



# Boolean Expressions and Conditionals

# Boolean

- Something that evaluates to True or False
- Typically shown with relational operator and/or boolean operator

# Boolean

- Something that evaluates to True or False
- Typically shown with relational operator and/or boolean operator
  - "Hello" == "hello"
  - 4 >= 2

# Boolean Operators

- not, and, or
- Can be used to express more with booleans
  - It is not rainy: `weather != "rain"`

# Boolean Operators

- not, and, or
- Can be used to express more with booleans
  - It is not rainy: `not (weather == "rain")`

# Boolean Operators

- not, and, or
- Can be used to express more with booleans
  - It is not rainy: **not** (weather == "rain")
  - It is rainy and it is cold: (weather == "rain") **and** (weather == "cold")

# Boolean Operators

- not, and, or
- Can be used to express more with booleans
  - It is not rainy: **not** (weather == "rain")
  - It is rainy and it is cold: (weather == "rain") **and** (weather == "cold")
  - It is rainy or it is snowy: (weather == "rain") **or** (weather == "snow")

# Not

- `not` inverts the value of a boolean expression

b	<code>not b</code>



# Not

- `not` inverts the value of a boolean expression

b	<code>not b</code>
True	False
False	True

# and

- booleans combined with **and** evaluate to True if and only if both booleans are True

a	b	a <b>and</b> b

# and

- booleans combined with **and** evaluate to True if and only if both booleans are True

a	b	a <b>and</b> b
True	True	True
True	False	False
False	True	False
False	False	False

or

- booleans combined with **or** evaluate to True if at least one is True

a	b	a <b>or</b> b

# or

- booleans combined with **or** evaluate to True if at least one is True

a	b	a or b
True	True	True
True	False	True
False	True	True
False	False	False

# Ordering

P

E

MD

AS

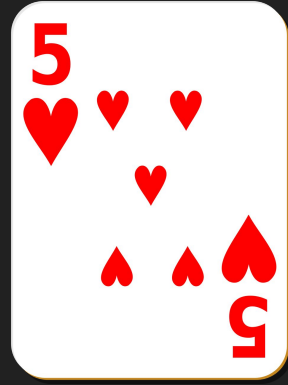
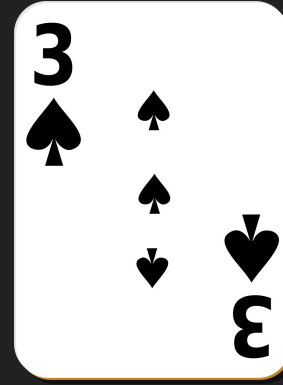
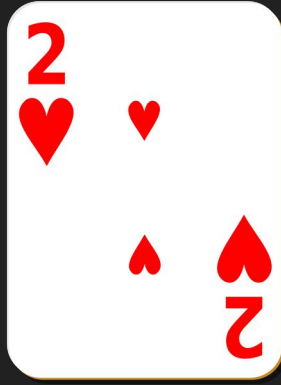
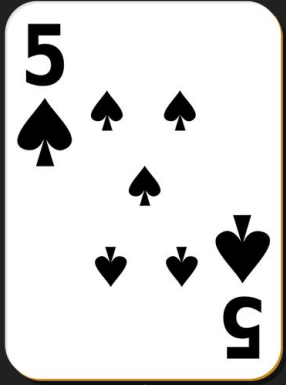
not

and

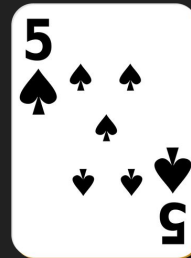
or

# Conditionals

## Recall: Finding the Lowest Card



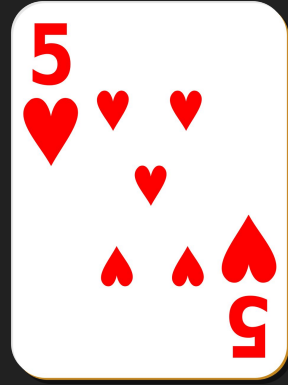
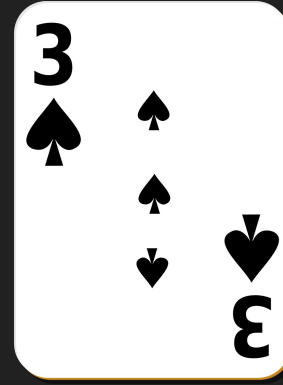
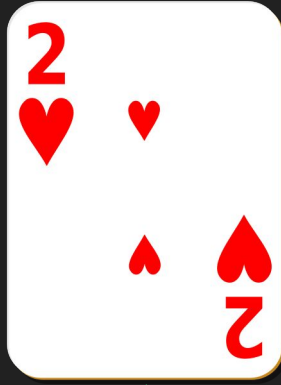
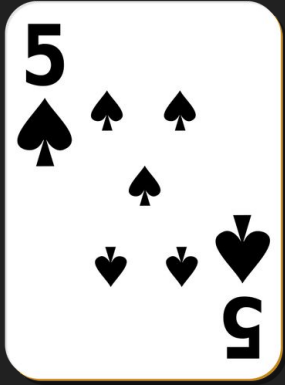
Low card:



If current card < low card,  
make it the low card.



