



Dự án cột mốc 2

PHẦN



Hoàn thành chương trình đào tạo Python

- Bây giờ bạn đã tìm hiểu về OOP, hãy cùng kiểm tra kỹ năng mới của bạn với một Dự án quan trọng khác.
- Để khởi động, trước tiên chúng tôi sẽ hướng dẫn bạn cách tạo một trò chơi bài đơn giản, sau đó bạn sẽ thử Bài tập dự án cột mốc thứ 2.



DỰ ÁN KHỞI ĐỘNG



Hoàn thành chương trình đào tạo Python

- Để khởi động cho Dự án cột mốc thứ 2, chúng tôi sẽ tạo lại trò chơi bài mang tên “Chiến tranh”.
- Chúng ta hãy có một cái nhìn tổng quan nhanh về trò chơi.



Hoàn thành chương trình đào tạo Python

- Để khởi động cho Dự án cột mốc thứ 2, chúng tôi sẽ tạo lại trò chơi bài mang tên “Chiến tranh”.
- Chúng ta hãy có một cái nhìn tổng quan nhanh về trò chơi.
 - [wikipedia.org/wiki/War_\(card_game\)](https://en.wikipedia.org/wiki/War_(card_game))

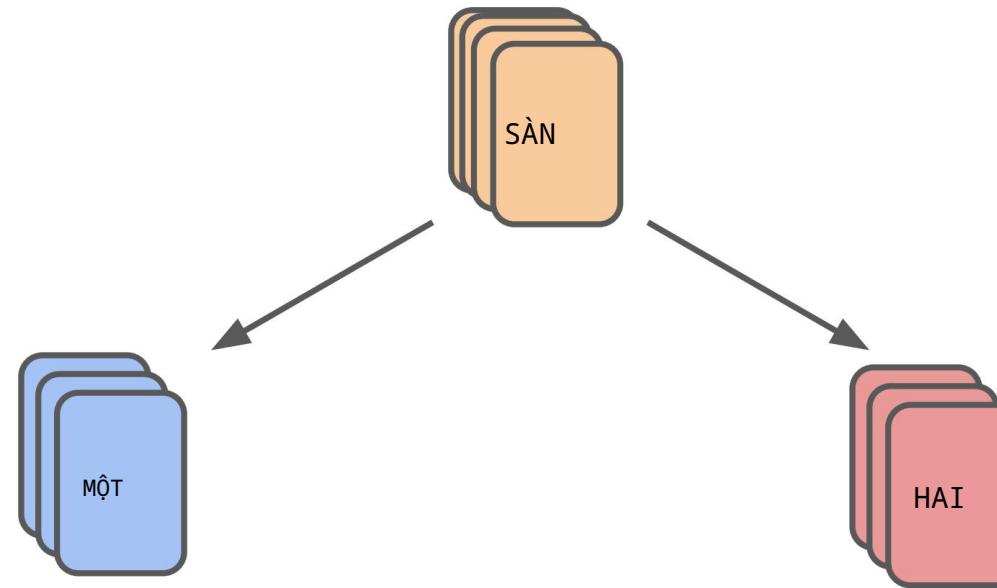


Hoàn thành chương trình đào tạo Python



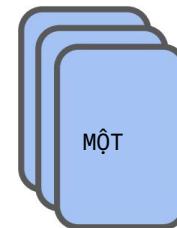


Hoàn thành chương trình đào tạo Python



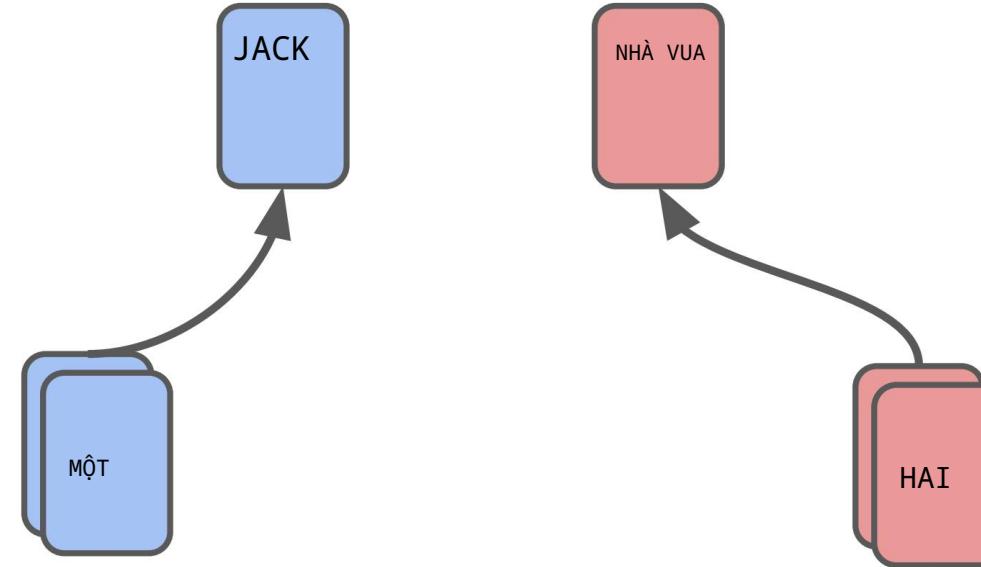


Hoàn thành chương trình đào tạo Python





Hoàn thành chương trình đào tạo Python





Hoàn thành chương trình đào tạo Python

JACK

