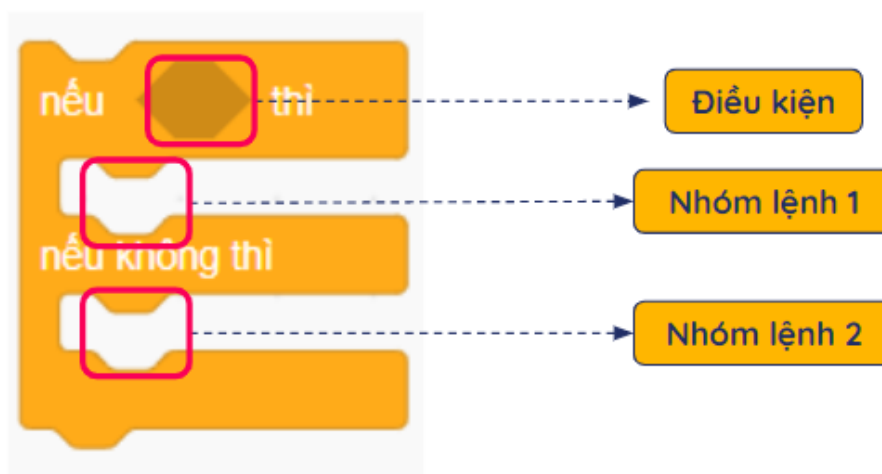
### Một ví dụ khác đơn giản hơn:



- Chương trình sẽ kiểm tra “Nếu hôm nay là chủ nhật? Thì đến thăm ông bà”. Trong trường hợp hôm nay không phải là chủ nhật, thì không làm gì.

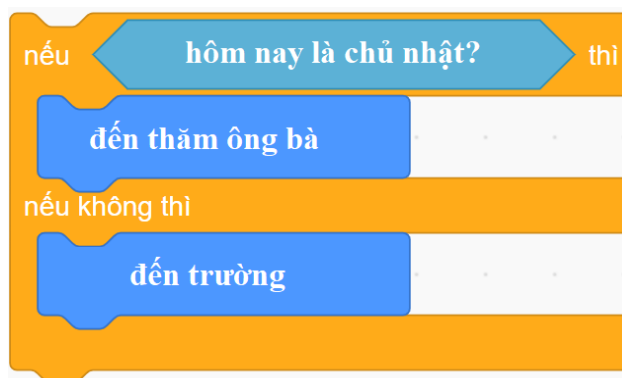
### MỞ RỘNG:

- Ngoài ra còn có câu lệnh **Nếu ... thì ... Nếu không ... thì ...**



- Câu lệnh này **gần giống** với câu lệnh **Nếu ... thì ...** nguyên bản, nhưng điểm đặc biệt ở chỗ: Sau khi kiểm tra điều kiện, nếu **thỏa mãn đúng điều kiện thì thực hiện nhóm lệnh 1**, nếu **không thỏa mãn điều kiện thì thực hiện nhóm lệnh 2**.

### Ví dụ:



- Chương trình sẽ kiểm tra “**Nếu hôm nay là chủ nhật? Thì đến thăm ông bà**”. Trong trường hợp hôm nay không phải là chủ nhật thì sẽ **đến trường**.

### 3. Câu lệnh “Đang chạm ...?” và “Phím .... được bấm?”

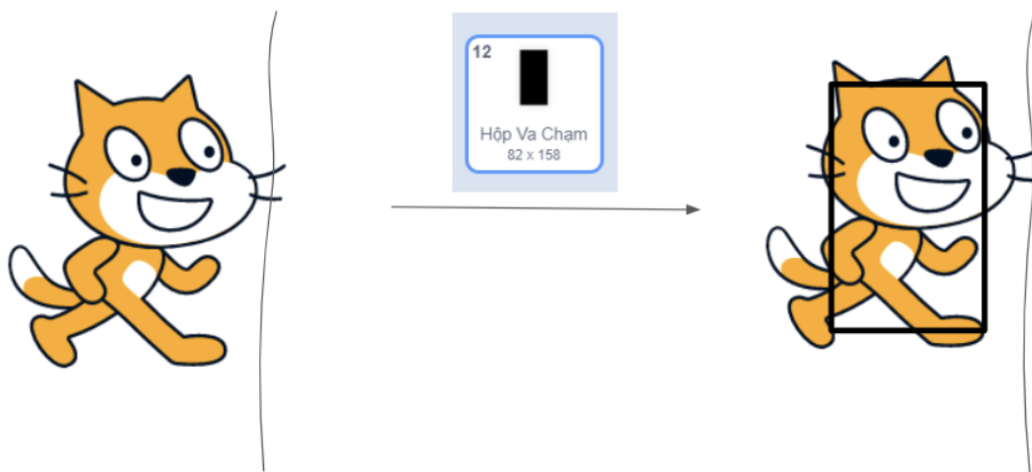
#### 3.1. Câu lệnh “Đang chạm ...?”

