CL03 - Boolean Operators 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	not b

Not

- not inverts the value of a boolean expression

b	not b
True	False
False	True

and

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

a	b	a and b

and

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

a	b	a and 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 or 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

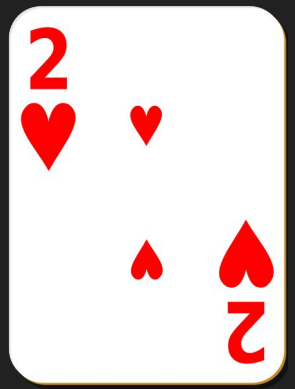
not

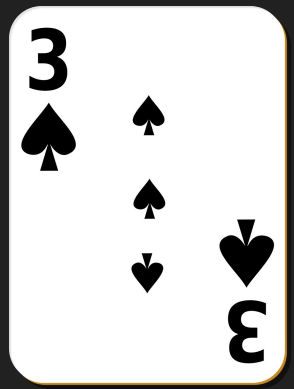
and

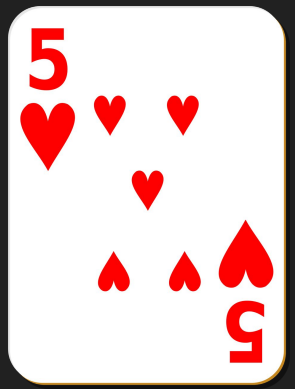
or

Conditionals

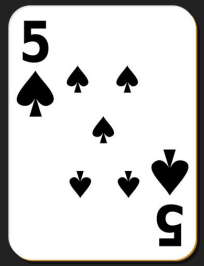
Recall: Finding the Lowest Card





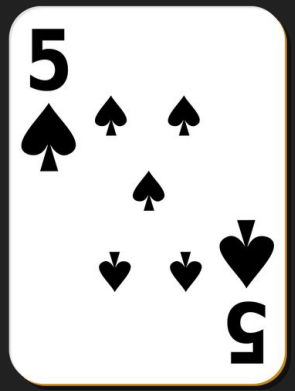


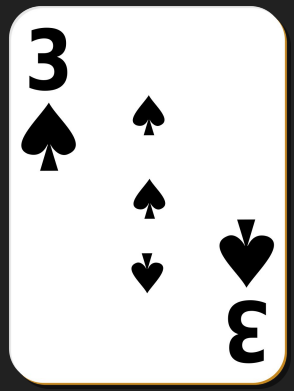
Low card:

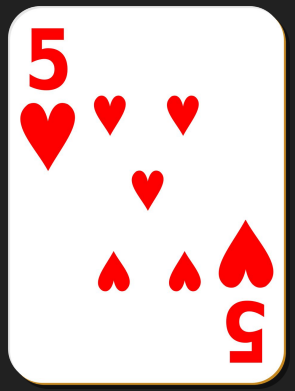


If current card < low card, make it the low card.

Recall: Finding the Lowest Card

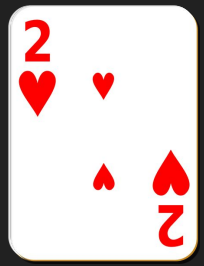






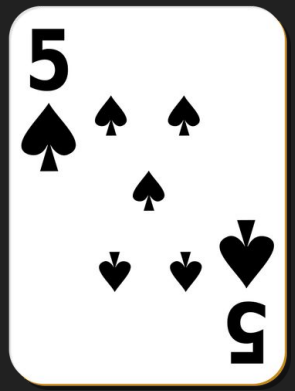
2 < 5? ✓

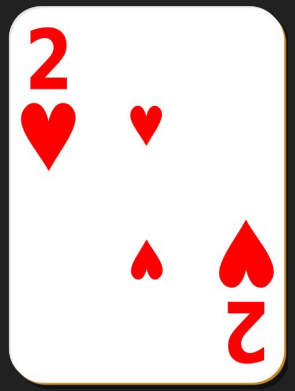
Low card:

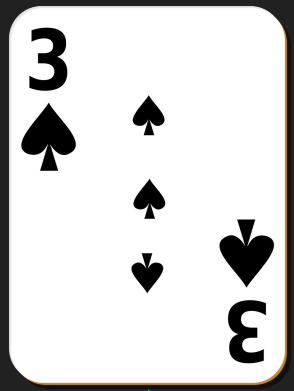


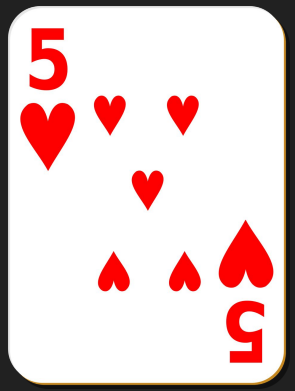
If current card < low card, make it the low card.

