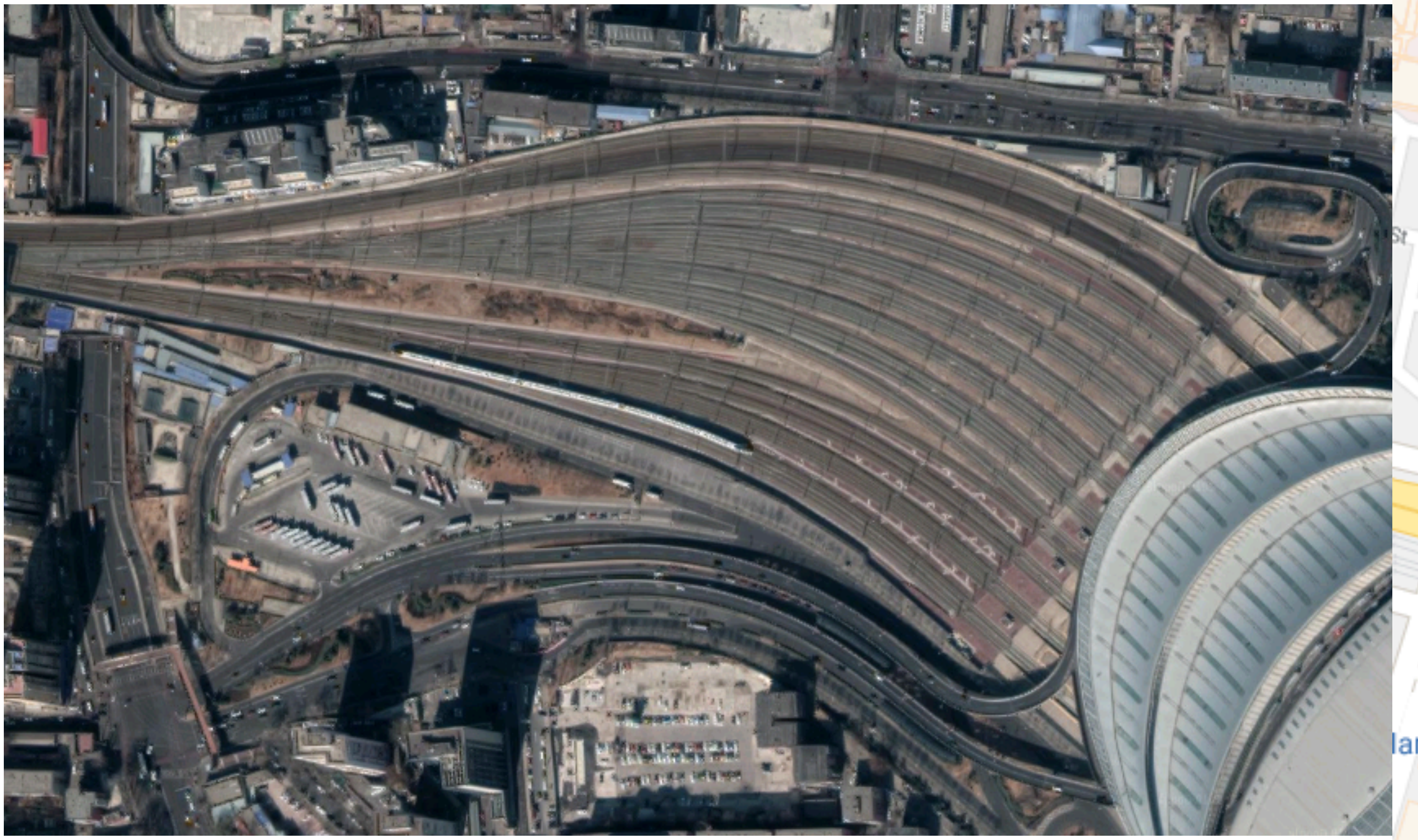