- Một trong những câu lệnh đầu tiên trong khối “cảm biến” là câu lệnh “đang chạm....?”. Câu lệnh này kiểm tra xem nhân vật của nó có chạm vào con trỏ chuột, cạnh hoặc một nhân vật khác hay không (ví dụ như nhân vật “winter hat” trong hình minh họa). **Nếu nhân vật hiện tại có đang chạm vào đối tượng đã chọn, khối sẽ trả về “đúng”; nếu không, nó trả về “sai”.**



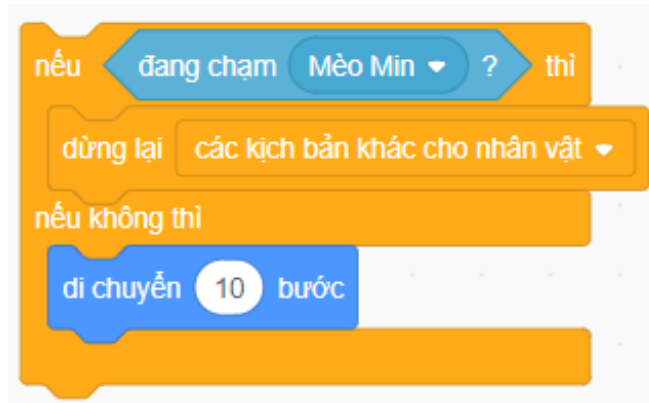
- Việc một nhân vật có chạm vào người khác hay không phụ thuộc vào việc một phần kết cấu hình ảnh của nhân vật đó (được xác định bởi “**Trang phục**” của nhân vật ấy) có chạm vào người khác hay không.

**Ví dụ:** nếu “Trang phục” của một nhân vật có hình vuông, nó sẽ có một “**hộp va chạm**” hình vuông khi phát hiện nếu nó chạm vào các nhân vật khác.



- Trong hình trên, Miu phía bên trái không có hộp va chạm thì Miu sẽ dừng lại khi râu của Miu chạm vào tường. Khi đó, tuy chỉ có râu của Miu chạm vào tường nhưng trò chơi sẽ tính là cả nhân vật chạm vào tường. Như vậy người chơi sẽ có thể cảm thấy trò chơi chưa đủ chân thật và sống động.
- Ngược lại, Miu ở phía bên phải sử dụng “hộp va chạm”, Miu sẽ không dừng lại khi râu của mình chạm vào tường nữa. Miu sẽ chỉ dừng lại và bị tính là chạm tường khi “hộp va chạm” chạm vào tường.

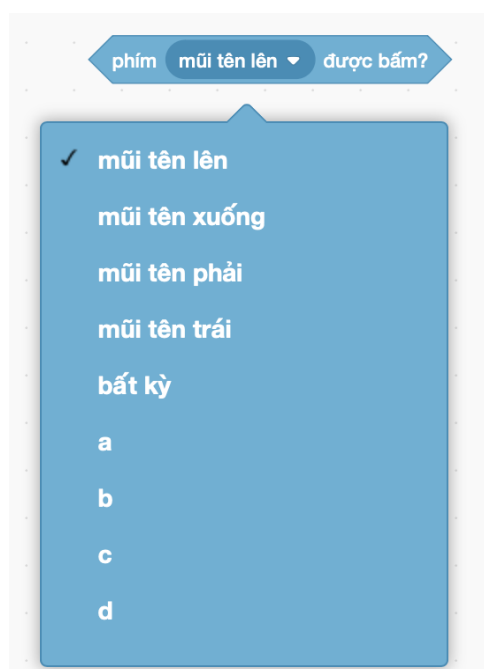
**Ví dụ:** Kết hợp “Đang chạm...?” cùng với câu lệnh “Nếu ... thì ...” và “Nếu không ... thì”



- Ở ví dụ này, nếu nhân vật đang chạm Mèo Min thì sẽ thực hiện dừng lại các kịch bản khác cho nhân vật. Trong trường hợp ngược lại (nếu đang không chạm Mèo Min) thì nhân vật sẽ di chuyển 10 bước.

