



# Introduction to Programming

---

Department of Computer Science and Information  
Systems

Lecturers: Tingting Han and Steve Maybank

[sjmaybank@dcs.bbk.ac.uk](mailto:sjmaybank@dcs.bbk.ac.uk)

Autumn 2019 and Spring 2020

**Week 7: if Statement**



# Revision: Relational Operators

---

■ `5 > 4`

# True

■ `5 >= 7`

# False

■ `"9032" < "0AB"`

# False

■ `"ABC" <= "AB"`

# False

■ `48 == 30+18`

# True

■ `"AB" != "ABC"`

# True



# Revision: Lexicographic Ordering

- Given strings  $s_1, s_2$  such that neither string is a prefix of the other, find the least non-negative value of  $i$  such that  $s_1[i] \neq s_2[i]$
- The ordering of  $s_1, s_2$  is the same as the ordering of  $s_1[i], s_2[i]$
- Example,  $i = 4$  and "h" < "n", therefore "alight" < "aligned"

a	l	i	g	n	e	d
a	l	i	g	h	t	
0	1	2	3	4	5	6

What happens if one of the strings  $s_1, s_2$  is a prefix of the other?



# Revision: Boolean Operators

---

- The Boolean operators in Python are `and`, `or` and `not`
- `and`, `or` and `not` are used to make new Boolean expressions, e.g.

```
a or b and c
```

```
(a or b) and c
```

```
a or (b and c)
```

- `5 == 0 and 6 == 0`

```
# False
```

- `5 == 0 or 5 > 4`

```
# True
```

- `not (5 == 0)`

```
# True
```



# Revision: Short Circuit Evaluation

---

- Boolean statements  $x$  and  $y$ ,  $x$  or  $y$  are evaluated **left to right**.
- Evaluation **stops as soon as** the truth value of the statement is determined.
- Example
  - False and  $3/0$ 
    - False
  - True and  $3/0$ 
    - ZeroDivisionError: integer division or modulo by zero
  - True or  $3/0$ 
    - True
  - False or  $3/0$ 
    - ZeroDivisionError: integer division or modulo by zero

# if Statement

- Motivation Example – there is no 13<sup>th</sup> floor!
- How to get the actual floor number?
  - For instance to calculate the distance between the floor and the ground





# if Statement

---

- Example – there is no 13<sup>th</sup> floor!

```
actualFloor = 0    # define variable

if floor > 13 :
    actualFloor = floor - 1
else :
    actualFloor = floor
```



# Alternative if Statement

---

```
actualFloor = 0    # define variable

if floor > 13 :
    actualFloor = floor - 1
else :
    actualFloor = floor
```

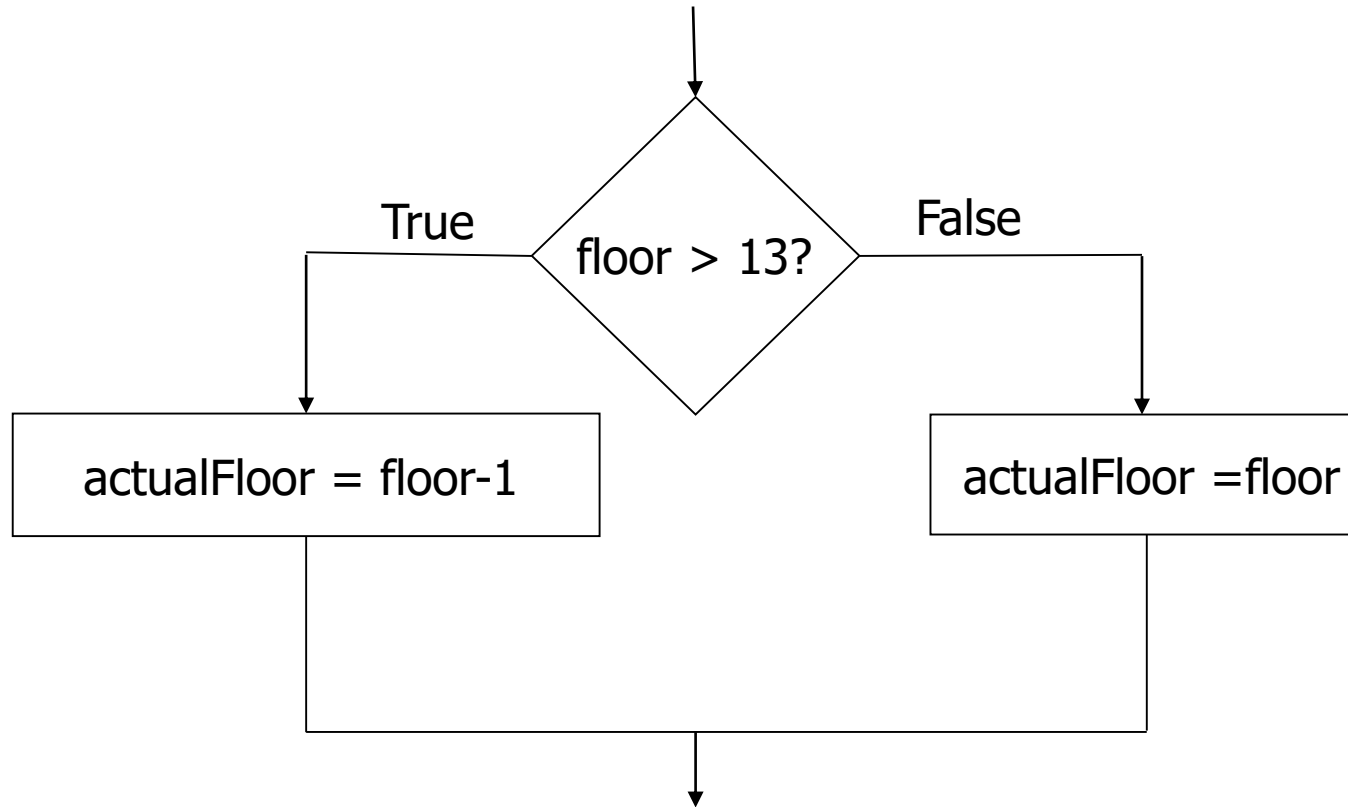
Alternative way :

```
actualFloor = floor
```

```
if floor > 13 :
    actualFloor = actualFloor - 1
```