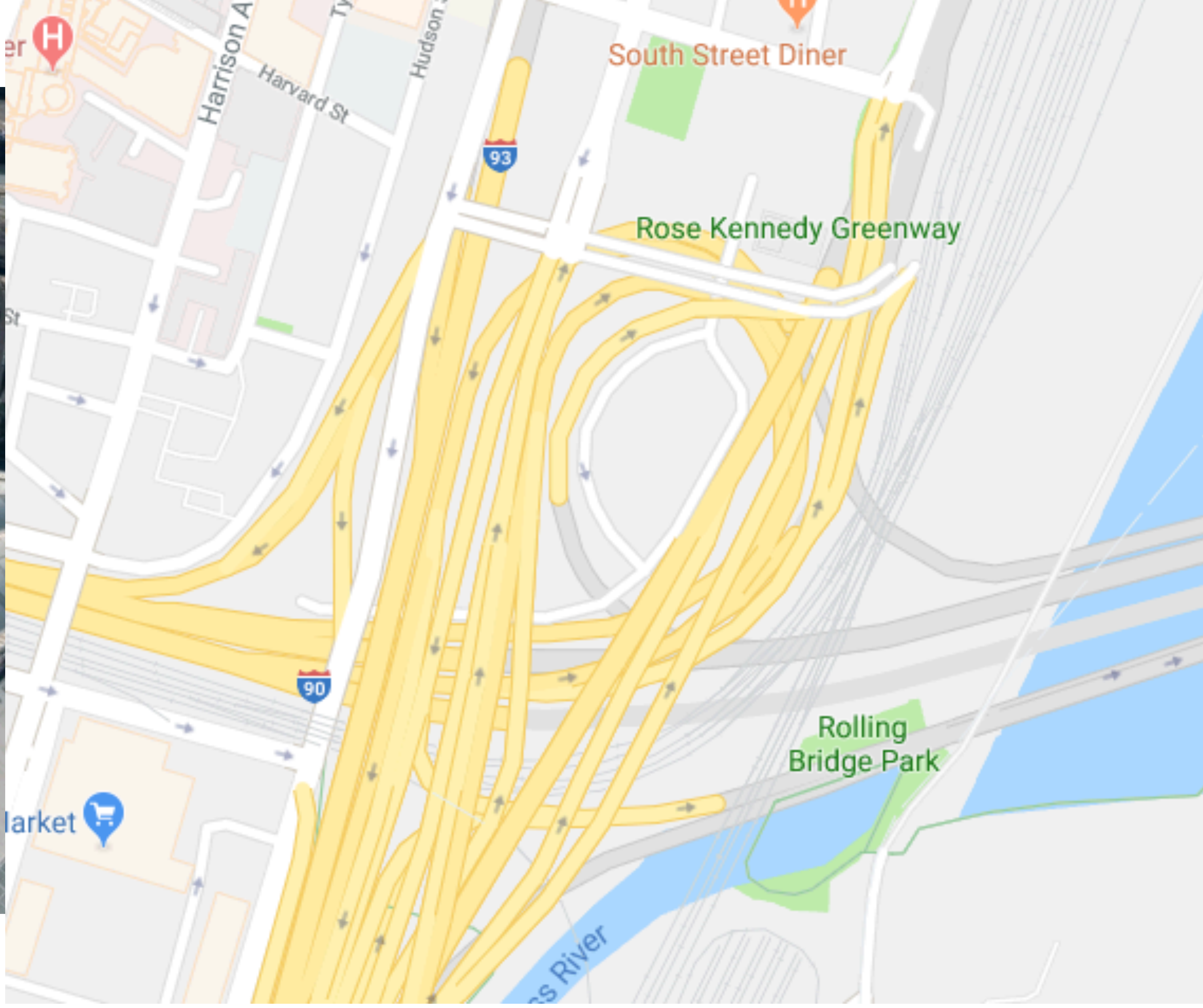