NHÀ VUA

MỘT

HAI



Hoàn thành chương trình đào tạo Python

JACK



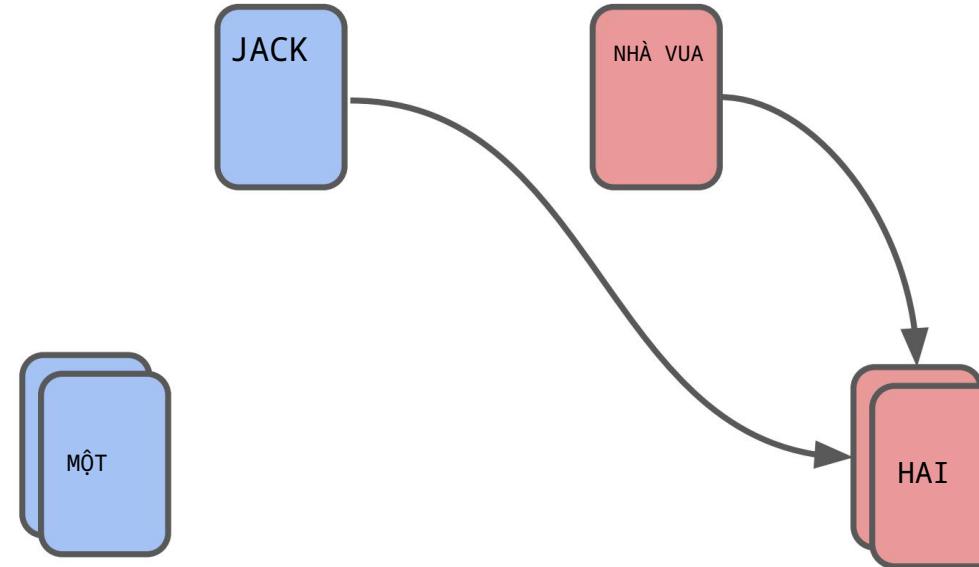
NHÀ VUA

MỘT

HAI

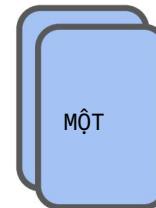


Hoàn thành chương trình đào tạo Python



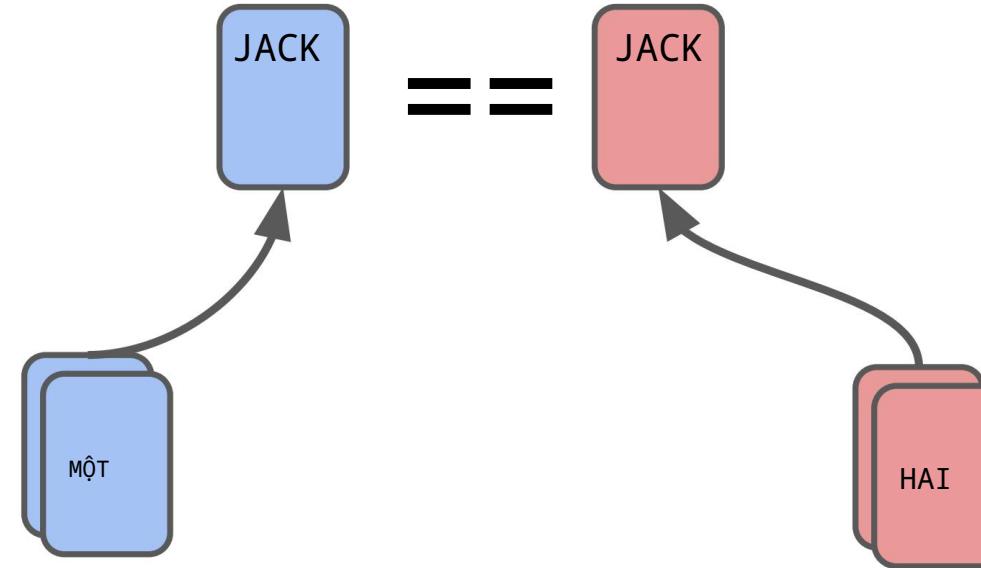


Hoàn thành chương trình đào tạo Python



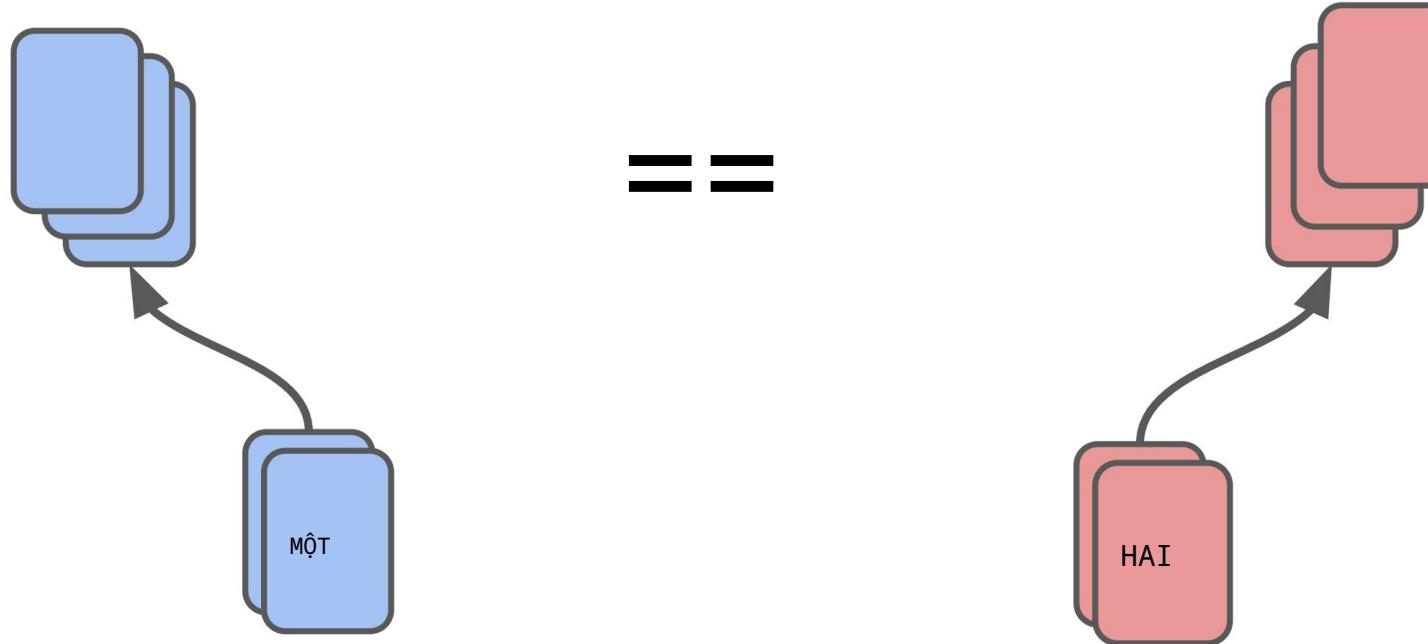


Hoàn thành chương trình đào tạo Python



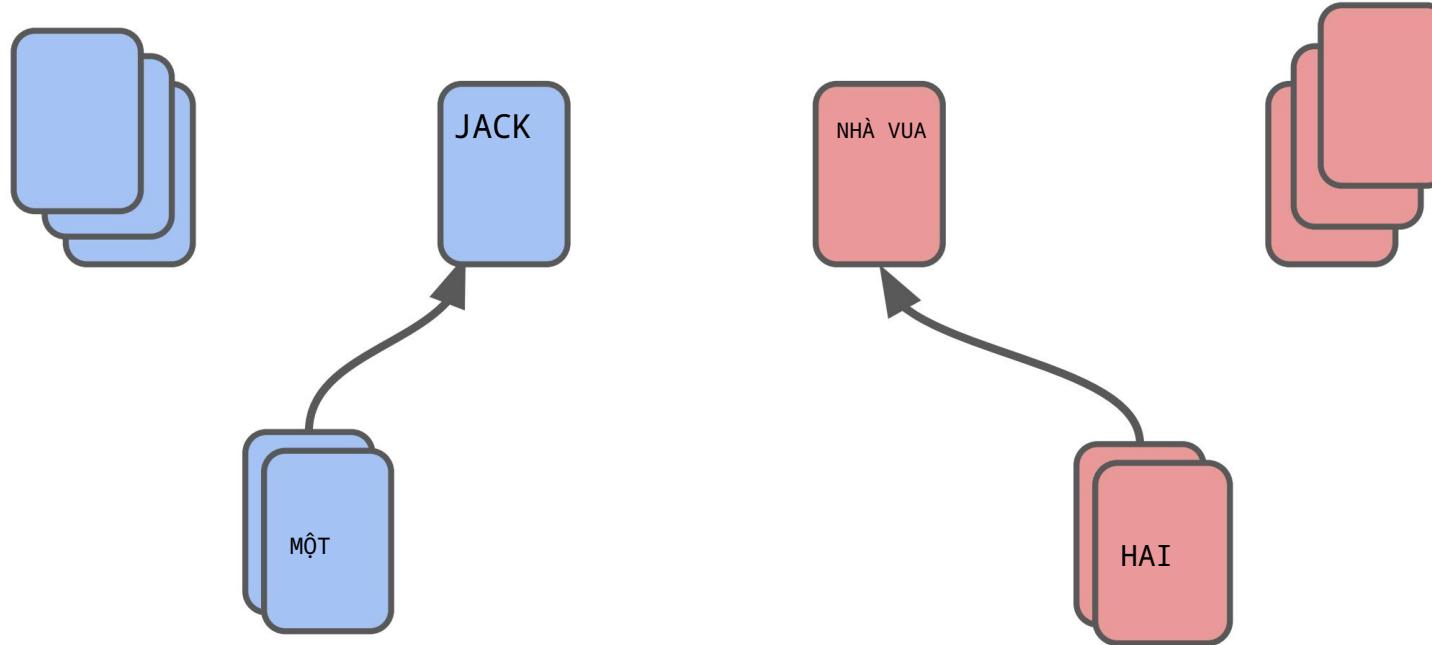


Hoàn thành chương trình đào tạo Python



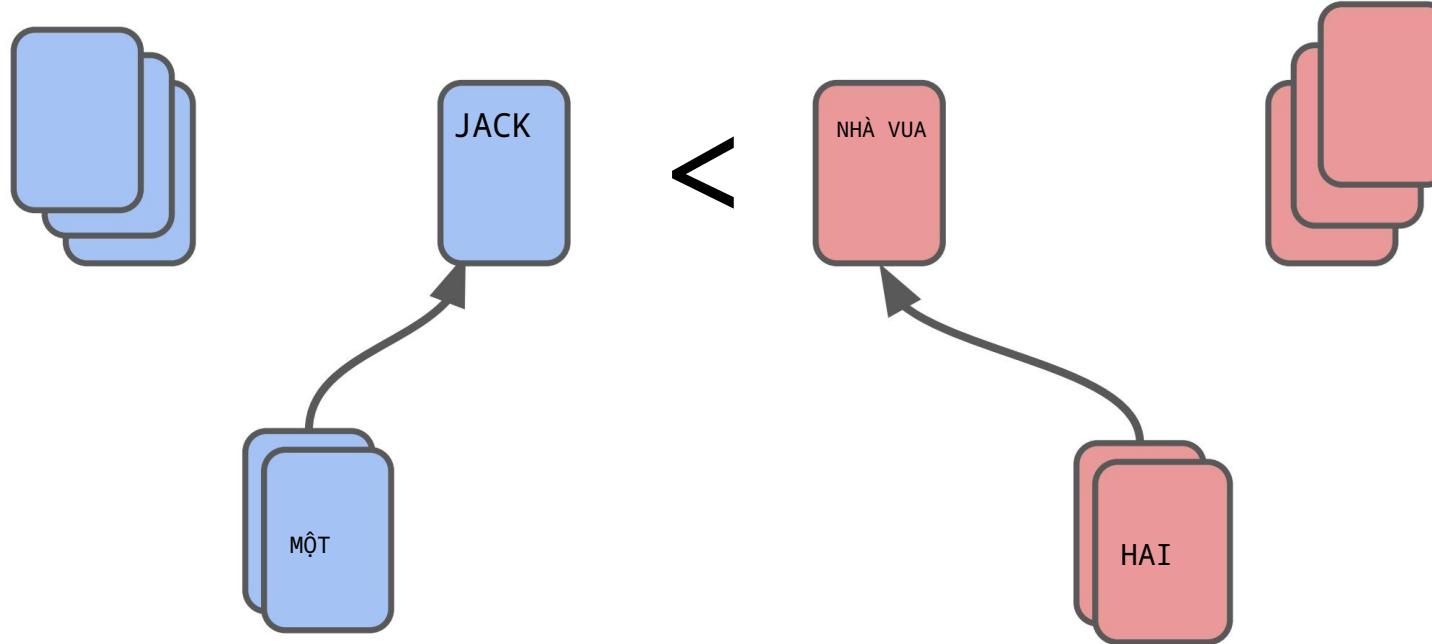


Hoàn thành chương trình đào tạo Python



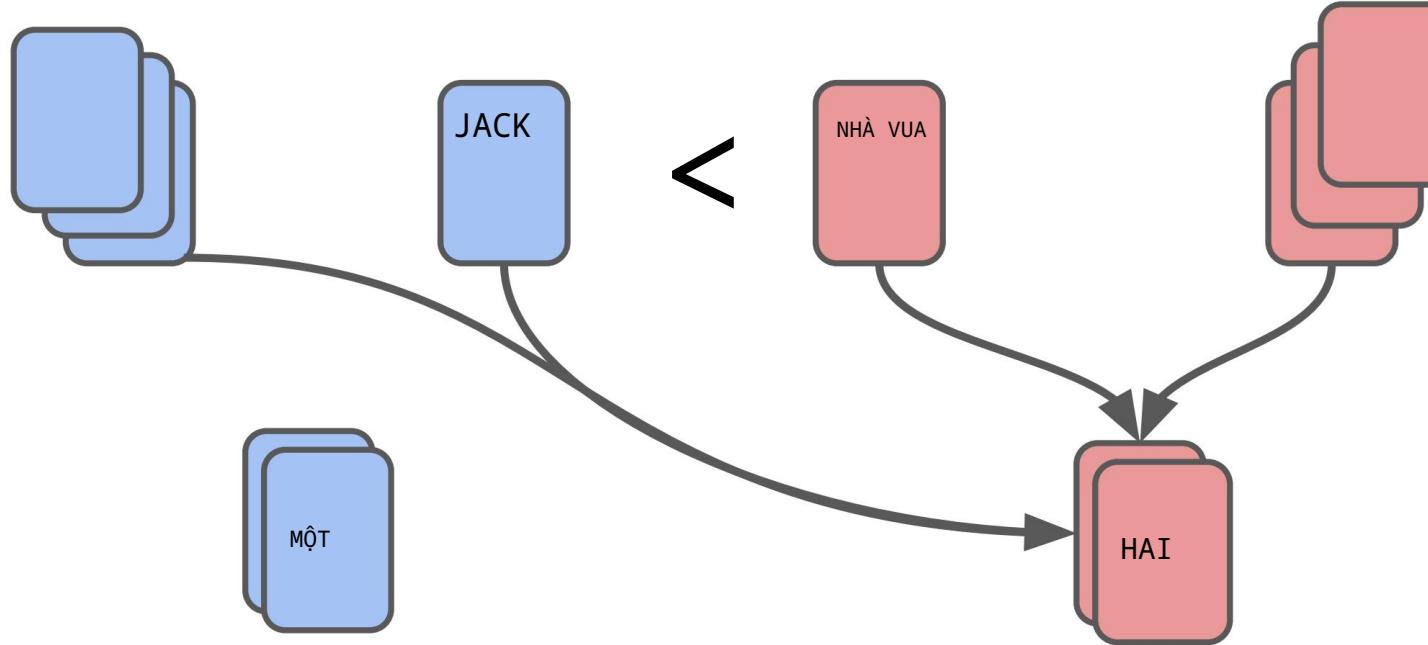


Hoàn thành chương trình đào tạo Python



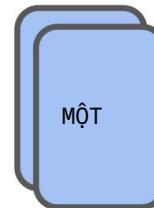


Hoàn thành chương trình đào tạo Python

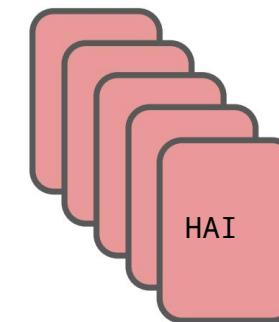




Hoàn thành chương trình đào tạo Python



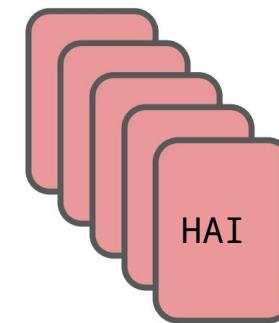
MỘT



HAI



Hoàn thành chương trình đào tạo Python





Hoàn thành chương trình đào tạo Python

- Quá trình “chiến tranh” có thể được lặp lại trong trường hợp quan hệ qua lại này.
- Để xây dựng trò chơi này, chúng ta sẽ tạo:
 - Loại thẻ •
 - Loại bộ bài •
 - Loại người chơi
 - Logic trò chơi



Bắt đầu nào !