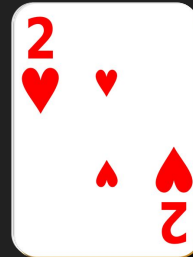
## Recall: Finding the Lowest Card



$2 < 5?$

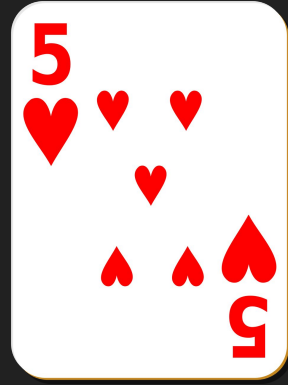
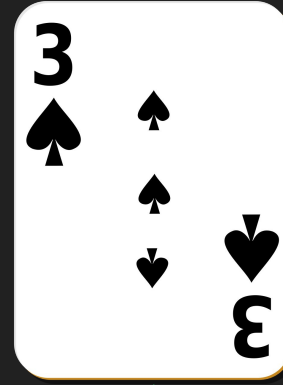
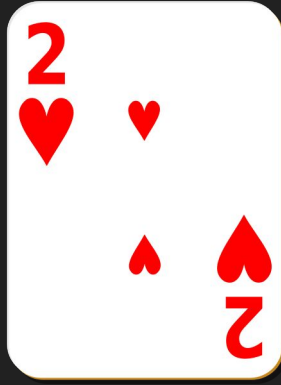
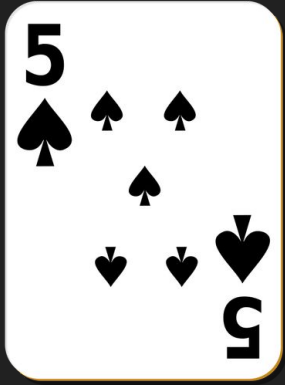


Low card:



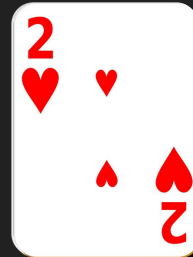
If current card  $<$  low card,  
make it the low card.

## Recall: Finding the Lowest Card



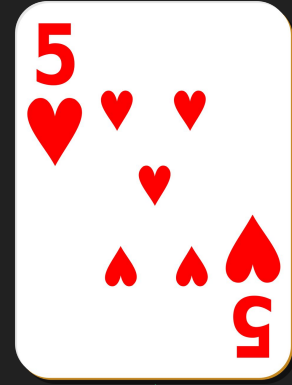
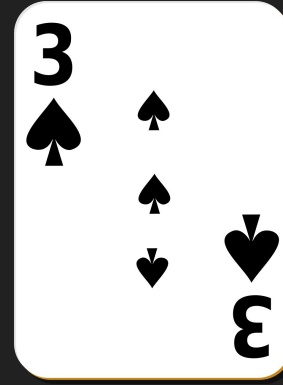
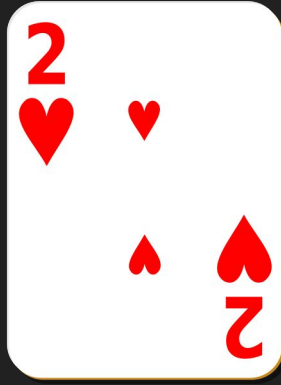
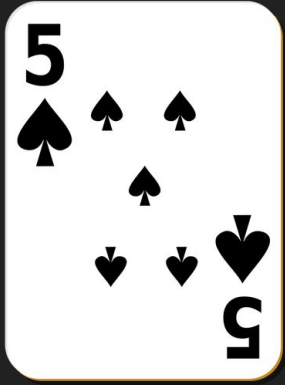
$3 < 2?$  

Low card:



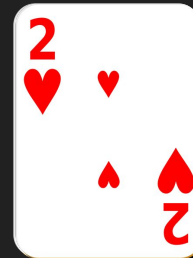
If current card  $<$  low card,  
make it the low card.