CSCI 141

Lecture 8:
Conditionals, continued:
nested and chained conditionals

Happenings

Tuesday, 4/23 – Peer Lecture Series: React Workshop

—5 pm in CF 162

Tuesday, 4/23 – Artificial Intelligence Presents: Visual Recognition

—6 pm in PH 228

Announcements

- A2 deadline moved to Wednesday of next week
- A3 will be out Monday as scheduled, due the following Wednesday 5/1
- Midterm exam is in 2 weeks: Friday 5/3

Goals

- Know how to use an `if` statement to conditionally execute a block of code.
- Know how to use an `if/else` statement to choose which of two code blocks to execute.
- Understand the behavior of the equality comparison operators (==, !=) on non-numeric types.
- Understand how conditional statements can be nested to make decisions among more than two possibilities.
- Know how to use chained conditionals (`if/elif/else`)

Equality Comparisons

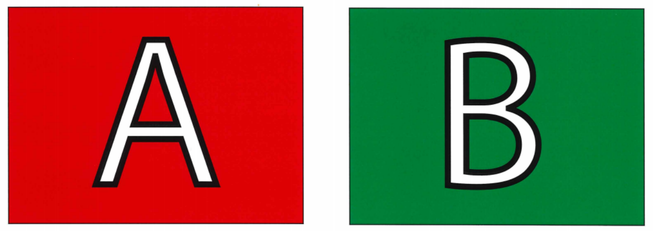
- The operators == and != check whether two values are equal or not.
- Unlike some operators (e.g., //), the concept of equality has meaning for some non-numeric types:

```
4 == 5                  => False
“abc” == “bcd”          => False
“abc” == “abc”          => True
type(4) == type(5)      => True
5.0 == 5                => True
```

Equality Comparisons

Lightning round!

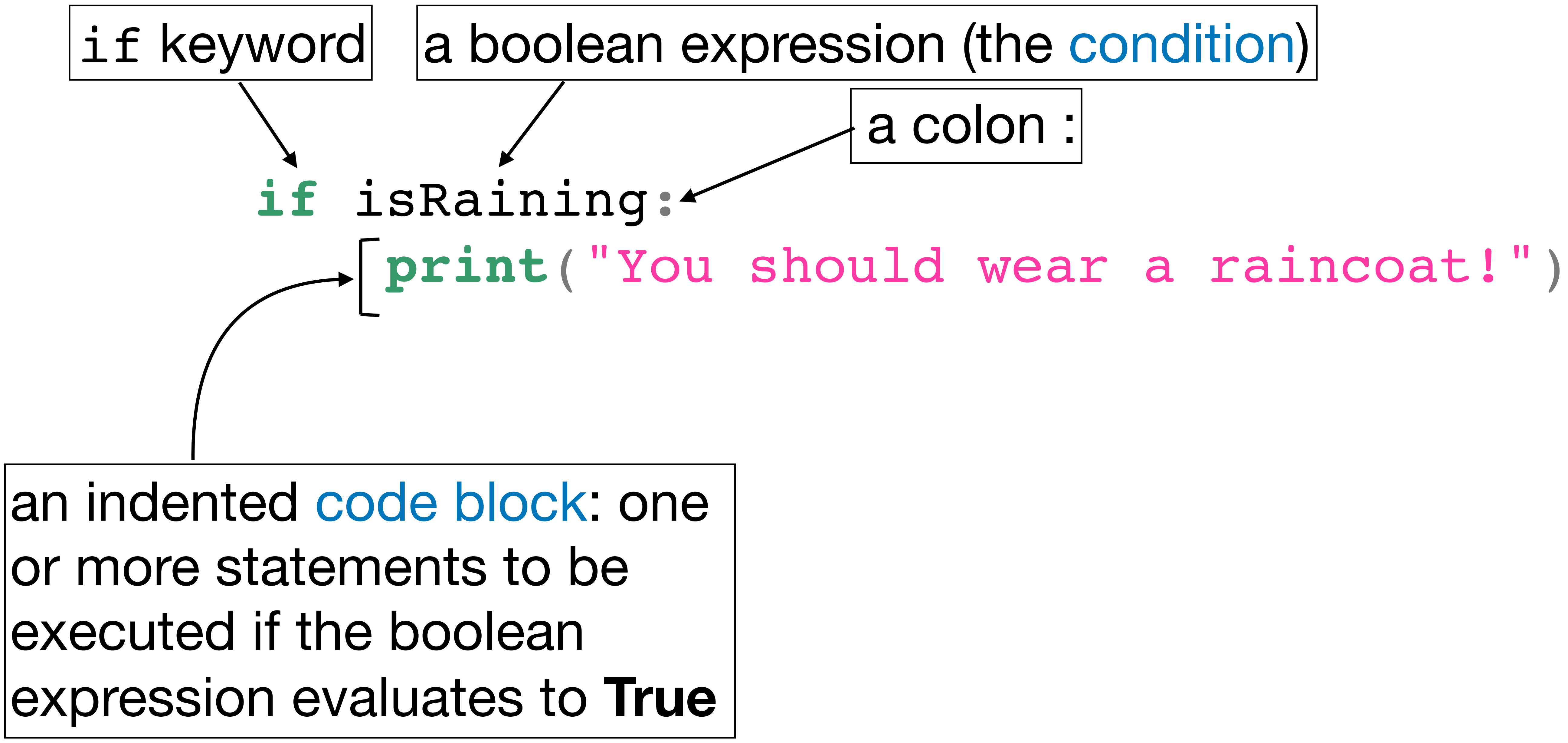
```
10 == 4 + 6   => True

“abc” == “ab” + “c” => True

‘abc’ == “abc” => True

“Scott” == “scott” => False

(4+3 > 5) == (1.0 > 4) => False

int(5.6) != int(5.1) => False
```