Hạng thẻ



Lớp boong



Hoàn thành chương trình đào tạo Python

- Lớp boong
 - Tạo một bộ bài mới
 - Tạo tất cả 52 đối tượng Thẻ ■
Giữ dưới dạng danh sách các đối tượng Thẻ
 - Xáo trộn Bộ bài thông qua lệnh gọi phương thức ■ Hàm shuffle() thư viện ngẫu nhiên
 - Chia bài từ đối tượng Bộ bài ■ Phương thức lấy bài từ danh sách bài



Hoàn thành chương trình đào tạo Python

- Lớp boong
 - Chúng ta sẽ thấy lớp Deck chứa một danh sách các đối tượng Card.
 - Điều này có nghĩa là lớp Deck sẽ trả về các thẻ hiện đối tượng của lớp Card chứ không chỉ các kiểu dữ liệu python thông thường.
 - Hãy bắt đầu!



Lớp người chơi



Hoàn thành chương trình đào tạo Python

- Lớp người chơi
 - Lớp này sẽ được sử dụng để giữ người chơi danh sách thẻ hiện tại.
 - Người chơi có thể thêm hoặc xóa các lá bài khỏi “tay” của họ (danh sách các đối tượng Thẻ).



Hoàn thành chương trình đào tạo Python

- Lớp người chơi
 - Chúng tôi muốn người chơi có thể thêm một thẻ hoặc nhiều thẻ vào danh sách của họ, vì vậy chúng tôi cũng sẽ khám phá cách thực hiện điều này trong một lệnh gọi phương thức.



Hoàn thành chương trình đào tạo Python

- Lớp người chơi
 - Điều cuối cùng chúng ta cần nghĩ tới là dịch Bộ bài/Bộ bài có phần trên và phần dưới sang danh sách Python.
 - Hãy thử hình dung điều này.



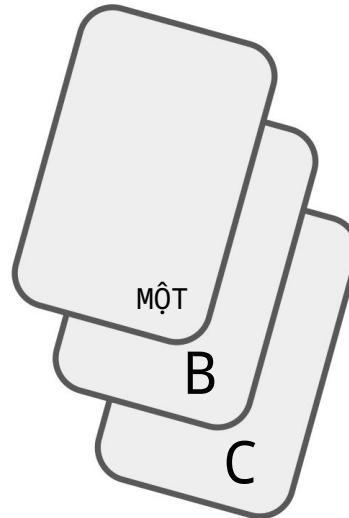
Hoàn thành chương trình đào tạo Python

- Lớp người chơi sẽ có danh sách `self.all_cards`



Hoàn thành chương trình đào tạo Python

- Lớp người chơi sẽ có danh sách self.all_cards

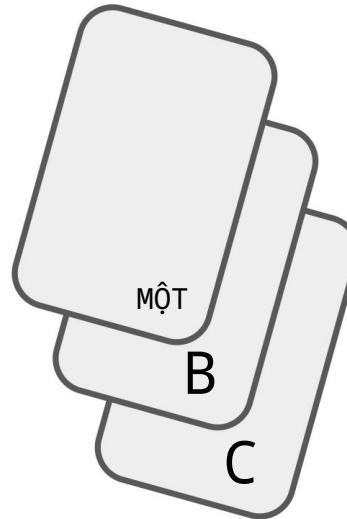


```
thẻ = ["A", "B", "C"]
```



Hoàn thành chương trình đào tạo Python

- Lớp người chơi sẽ có danh sách self.all_cards

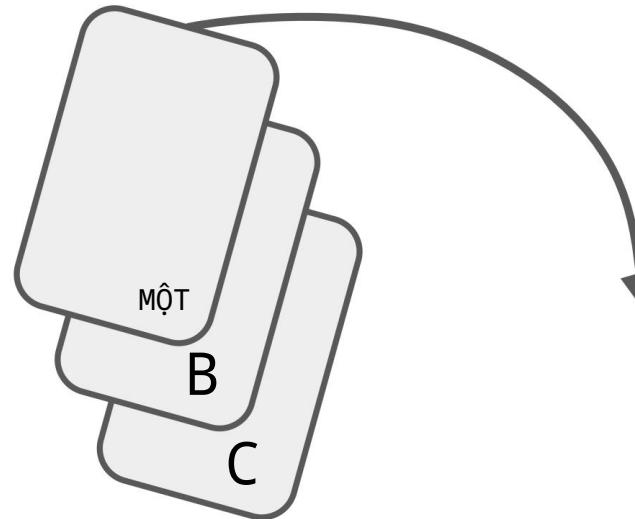


[“A”, “B”, “C”]



Hoàn thành chương trình đào tạo Python

- Người chơi “đánh” một lá bài từ trên xuống

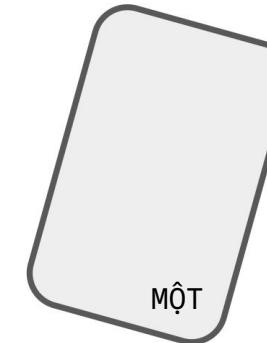
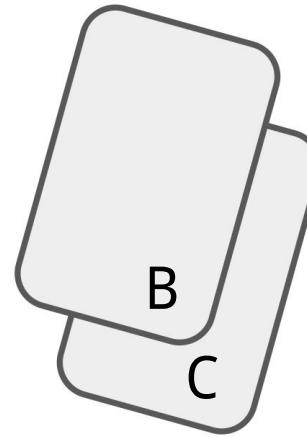


[“A”, “B”, “C”]



Hoàn thành chương trình đào tạo Python

- Người chơi “đánh” một lá bài từ trên xuống

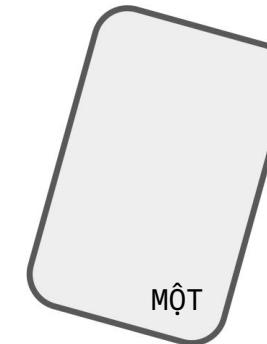
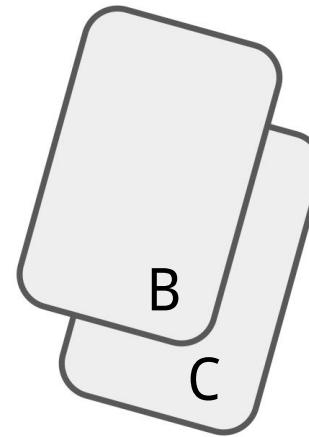


["A" , "B" , "C"]



Hoàn thành chương trình đào tạo Python

- Người chơi “đánh” một lá bài từ trên xuống



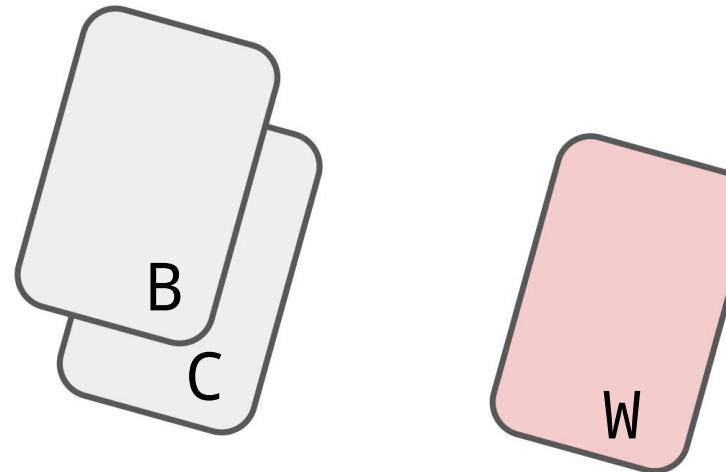
[“B” , “C”]

thẻ.pop(0)



Hoàn thành chương trình đào tạo Python

- Người chơi sẽ thêm thẻ vào “dưới cùng”

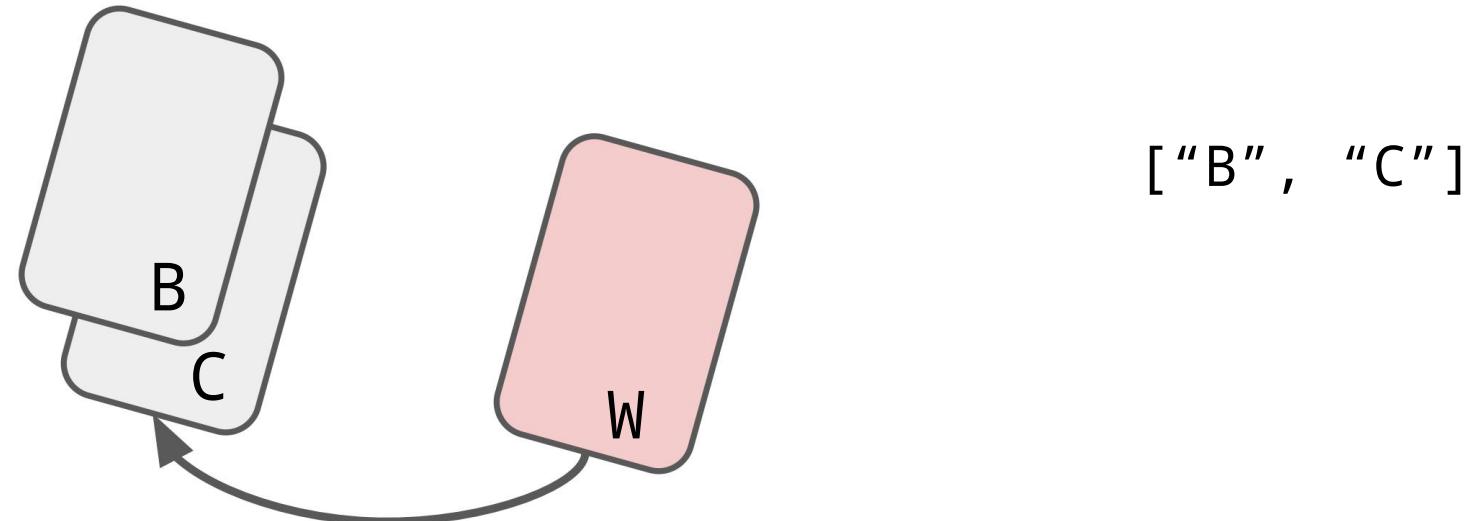


["B" , "C"]