Recall: Finding the Lowest Card



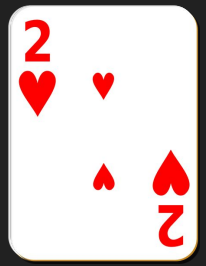






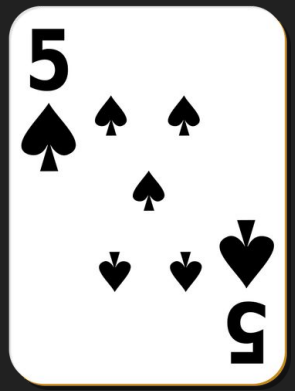
3 < 2? ✗

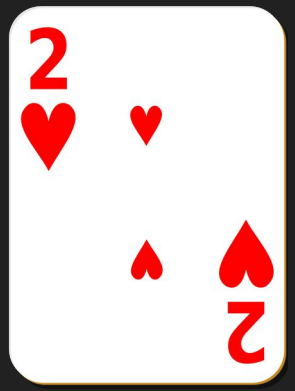
Low card:

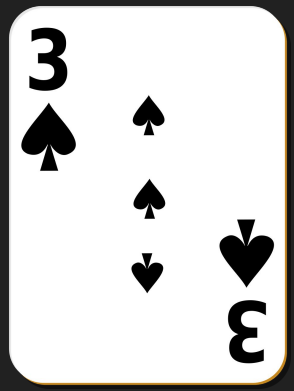


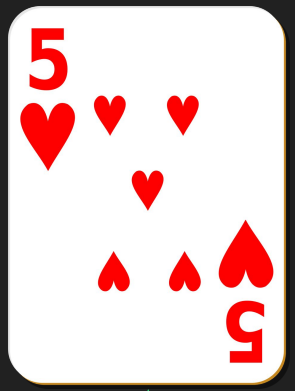
If current card < low card, make it the low card.

Recall: Finding the Lowest Card



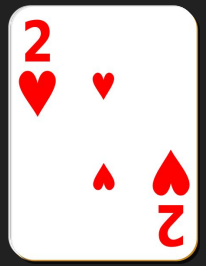






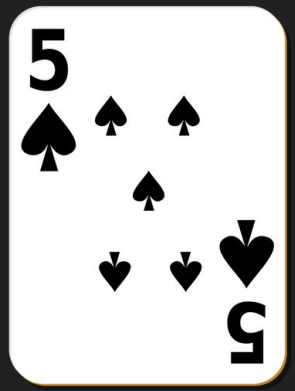
5 < 2?

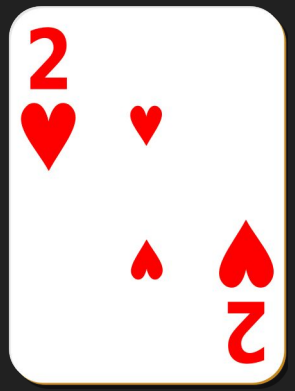
Low card:

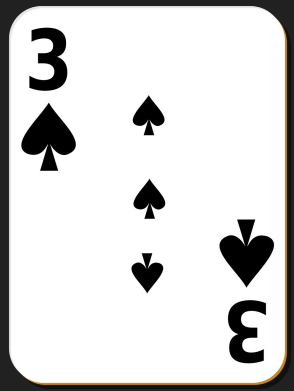


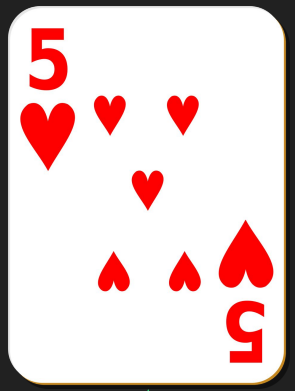
If current card < low card,
make it the low card.