A. False

B. True

Last time: `if` statement



if keyword

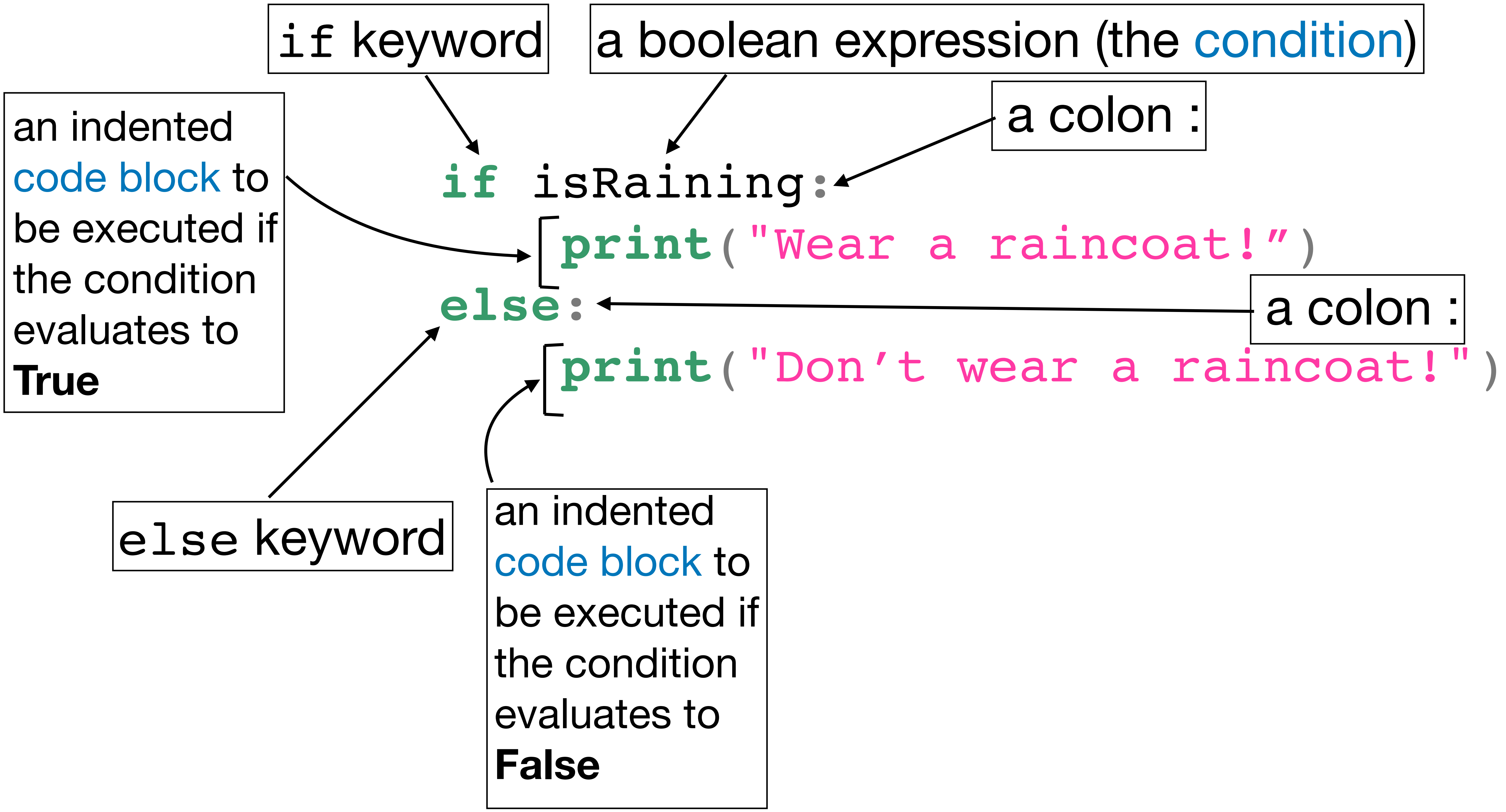
a boolean expression (the condition)

a colon :

```
if isRaining:
    print("You should wear a raincoat!")
```

an indented code block: one or more statements to be executed if the boolean expression evaluates to **True**