Hoàn thành chương trình đào tạo Python

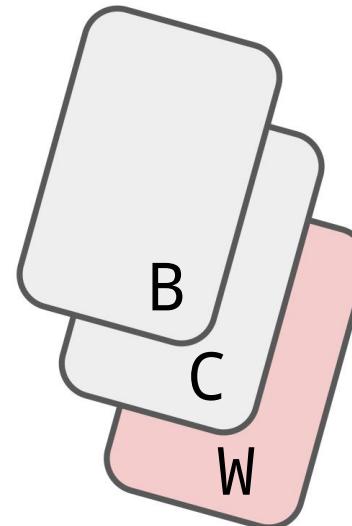
- Người chơi sẽ thêm thẻ vào “dưới cùng”





Hoàn thành chương trình đào tạo Python

- Người chơi sẽ thêm thẻ vào “dưới cùng”



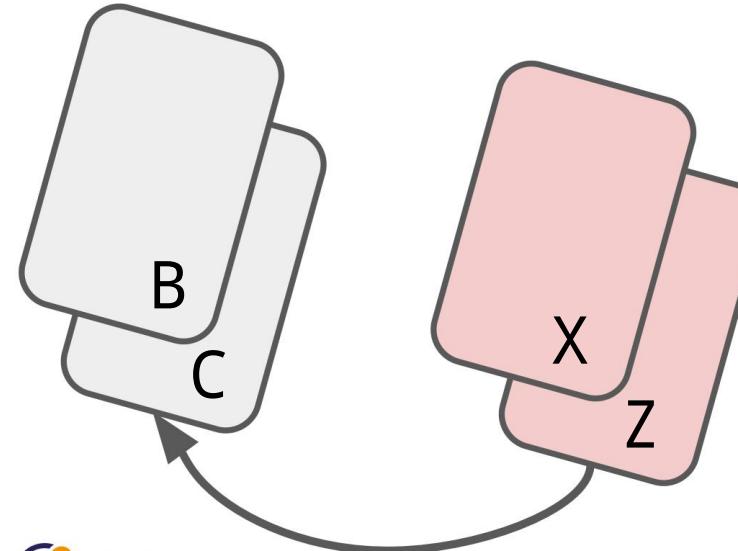
["B" , "C" , "W"]

`cards.append("W")`



Hoàn thành chương trình đào tạo Python

- Người chơi thêm nhiều thẻ sẽ sử dụng phần mở rộng()



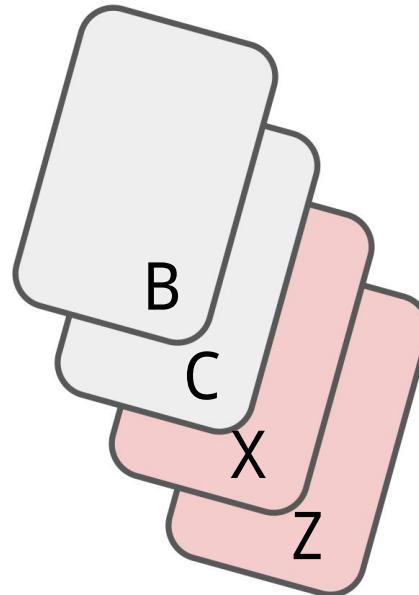
```
thẻ = [ "B", "C" ]
```

```
mới = [ "X", "Z" ]
```



Hoàn thành chương trình đào tạo Python

- Người chơi thêm nhiều thẻ sẽ sử dụng phần mở rộng()



```
thẻ = [ "B", "C" ]
```

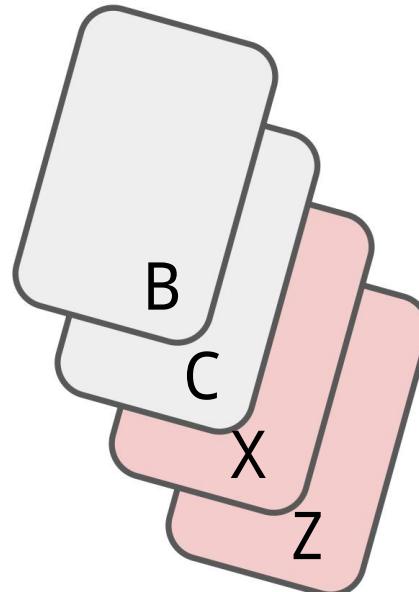
```
mới = [ "X", "Z" ]
```

```
thẻ.extend (mới)
```



Hoàn thành chương trình đào tạo Python

- Người chơi thêm nhiều thẻ sẽ sử dụng phần mở rộng()



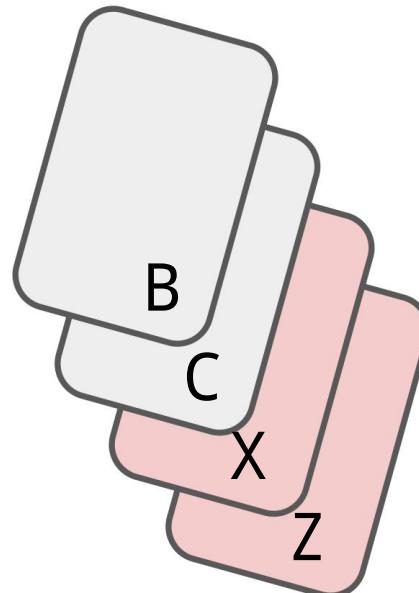
```
thẻ = [ "B" , "C" , "X" , "Z" ]
```

```
thẻ.extend (mới)
```



Hoàn thành chương trình đào tạo Python

- Đừng sử dụng `add()` nếu không các danh sách sẽ bị lồng vào nhau!



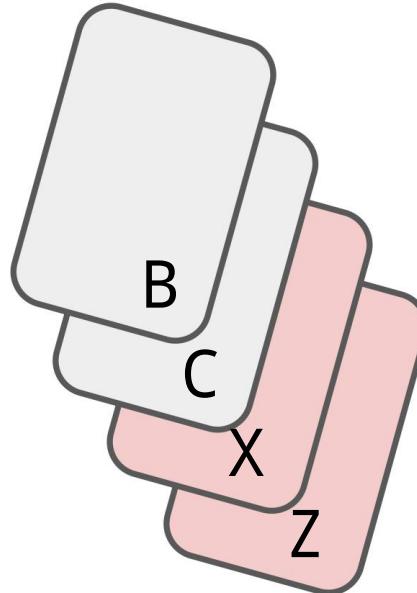
```
thẻ = [ "B", "C", ["X", "Z"] ]
```

`thẻ.append (mới)`

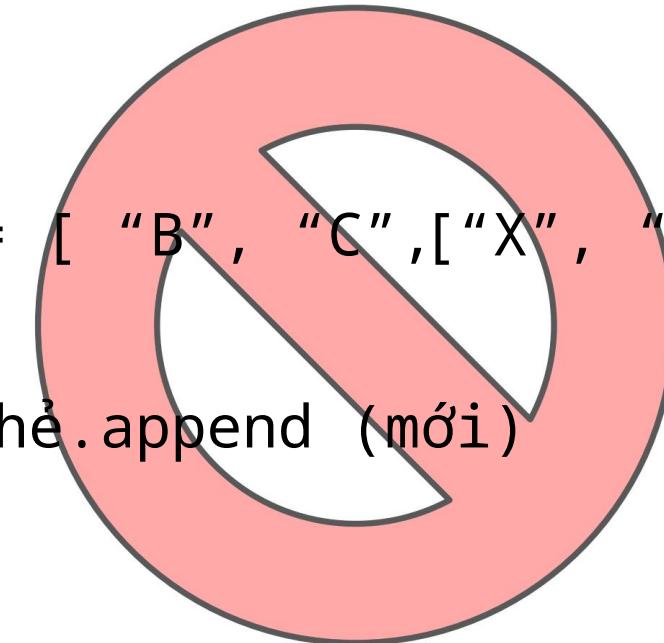


Hoàn thành chương trình đào tạo Python

- Đừng sử dụng add() nếu không các danh sách sẽ bị lồng vào nhau!



thẻ = ["B", "C", ["X", "Z"]]
thẻ.append (mới)





Bắt đầu nào !



Trò chơi logic

PHẦN MỘT



Hoàn thành chương trình đào tạo Python

- Tạo logic tổng thể thường là phần khó nhất của một dự án như thế này!
- Điều quan trọng cần lưu ý là chúng tôi đã lên kế hoạch cho các lớp xoay quanh logic sắp tới, vì vậy, trong tình huống thực tế, bạn thường nghĩ đến cả cấu trúc logic và lớp cùng một lúc.



Hoàn thành chương trình đào tạo Python

- Hãy phác thảo logic của chúng ta cho trò chơi!



Hoàn thành chương trình đào tạo Python

- Hãy phác thảo logic của chúng ta cho trò chơi!

Người chơi một

Người chơi thứ hai



Hoàn thành chương trình đào tạo Python

- Hãy phác thảo logic của chúng ta cho trò chơi!

Người chơi một

Bộ bài mới

Người chơi thứ hai



Hoàn thành chương trình đào tạo Python

- Hãy phác thảo logic của chúng ta cho trò chơi!

Người chơi một

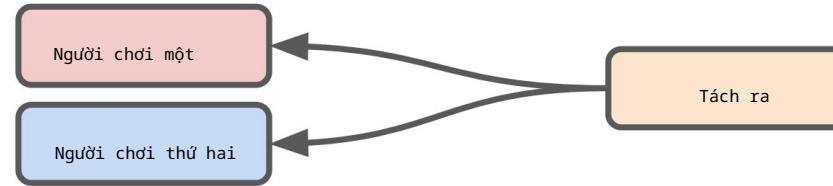
Người chơi thứ hai

Trộn bài



Hoàn thành chương trình đào tạo Python

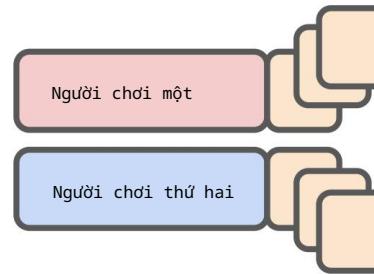
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

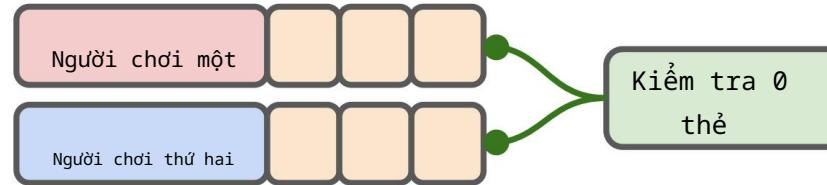
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

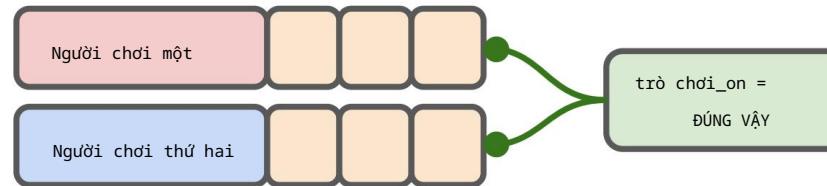
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