## Recall: Finding the Lowest Card



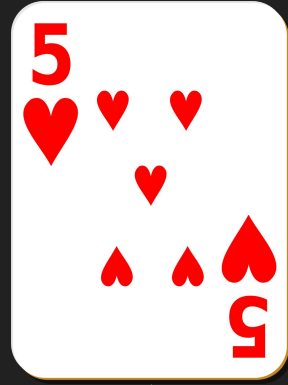
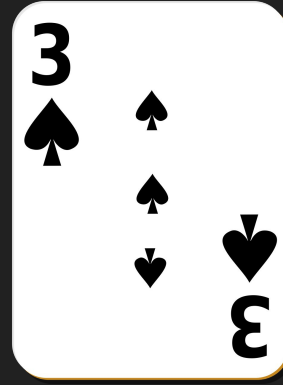
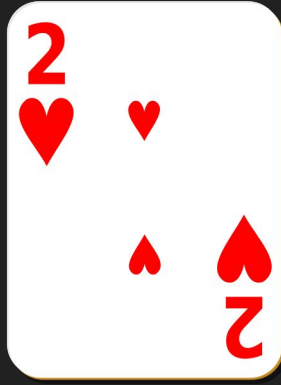
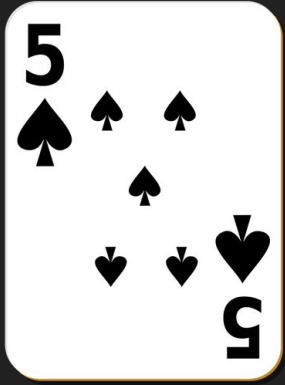
5 < 2? 

Low card:



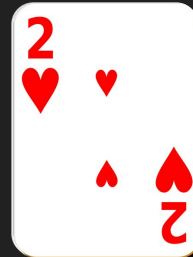
If current card < low card,  
make it the low card.

## Recall: Finding the Lowest Card



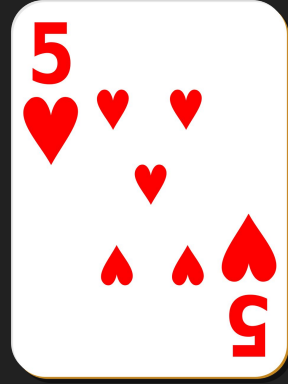
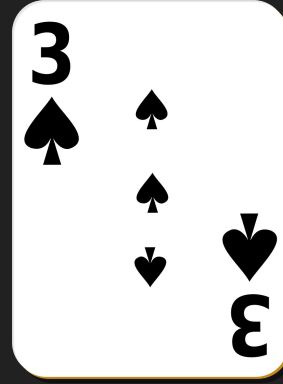
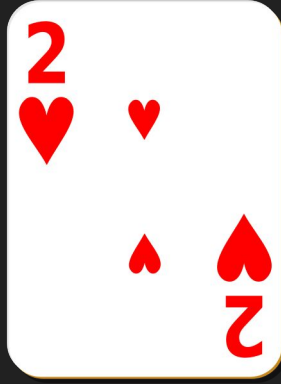
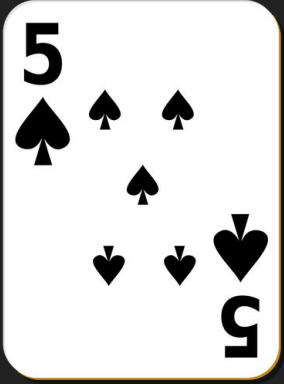
5 < 2? 

Low card:

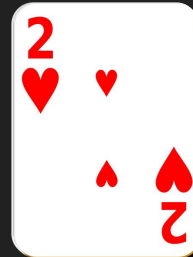


If current card < low card,  
make it the low card.

## Recall: Finding the Lowest Card



Low card:



Conditional Statement



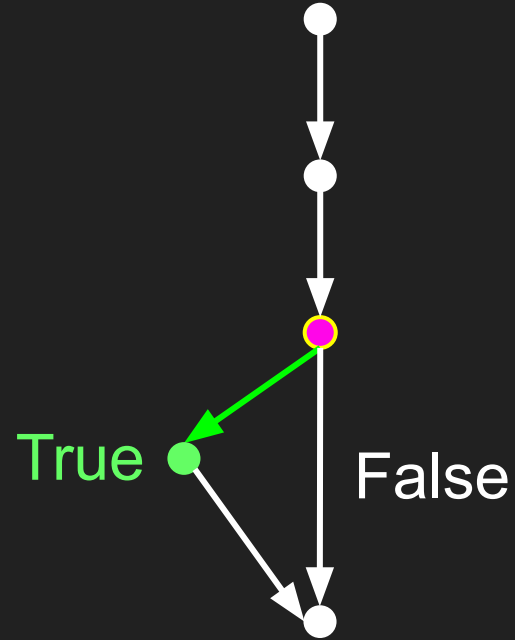
If current card < low card,  
make it the low card.