Last time: `if` statement with an `else` clause


if keyword
a boolean expression (the condition)
a colon :
an indented code block to be executed if the condition evaluates to **True**
a colon :
else keyword
an indented code block to be executed if the condition evaluates to **False**

```
if isRaining:
    print("Wear a raincoat!”)
else:
    print("Don’t wear a raincoat!")
```

Demo:
Get `isRaining` from the user

Demo:
Get `isRaining` from the user

- Update ifelse.py to ask the user whether it’s raining, and set the isRaining bool accordingly.

Today's Quiz

- 3 minutes

Today’s Quiz

- 3 minutes
- Working with a neighbor: do your answers agree? (2 minutes)

Nested Conditionals

If/else lets you choose between two options.

What if there are more than two possibilities?

```
# assume x and y are numbers
if x < y:
    print("x is less than y")
else:
    if x > y:
        print("x is greater than y")
    else:
        print("x and y must be equal")
```

an indented code block containing one or more statements

Note: the conditions still have to be boolean expressions (i.e., they evaluate to True or False)

the inner `if` statement is the indented code block for the `else` clause of the outer `if` statement.

Nested Conditionals

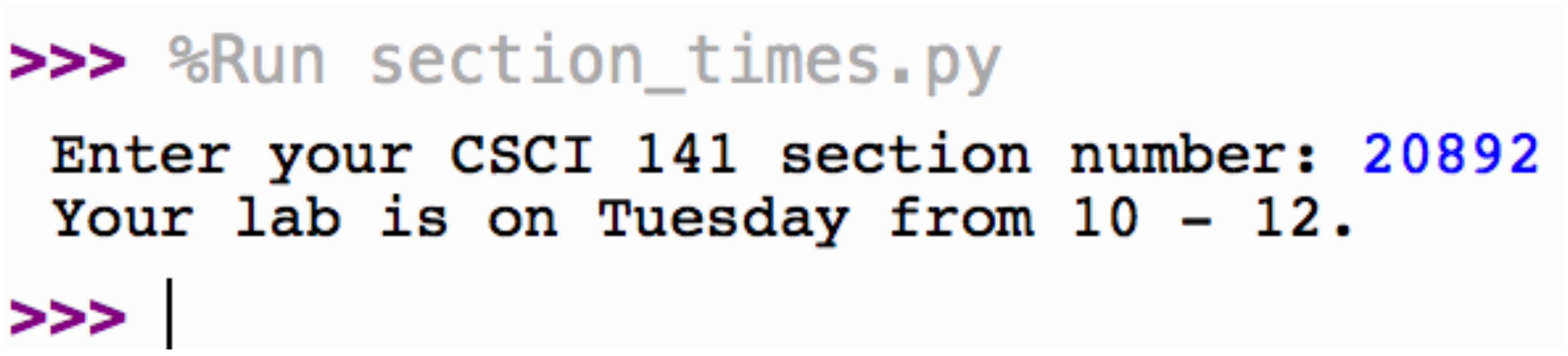
How many comparison operators (<, >) are evaluated by the following code?

A. 0
B. 1
C. 2
D. 3

```
x = 4
y = 5
if x < y:
    print("x is less than y")
else:
    if x > y:
        print("x is greater than y")
    else:
        print("x and y must be equal")
```

Demo

Task: Write a program to ask the user for their 141 section number and print out when their lab section happens.



```
>>> %Run section_times.py
 Enter your CSCI 141 section number: 20892
 Your lab is on Tuesday from 10 - 12.
>>> 
```

Chained Conditionals: Demo

Chained Conditionals: Demo

- sections.py: with chained if/else statements
- sections_elif.py: with if/elif/else
- sections_refactored.py: refactored to set variables then call print once
- sections_refactored.py: with feature to check for conflicts with lab

Chained Conditionals: Syntax

`elif` keyword

```
if isRaining and not isWindy:
    print("Bring an umbrella!")
elif isRaining and isWindy:
    print("Wear a raincoat!")
else:
    print("No rain gear needed!)
```

an indented code block to be executed if:

- **none** of the above conditions was True
- ***and*** this `elif`'s condition is True

(this behaves exactly like nesting an if inside each else)

an indented code block to be executed if the **none** of the above conditions was true

(the else clause is optional)