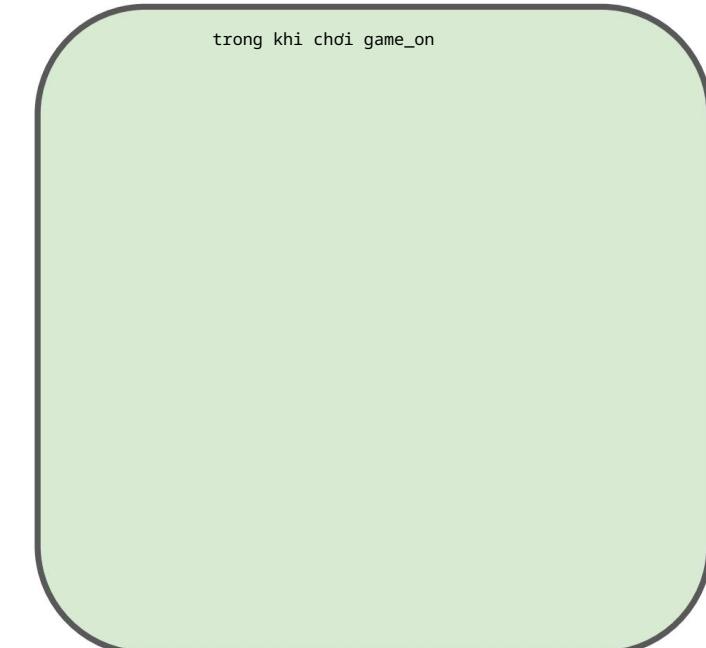
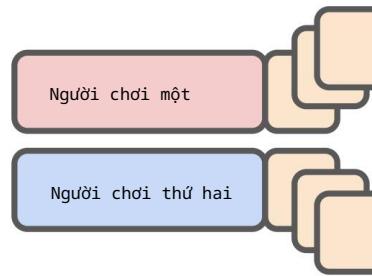
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

- Hãy phác thảo logic của chúng ta cho trò chơi!

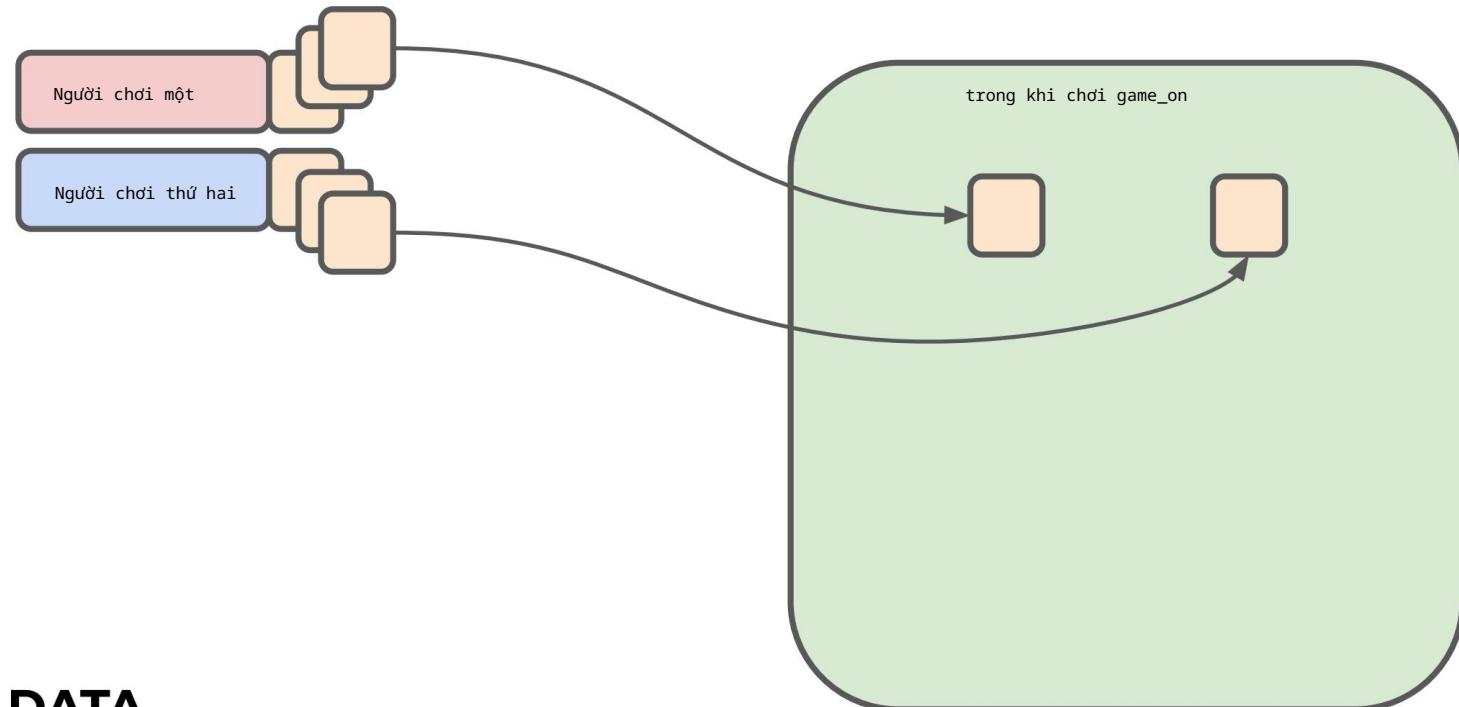


trong khi chơi game_on



Hoàn thành chương trình đào tạo Python

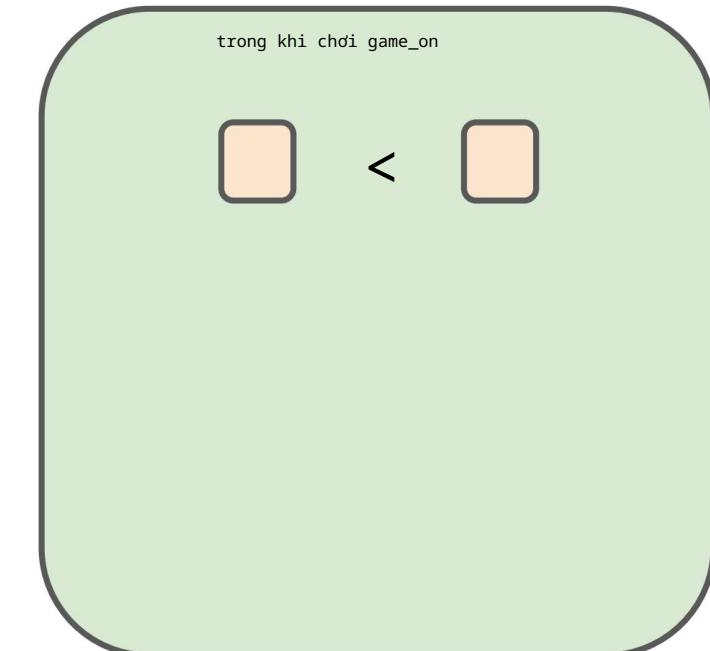
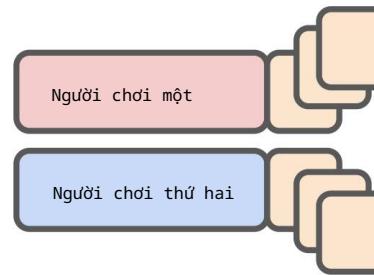
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

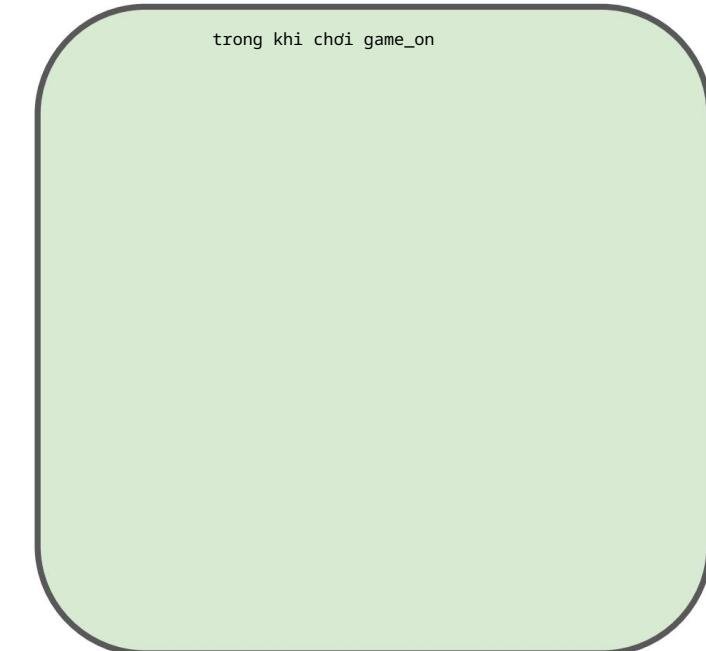
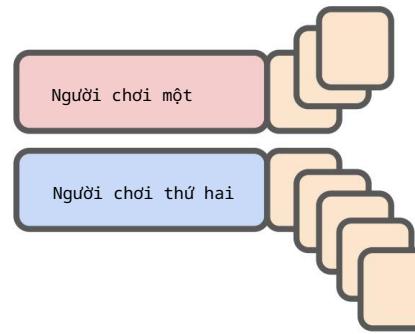
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

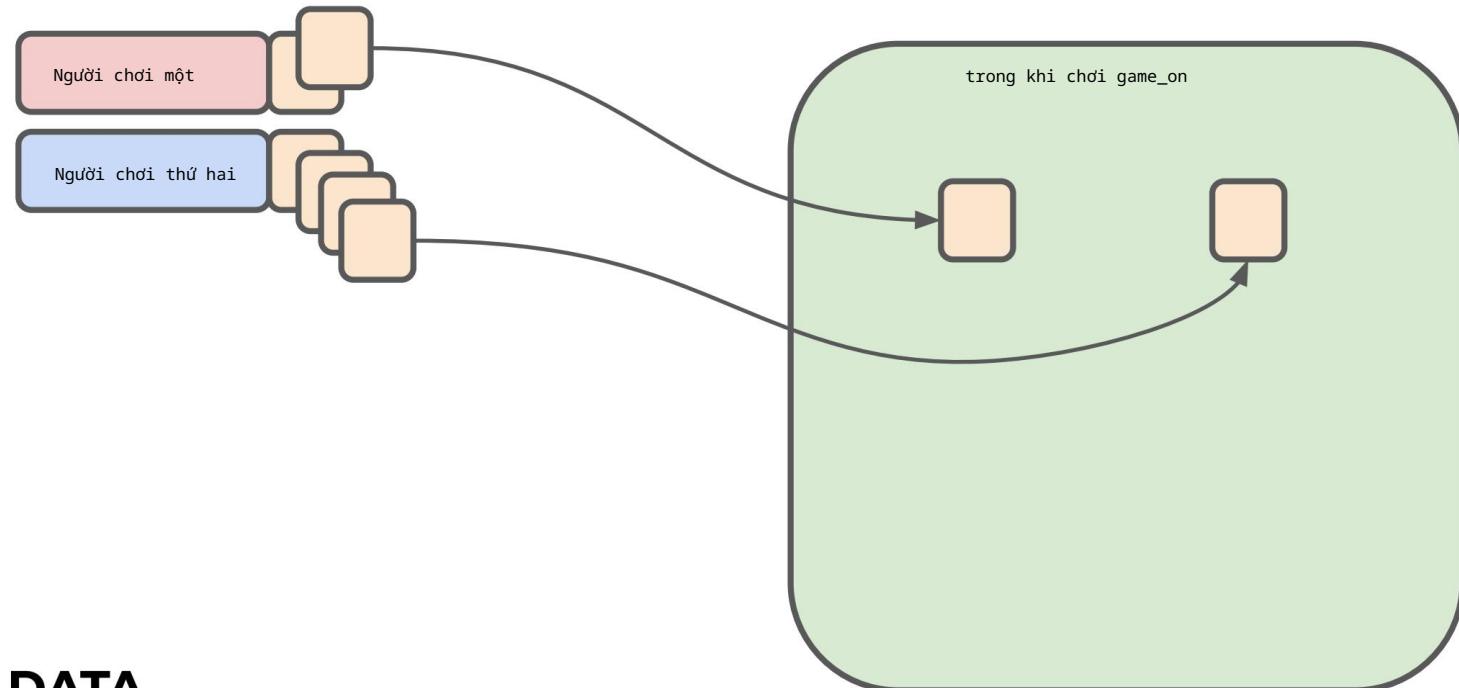
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