### 3.2. Câu lệnh “Phím ... được bấm?”

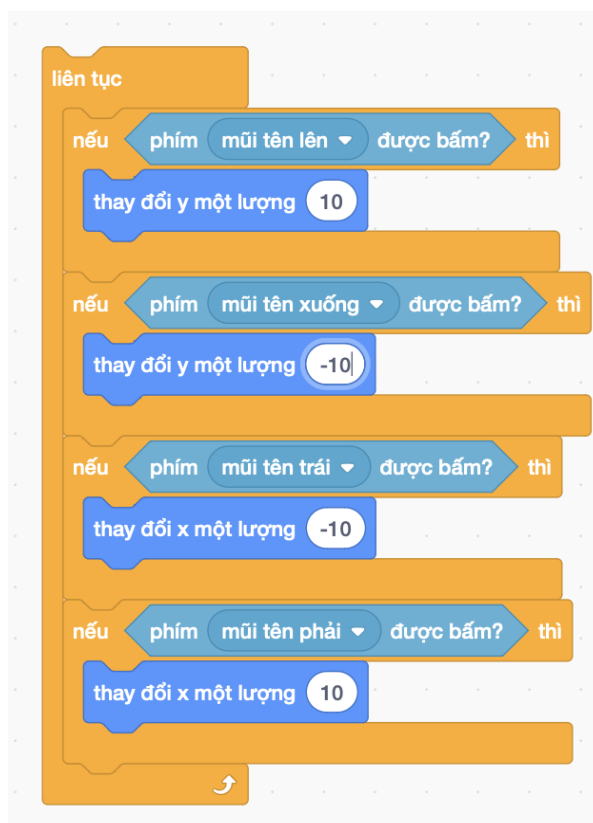
- Câu lệnh “Phím ... được bấm?” là thuộc khối “Cảm biến”, câu lệnh này kiểm tra xem phím được chỉ định có được nhấn hay không. Nếu phím đang được nhấn, khối trả về “đúng”; nếu không, nó trả về “sai”.
- Các phím có sẵn để sử dụng trong cùng với câu lệnh này bao gồm **toàn bộ bảng chữ cái tiếng Anh (a, b, c, v.v.), các phím số (0, 1, 2, v.v.), các phím mũi tên (← ↑ → ↓) và phím cách**. Câu lệnh này còn có thể cho phép **chọn một phím bất kỳ**, cho phép một người nhấn bất kỳ phím nào để vận hành khối.



- Ngoài ra, câu lệnh **“Phím ... được bấm?”** có thể được sử dụng để thay thế cho câu lệnh **“Khi bấm phím ...”** vốn thuộc khối lệnh **“Sự kiện”**.



**Ví dụ:** một chức năng hay được dùng của câu lệnh **“Phím ... được bấm?”** là dùng để điều khiển chuyển động của nhân vật một cách mượt mà.



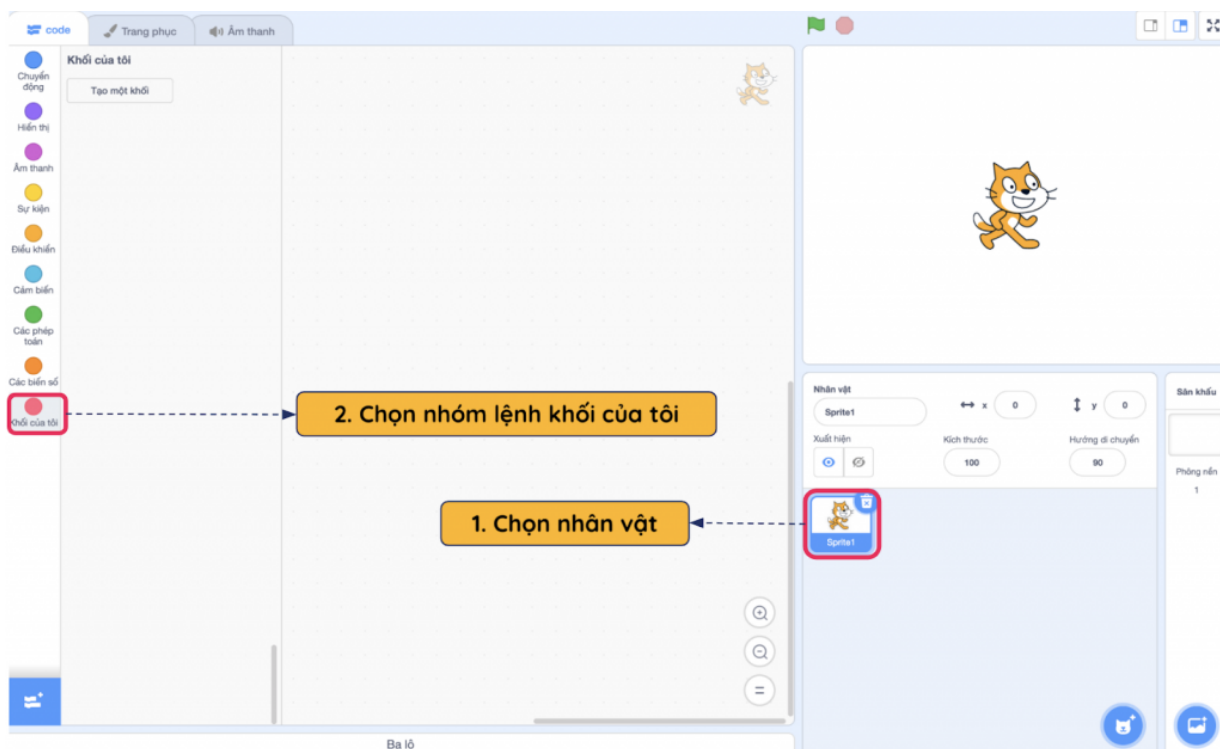
- Trong ví dụ này, câu lệnh **“Phím ... được bấm?”** được kết hợp với câu lệnh **“Nếu ... thì ...”** và **vòng lặp liên tục**. Ý nghĩa của chương trình này là **liên tục** đợi chúng ta lệnh nhấn **phím mũi tên (← ↑ → ↓)** và thực hiện các lệnh làm thay đổi hoành độ x hoặc tung độ y của nhân vật một lượng + 10 (hoặc -10) đơn vị.
- Kết quả là nhân vật của chúng ta sẽ có hiệu ứng di chuyển sang các hướng phải, trái, lên, xuống tùy vào phím bấm phù hợp được chọn lựa.