# Conditional Statements

if <something>: ← bool

<do something>

<rest of program>



# Conditional Statements

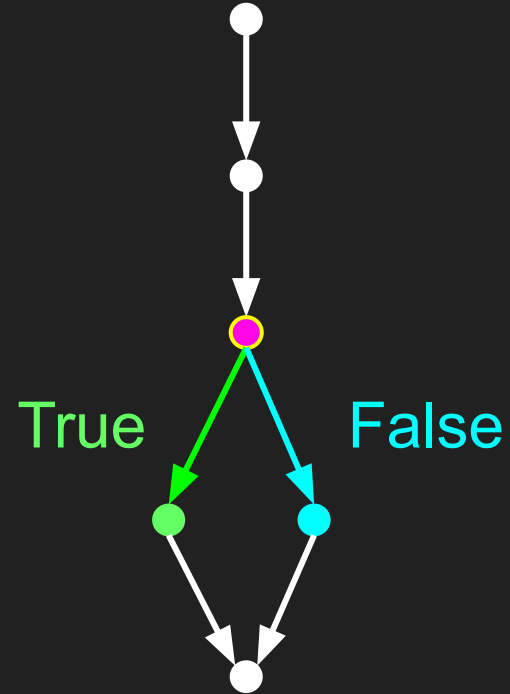
if <something>:

    <do something>

else:

    <do something else>

<rest of program>



# Conditional Statements

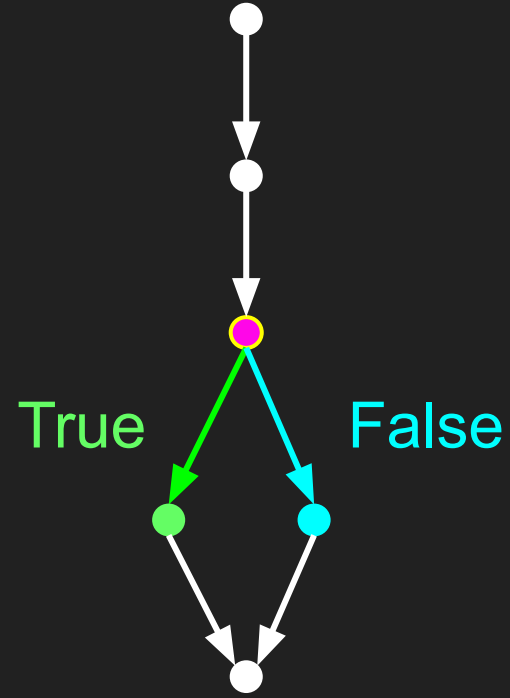
if <something>:

    <do something>

else:

    <do something else>

<rest of program>





# Discussion

What is a decision you make in your day-to-day that you can express as an conditional (if-else) statement?

E.g. If I my assignment is due tomorrow, I start working on it. Else (it's not due tomorrow), I procrastinate another day.

*(This is bad behavior and I don't condone it!)*

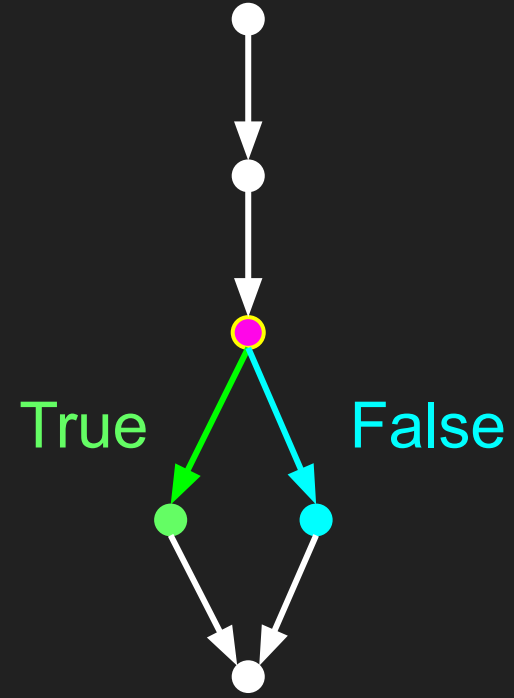
# Conditional Statements

if

:



else:



# Practice

Write a function called `check_first_letter` that takes as input two `strs`: `word` and `letter`

It should return `"match!"` if the first character of `word` is `letter`

Otherwise, it should return `"no match!"`

Examples:

- `check_first_letter(word="happy", letter="h")` would return `"match!"`
- `check_first_letter(word="happy", letter="s")` would return `"no match!"`