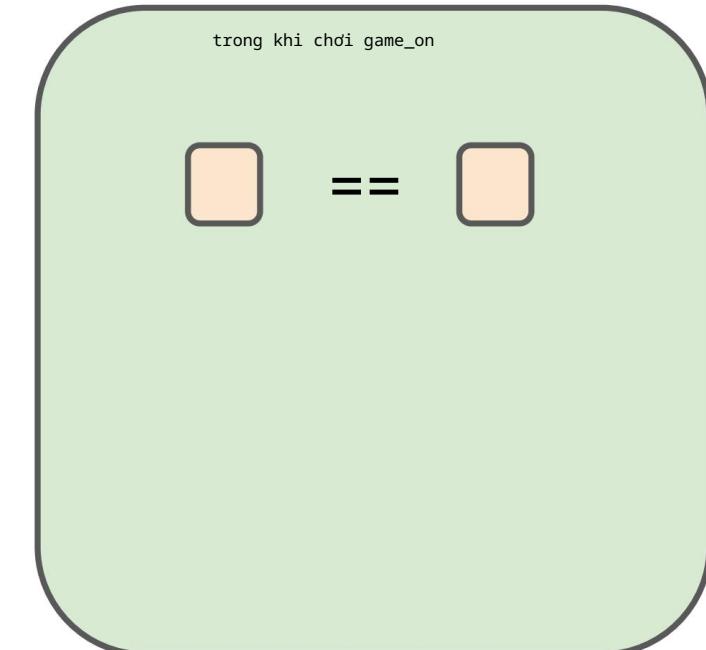
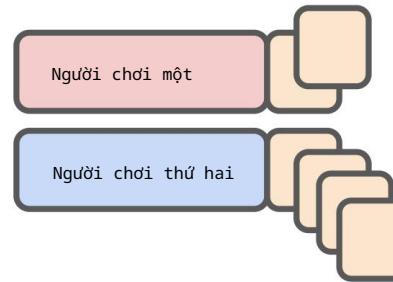
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

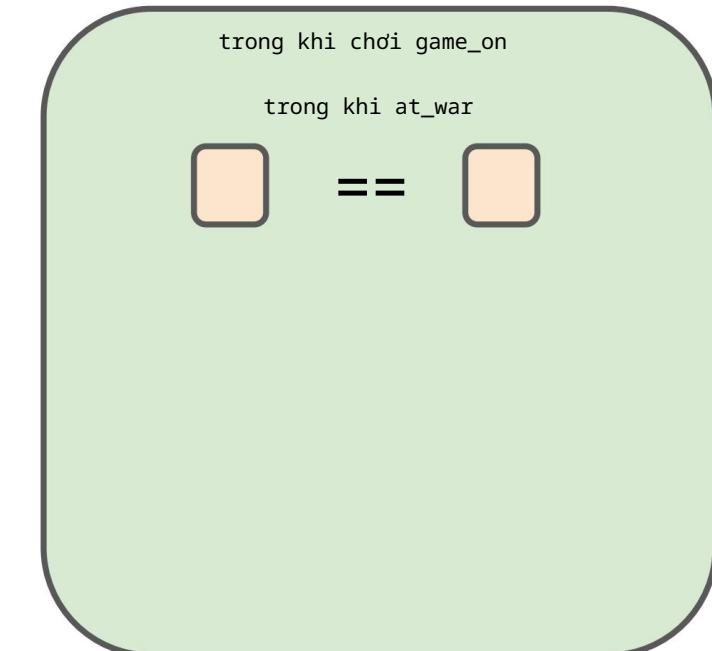
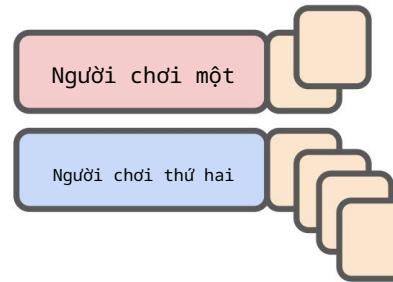
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

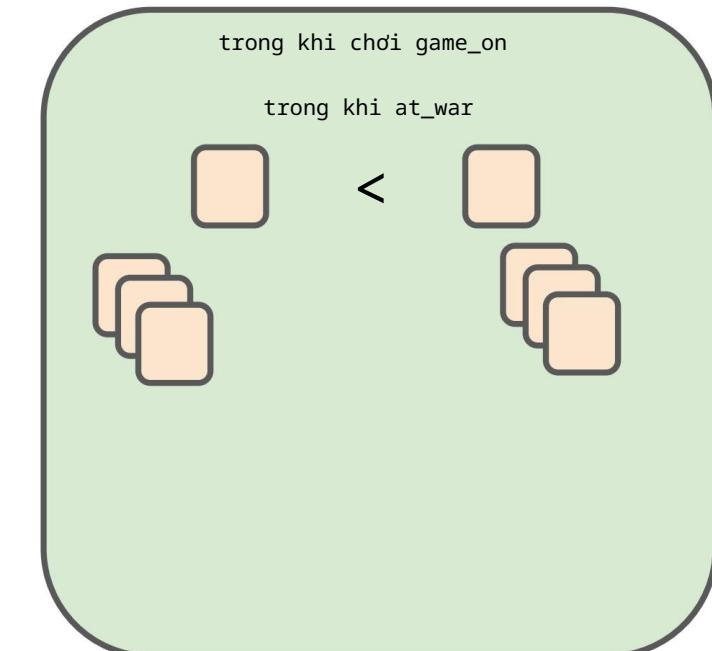
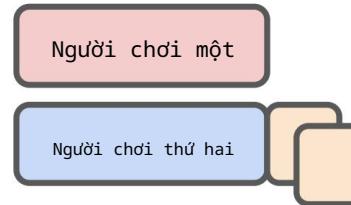
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

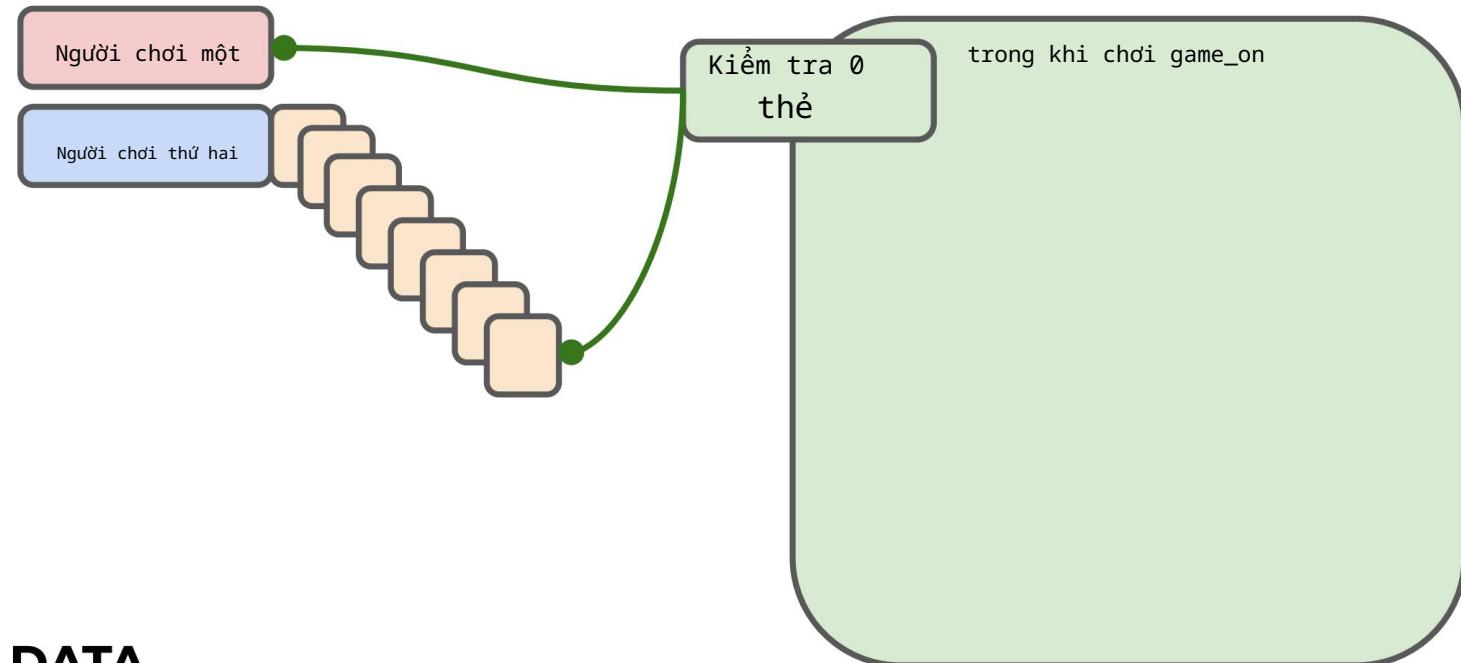
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