## 4. Khối của tôi

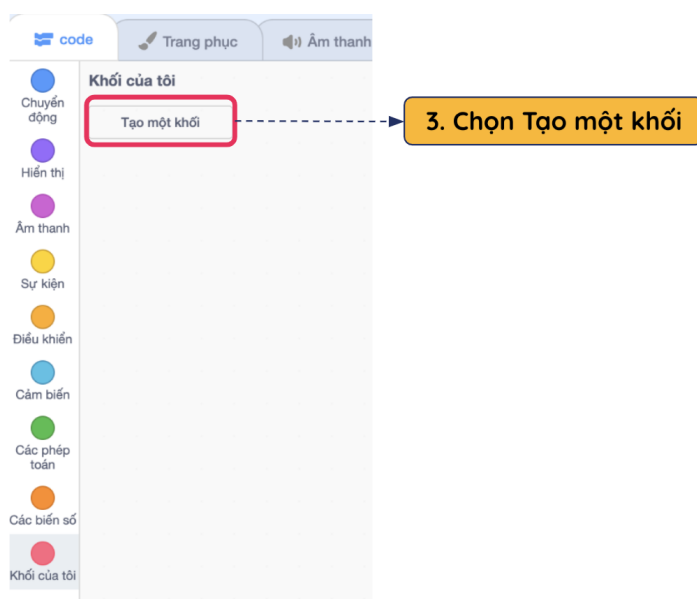
### 4.1. Cách tạo Khối của tôi

**Bước 1: Chọn nhân vật bạn muốn tạo khối của tôi.**



- **Lưu ý:** Khi tạo khối của tôi cho nhân vật nào thì chỉ nhân vật đó mới có thể sử dụng khối lệnh này.

**Bước 2: Chọn nhóm lệnh Khối của tôi**



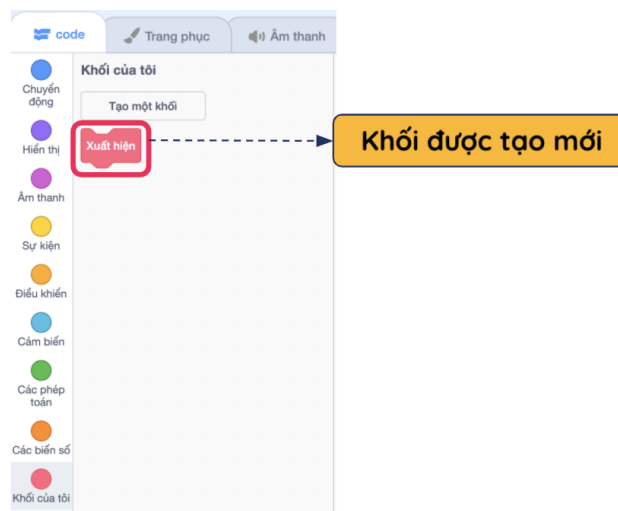
**Bước 3: Chọn Tạo một khối. Sau đó màn hình sẽ hiển thị hộp thoại**