Recall: Finding the Lowest Card



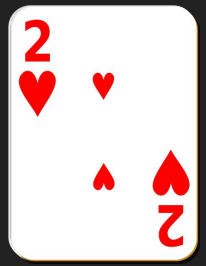






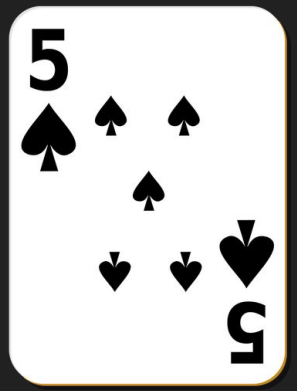
5 < 2?

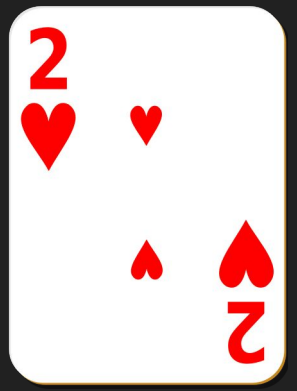
Low card:

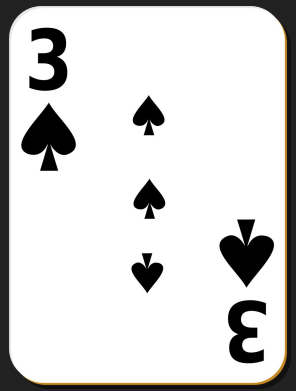


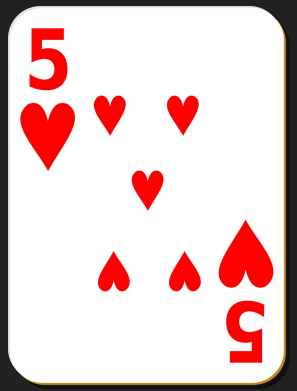
If current card < low card,
make it the low card.

Recall: Finding the Lowest Card

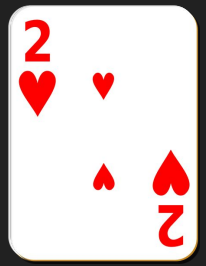








Low card:



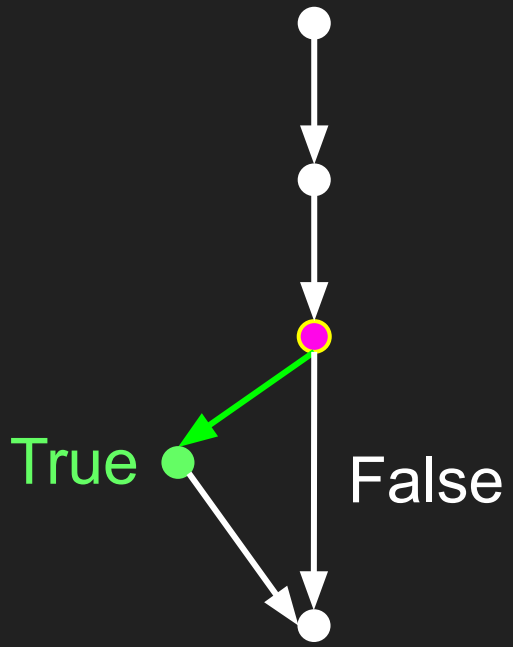
Conditional Statement

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

Conditional Statements

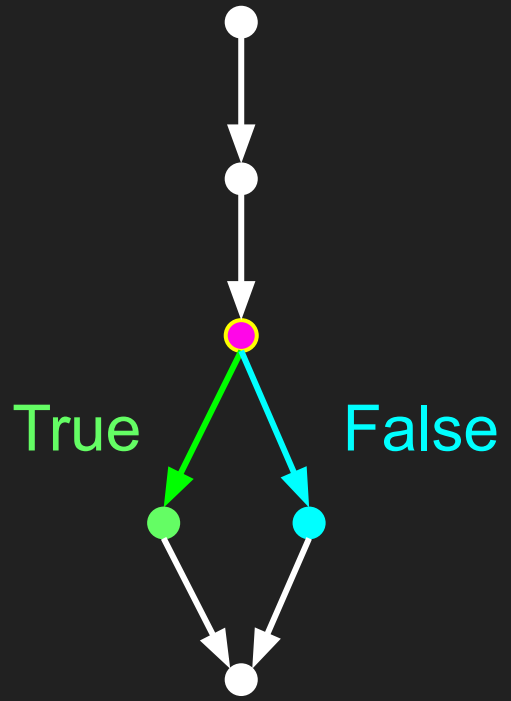
bool

```
if <something>:
    <do something>
<rest of program>
```