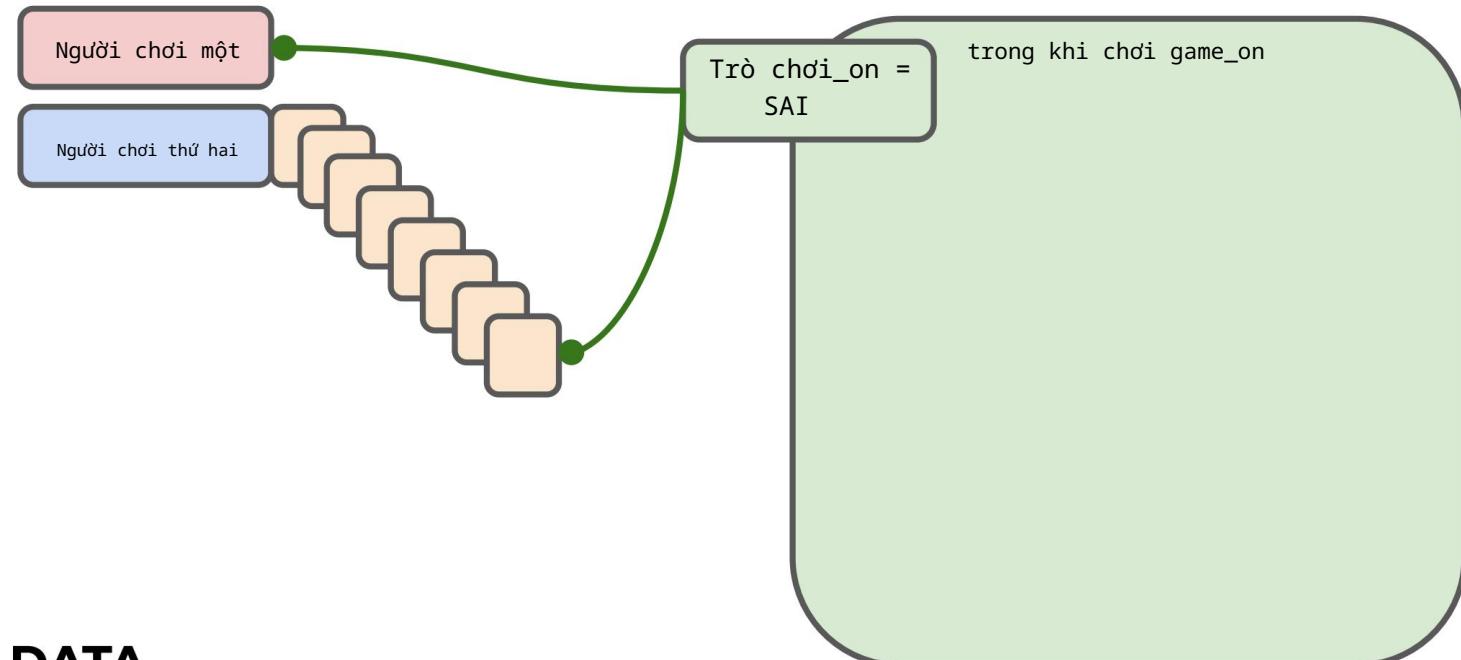
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

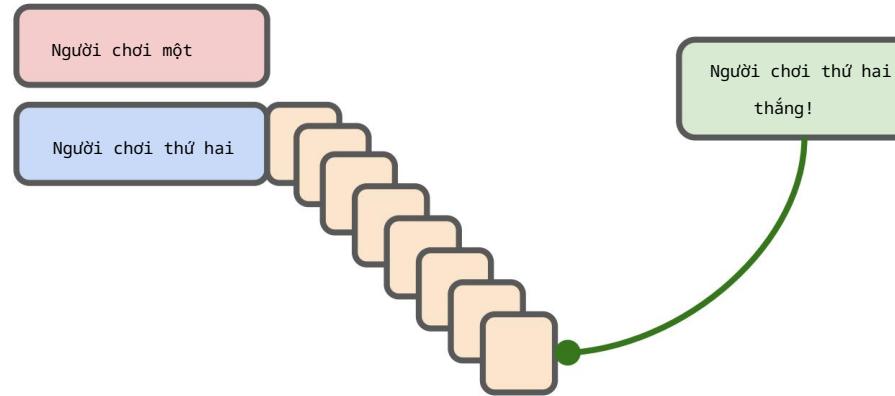
- Hãy phác thảo logic của chúng ta cho trò chơi!





Hoàn thành chương trình đào tạo Python

- Hãy phác thảo logic của chúng ta cho trò chơi!





Trò chơi logic

PHẦN HAI



Trò chơi logic

MỘT PHẦN BA



Hoàn thành chương trình đào tạo Python

- Bây giờ là lúc kiểm tra bài của người chơi với nhau. • Có rất nhiều cách để thực hiện việc này!
- Chúng ta có 3 tình huống:
 - Người chơi Một > Người chơi Hai
 - Người chơi Một < Người chơi Hai
 - Người chơi Một == Người chơi Hai



Hoàn thành chương trình đào tạo Python

- Cách chúng ta viết bài này là sử dụng một `if/elif/else` trong một vòng lặp `while` giả định rằng một “cuộc chiến” đã xảy ra.
- Chúng tôi sẽ tuyên bố `at_war = Sai` nếu người chơi giải quyết việc so khớp trên quân bài được rút đầu tiên, nếu không chúng tôi sẽ thêm quân bài vào các quân bài hiện tại trên bàn.

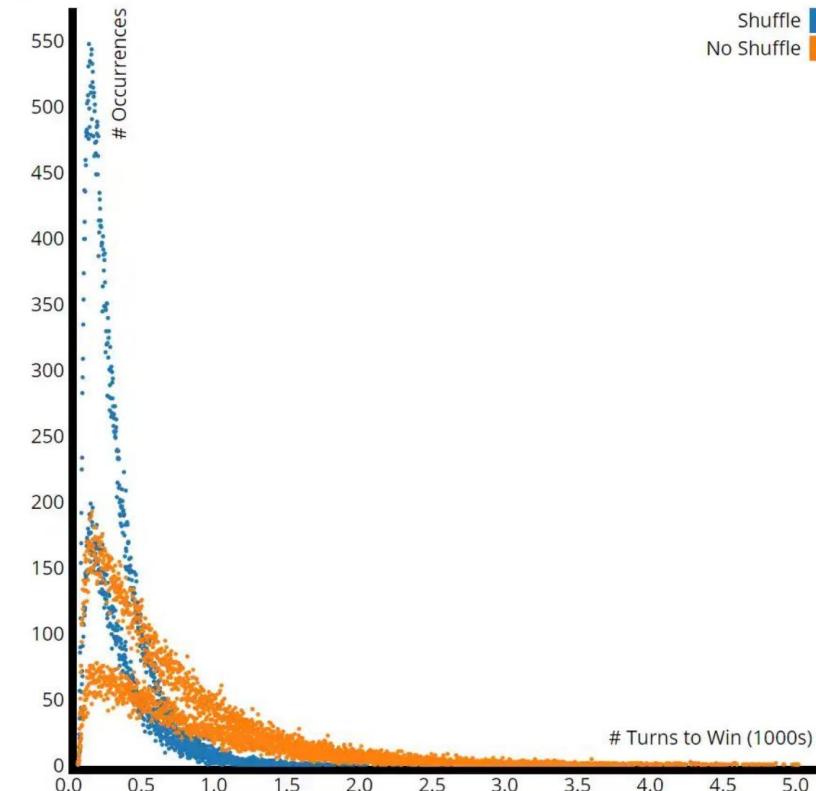


Hoàn thành chương trình đào tạo Python

- Quy tắc chúng tôi sẽ sử dụng trong phiên bản này là nếu hòa, mỗi người chơi cần rút thêm 5 lá bài.
- Chúng tôi cũng sẽ nói rằng người chơi thua nếu họ không có ít nhất 5 lá bài để chơi cuộc chiến.
- Logic này có thể dễ dàng chỉnh sửa để phù hợp với bất kỳ quy tắc nào cấu trúc bạn muốn.



Hoàn thành chương trình đào tạo Python





Hoàn thành chương trình đào tạo Python

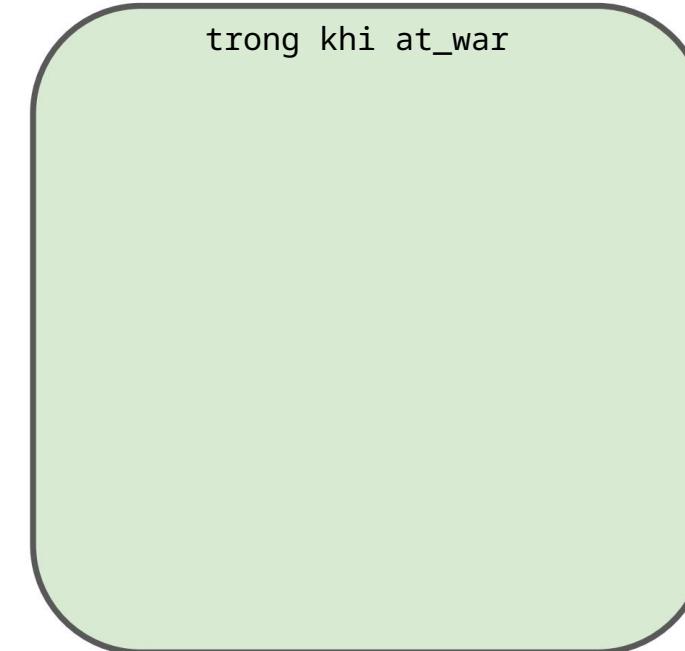
- Hãy nhanh chóng khám phá vòng lặp này một cách trực quan trước khi chúng tôi mã hóa nó!



Hoàn thành chương trình đào tạo Python

- So sánh logic trò chơi

trong khi at_war





Hoàn thành chương trình đào tạo Python

at_war = Đúng

trong khi at_war



Hoàn thành chương trình đào tạo Python

at_war = Đúng

trong khi at_war

- Một > Hai
- Thêm thẻ vào

Một • at_war = Sai

nếu như



Hoàn thành chương trình đào tạo Python

at_war = Đúng

trong khi at_war

- Một > Hai
 - Thêm thẻ vào
- Một • at_war = Sai

nếu như

yêu tinh

- Một < Hai
 - Thêm thẻ vào
- Hai • at_war = Sai



Hoàn thành chương trình đào tạo Python

at_war = Đúng

trong khi at_war

- Một > Hai •
Thêm thẻ vào Một •
at_war = Sai

nếu như

yêu tính

- Một < Hai •
Thêm thẻ vào Hai •
at_war = Sai

khác

- Kiểm tra xem người chơi
có đủ bài hay
không • Rút thêm bài



Hoàn thành chương trình đào tạo Python

at_war = Đúng

trong khi at_war

nếu như

- Một > Hai •
Thêm thẻ vào Một •
at_war = Sai

yêu tính

- Một < Hai •
Thêm thẻ vào Hai •
at_war = Sai

khác

- Kiểm tra xem người chơi
có đủ bài hay
không • Rút thêm bài



Dự án cột mốc 2



Hoàn thành chương trình đào tạo Python

- Chúng tôi đã học đủ để bắt đầu dự án quan trọng thứ hai!
- Bạn có thể xử lý dự án này theo một số cách:
 - Mã hóa dự án cùng với các giải pháp.
 - Tự mình thực hiện dự án.
 - Sử dụng sách bài tập làm tài liệu hướng dẫn cho dự án của riêng bạn.



Hoàn thành chương trình đào tạo Python

- Đối với dự án này, bạn sẽ sử dụng OOP để tạo Trò chơi BlackJack bằng Python.
- Hãy nhanh chóng điỂm qua ý chính của trò chơi và thảo luận về cách sử dụng OOP cho dự án này.



Hoàn thành chương trình đào tạo Python

- Đối với phiên bản trò chơi của chúng tôi, chúng tôi sẽ chỉ có người chia bài là máy tính và người chơi là con người.
- Chúng ta bắt đầu với một bộ bài thông thường, bạn sẽ tạo một hình ảnh đại diện của bộ bài bằng Python.