#### Bước 4: Điền tên khối rồi nhấn nút OK

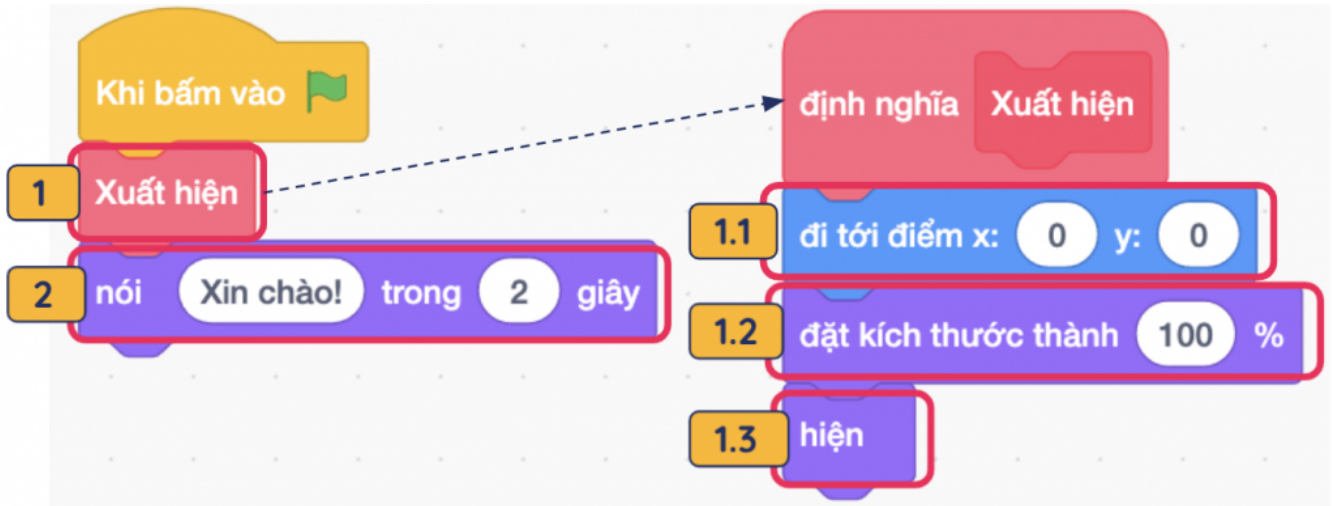
- Sau khi **nhấn OK**, một khối mới của tôi sẽ được tạo ra và xếp ở dưới như hình vẽ.

#### Kết quả:



#### 4.2. Cách sử dụng Khối của tôi

- **Khối của tôi** là một khối lệnh mới mà chúng ta tự tạo ra. Khối lệnh này được định nghĩa dựa trên những khối lệnh có sẵn trong Scratch.
- **Ví dụ:** khối lệnh Xuất hiện được định nghĩa là khi nhận vật sẽ xuất hiện ở chính giữa màn hình, có kích thước chính xác khi được tạo ra. Khi khối lệnh này được sử dụng ở chương trình chính (nổi vào bên dưới khối lệnh sự kiện như Khi bấm vào lá cờ), chương trình sẽ chạy lần lượt tất cả các khối lệnh ở phần định nghĩa của khối lệnh này.



- Khi ấn vào biểu tượng lá cờ, khối xuất hiện sẽ được gọi. Khi đó toàn bộ các khối lệnh bên trong khối xuất hiện sẽ chạy cho đến hết. Sau đấy mới quay trở về chạy tiếp khối nói “xin chào!” bên dưới khối xuất hiện. Thứ tự chạy cụ thể là: đi tới điểm chính giữa màn hình → đặt kích thước → hiện → nói xin chào.
- **Lưu ý:** Khối lệnh của tôi chỉ chạy khi được gọi đến, nên thầy/cô nhớ sử dụng khối lệnh của tôi sau khi tạo. Nếu chỉ có phần định nghĩa khối của tôi thì khối lệnh này sẽ không được chạy
- Đây là đoạn chương trình giống như phần dùng khối lệnh xuất hiện ở trên, các khối lệnh cũng được chạy theo thứ tự cụ thể là: đi tới điểm chính giữa màn hình → đặt kích thước → hiện → nói xin chào.



- Thầy/Cô có thể thấy phần chương trình dùng khối lệnh xuất hiện nhìn rõ ràng và dễ hiểu hơn khi gọi chung các khối hành động. Đặc biệt khi tìm sắp xếp theo các khối của tôi, việc tìm bug sẽ được nhanh chóng và hiệu quả hơn.