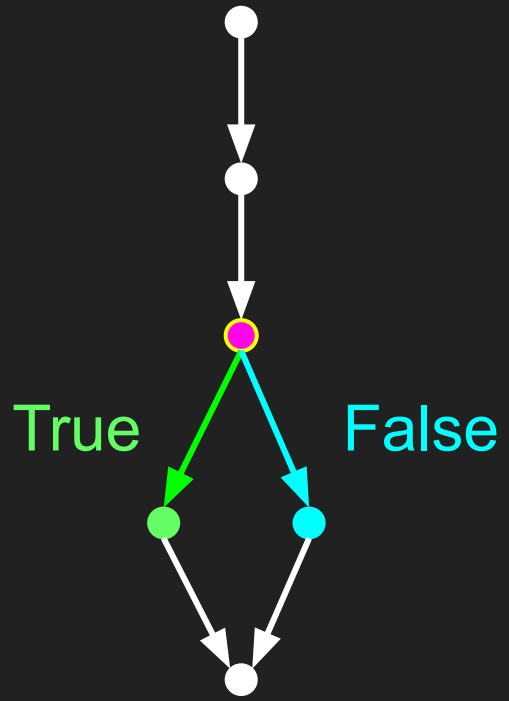
Conditional Statements

```
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

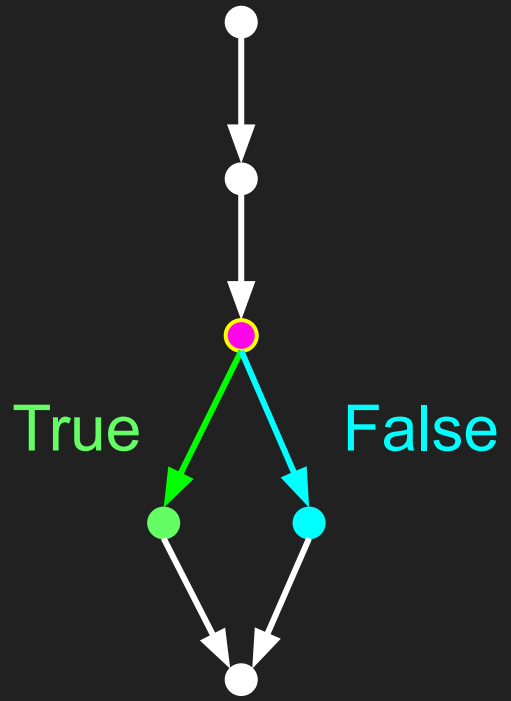
What is a decision you make in your day-to-day that you can express as an conditional (if-else) statement?

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 a 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!”

Diagram

```
1  def number_info(num: int) -> None:
2      if num < 10:
3          print("Small number.")
4      else:
5          if num % 2 == 0:
6              print("Even number.")
7          else:
8              print("Odd number.")
9      return num
10
11 number_info(num=11)
12 print(number_info(num=4))
```

What if…

```
1  def number_info(num: int) -> None:
2      if num < 10:
3          print("Small number.")
4      else:
5          if num % 2 == 0:
6              print("Even number.")
7          else:
8              print("Odd number.")
9      return num
```

What if…

```
1  def number_info(num: int) -> None:
2      if num < 10:
3          print("Small number.")
4      else:
5          if num % 2 == 0:
6              print("Even number.")
7          else:
8              print("Odd number.")
9      return num
```

What if…

```python
def number_info(num: int) -> None:
    if num < 10:
        print("Small number.")
    else:
        if num % 2 == 0:
            print("Even number.")
        else:
            print("Odd number.")
    return num
```