Hoàn thành chương trình đào tạo Python

ĐẠI LÝ MÁY TÍNH



NGƯỜI CHƠI CON NGƯỜI



Hoàn thành chương trình đào tạo Python

ĐẠI LÝ MÁY TÍNH



ĐỊA ĐIỂM NGƯỜI CHƠI
CƯỢC



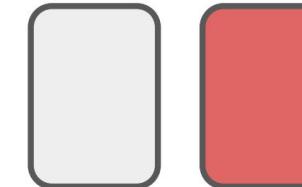
NGƯỜI CHƠI CON NGƯỜI



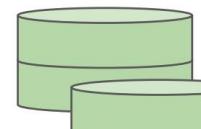
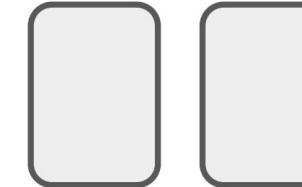
Hoàn thành chương trình đào tạo Python

ĐẠI LÝ MÁY TÍNH

Nhà cái bắt đầu với 1 lá bài ngửa và 1 lá bài ngửa Xuống



Người chơi bắt đầu với 2 lá bài ngửa



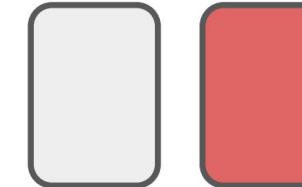
NGƯỜI CHƠI CON NGƯỜI



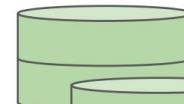
Hoàn thành chương trình đào tạo Python

ĐẠI LÝ MÁY TÍNH

Nhà cái bắt đầu với 1 lá bài ngửa và 1 lá bài ngửa Xuống

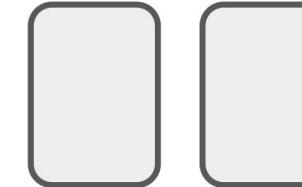


Người chơi bắt đầu với 2 lá bài ngửa



NGƯỜI CHƠI CON NGƯỜI

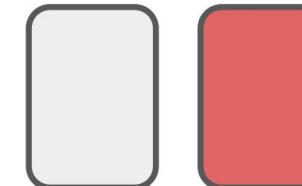
NGƯỜI CHƠI ĐI
ĐẦU TIÊN Ở
CHƠI TRÒ CHƠI





Hoàn thành chương trình đào tạo Python

ĐẠI LÝ MÁY TÍNH



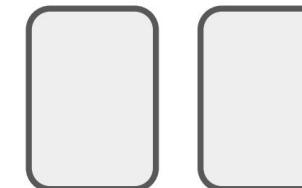
MỤC TIÊU CỦA NGƯỜI CHƠI: Đạt được tổng giá trị gần hơn là 21 so với nhà cái.



HIT để có thêm
thẻ từ DECK

Hành động có thể:

1. Đánh (Nhận thêm lá bài)
2. Ở lại (Ngưng nhận thẻ)



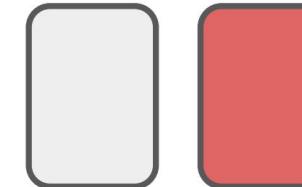
NGƯỜI CHƠI CON NGƯỜI

Chúng tôi sẽ bỏ qua những hành động như
“Bảo hiểm”, “Chia đôi” hoặc “Đôi
Xuông”



Hoàn thành chương trình đào tạo Python

ĐẠI LÝ MÁY TÍNH



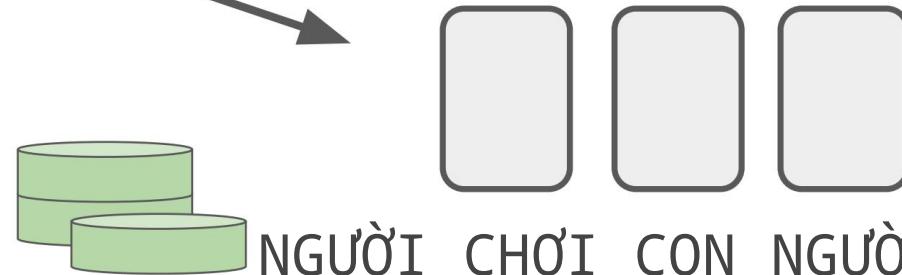
MỤC TIÊU CỦA NGƯỜI CHƠI: Đạt được tổng giá trị gần hơn là 21 so với nhà cái.



HIT để có thêm
thẻ từ DECK

Hành động có thể:

1. Đánh (Nhận thêm lá bài)
2. Ở lại (Ngưng nhận thẻ)

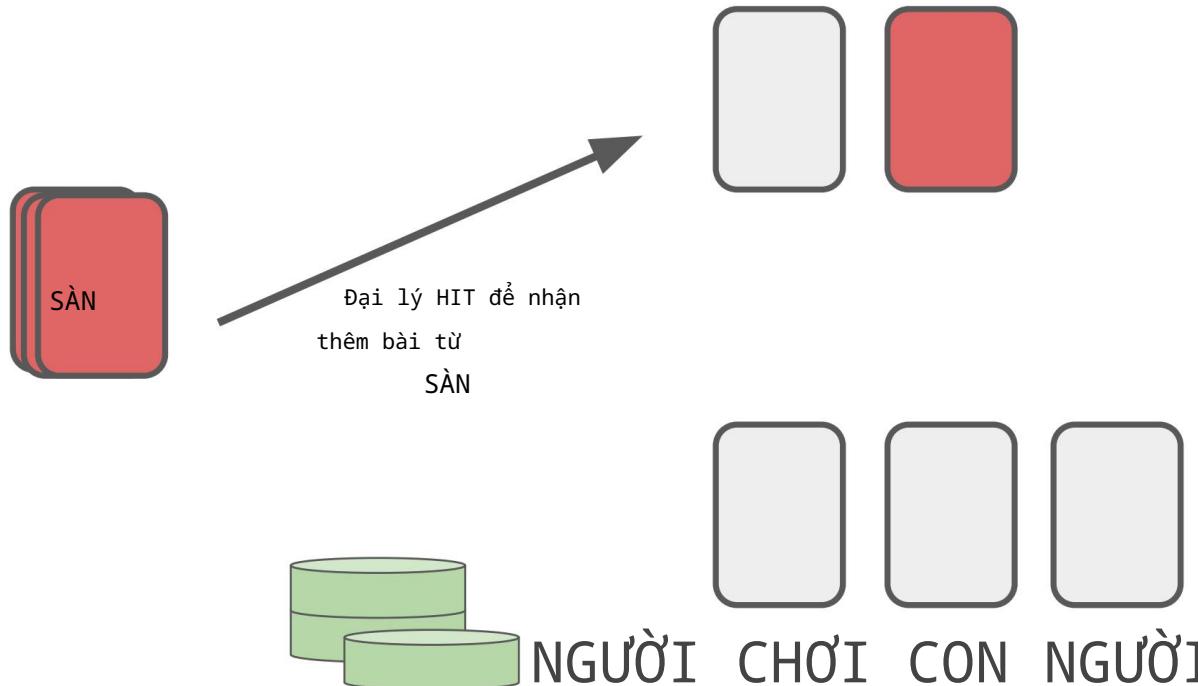


Chúng tôi sẽ bỏ qua những hành động như "Bảo hiểm", "Chia đôi" hoặc "Đôi Xuông"



Hoàn thành chương trình đào tạo Python

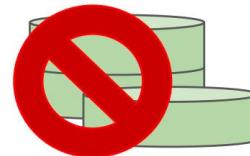
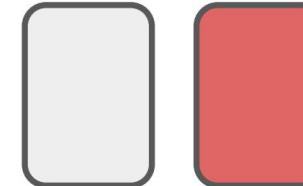
ĐẠI LÝ MÁY TÍNH



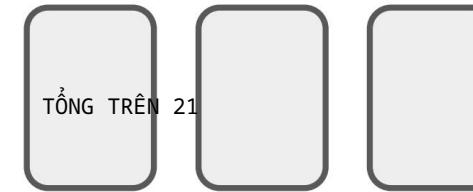


KẾT THÚC TRÒ CHƠI: NGƯỜI CHƠI BẮT BUỘC

ĐẠI LÝ MÁY TÍNH



NGƯỜI CHƠI CON NGƯỜI



SAU KHI NGƯỜI CHƠI LUỘT:

1. Nếu người chơi tiếp tục đánh vượt quá 21, họ sẽ thua và thua cược!

Trò chơi sau đó kết thúc và người chia bài sẽ thu thập tiền bạc.



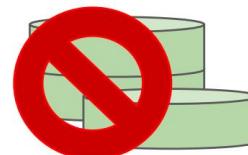
KẾT THÚC TRÒ CHƠI: Computer Beats Player

ĐẠI LÝ MÁY TÍNH

Tổng của máy tính
cao hơn tổng của người
chơi và vẫn dưới 21.



SAU KHI NGƯỜI CHƠI LUỘT:
2. Nếu người chơi dưới 21
tuổi, người chia bài sẽ đánh
cho đến khi họ đánh bại
người chơi hoặc người chia
bài bị phá sản.



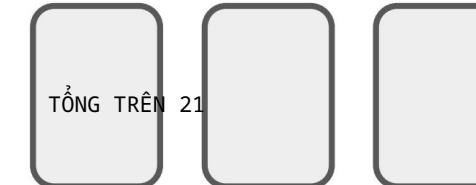
NGƯỜI CHƠI CON NGƯỜI





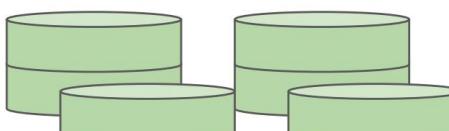
KẾT THÚC TRÒ CHƠI: NGƯỜI CHƠI THẮNG

ĐẠI LÝ MÁY TÍNH

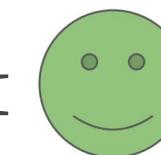
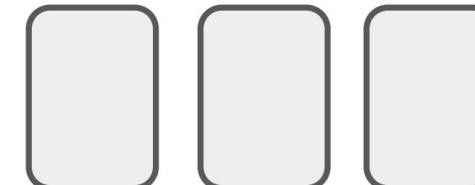


SAU KHI NGƯỜI CHƠI LUỘT:

2. Nếu người chơi dưới 21 tuổi, người chia bài sẽ đánh cho đến khi họ đánh bại người chơi hoặc người chia bài bị phá sản.



NGƯỜI CHƠI CON NGƯỜI





Hoàn thành chương trình đào tạo Python

- Quy tắc đặc biệt:
 - Thẻ Mặt (Jack, Queen, King) được tính là giá trị 10.
 - Ách có thể được tính là 1 hoặc 11 bất kỳ giá trị nào phù hợp hơn với người chơi.



Hoàn thành chương trình đào tạo Python

- Hãy xem các liên kết tài nguyên để biết các giải thích khác về BlackJack để biết thêm thông tin.
- Vậy giờ chúng ta hãy khám phá dự án và bắt đầu làm việc!



Dự án cột mốc 2 Giải pháp ví dụ