```python
def number_info(num: int) -> None:
    if num < 10:
        print("Small number.")
    elif num % 2 == 0:
        print("Even number.")
    else:
        print("Odd number.")
    return num
```

What if…

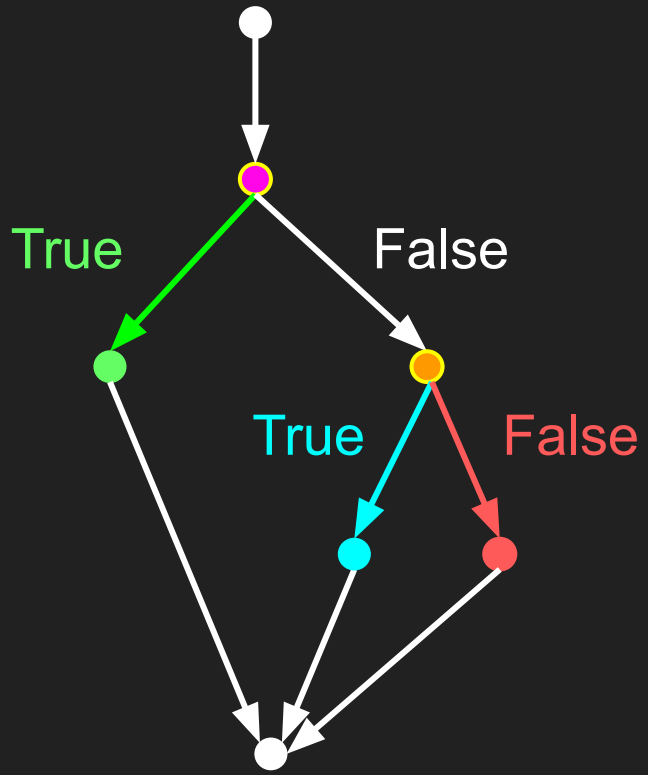
```
1  def number_info(num: int) -> None:
2      if num < 10:
3          print("Small number.")
4      else:
5          if num % 2 == 0:
6              print("Even number.")
7          else:
8              print("Odd number.")
9      return num
```

elif

```
1  def number_info(num: int) -> None:
2      if num < 10:
3          print("Small number.")
4      elif num % 2 == 0:
5          print("Even number.")
6      else:
7          print("Odd number.")
8      return num
```

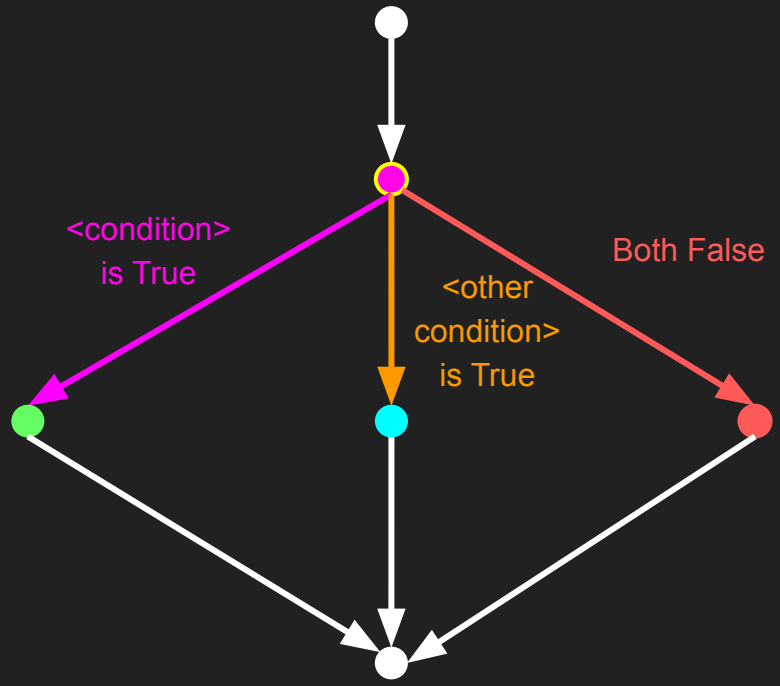
Previous Control Flow

```
if <condition>:
    <do something>
else:
    if <other condition>:
        <do something else>
    else:
        <do third thing>
<rest of program>
```



New Control Flow

```
if <condition>:
    <do something>
elif <other condition>:
    <do something else>
else:
    <do third thing>
<rest of program>
```



Practice

- Write a function called get_weather_report that takes no inputs and returns a str
- It should use the input function to ask the user “What is the weather?” and save that as the local variable weather
- If weather is "rainy" or "cold", it should print "Bring a jacket!"
- If weather is "hot", it should print "Keep cool out there!"
- Otherwise, it should print "I don't recognize this weather."
- return the weather variable
- (Call it with the input “cold” to see what you get!)