# Flow Chart





# Parts of an if Statement

---

```
if floor > 13 :          # The condition floor > 13 is True or False
    actualFloor = floor - 1 # execute only if the condition is True
else :
    actualFloor = floor    # execute only if the condition is False
```

# **Align** if and else

# **Indent** the statements in each branch



# Syntax of if Statements

---

- **Version 1**

```
if condition :  
    statements
```

- **Version 2**

```
if condition :  
    statements_1  
else :  
    statements_2
```

# The colon indicates the start of a compound statement



# Compound Statement

---

- A compound statement contains a **header** followed by a **statement block**. Example:

```
if totalSales > 100.0 :    # Colon indicates the header
    discount = totalSales * 0.05    # Block of statements
    totalSales = totalSales - discount
    print("You received a discount of", discount)
```

**# All the statements in the block have the same indentation**



# An if Statement Example

---

```
if floor > 13 :  
    actualFloor = floor - 1  
    print("Actual floor:", actualFloor)  
else :  
    actualFloor = floor  
    print("Actual floor:", actualFloor)
```

- Where could the code be improved?



# Avoid Duplication

---

- Avoid

```
if floor > 13 :
    actualFloor = floor - 1
    print("Actual floor:", actualFloor)
else :
    actualFloor = floor
    print("Actual floor:", actualFloor)
```

- and prefer

```
if floor > 13 :
    actualFloor = floor - 1
else :
    actualFloor = floor
print("Actual floor:", actualFloor)
```



# What's the difference?

---

- Program 1

```
if floor > 13 :
    actualFloor = floor - 1
else :
    actualFloor = floor
print("Actual floor:", actualFloor)
```

- Program 2

```
if floor > 13 :
    actualFloor = floor - 1
else :
    actualFloor = floor
print("Actual floor:", actualFloor)
```

Indentation  
plays an  
important role!

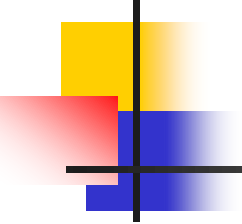


# Example 1

---

- The university bookstore has a **Kilobyte Day** sale every October 24, giving an 8% discount on all computer accessory purchases if the price is less than \$128, and a 16% discount if the price is at least \$128.
- Write a program that asks the cashier for the original price and then prints the discounted price.





# Example 1 - Solution

---

Giving an 8% discount on all computer accessory purchases if the price is less than \$128, and a 16% discount if the price is at least \$128.

```
originalPrice = float(input("Please input the original price:"))
```

```
if originalPrice < 128.0 :
```

```
    discountedPrice = originalPrice * (1 - 0.08)
```

```
else :
```

```
    discountedPrice = originalPrice * (1 - 0.16)
```

```
print("The discounted price is", discountedPrice)
```



# Avoid "Hard-Wiring"

---

```
highPriceThreshold = 128.0
```

```
lowDiscountRate = 0.08
```

```
highDiscountRate = 0.16
```

```
originalPrice = float(input("Please input the original price:"))
```

```
if originalPrice < highPriceThreshold :
```

```
    discountedPrice = originalPrice * (1 - lowDiscountRate)
```

```
else :
```

```
    discountedPrice = originalPrice * (1 - highDiscountRate)
```

```
print("The discounted price is", discountedPrice)
```



# Shipping Costs Example

---

- Shipping costs are \$5 inside the USA except that to Hawaii and Alaska they are \$10. International shipping costs are \$10.
- First design: use a **single if statement** to distinguish between the \$5 cost and the \$10 cost

```
if(country!="USA" or (country=="USA" and (state == "AK" or state == "HI"))):  
    ShippingCost = 10  
else :  
    ShippingCost = 5
```



# Second Design

---

- Shipping costs are \$5 inside the USA except that to Hawaii and Alaska they are \$10. International shipping costs are \$10.
- Separate the three branches:
  - i) inside the USA and in Hawaii or Alaska;
  - ii) inside the USA and not in Hawaii or Alaska;
  - iii) outside the USA.

```
if country == "USA" :
```

```
    if state == "AK" or state == "HI" :
```

```
        shippingCost = 10
```

```
    else :
```

```
        shippingCost = 5
```

```
else :
```

```
    shippingCost = 10
```

nested if

# Richter Scale

The **Richter scale** (or ML scale) rates earthquakes on an exponential scale, so that if an earthquake is rated 1, you can hardly feel it, but an earthquake rated 2 is ten times as strong as an earthquake rated 1, and an earthquake rated 3 is ten times as strong as an earthquake rated 2.

Value	Potential Hazard
10	Extraordinary
9	Outstanding
8	Far-reaching
7	High
6	Noteworthy
5	Intermediate
4	Moderate
3	Minor
2	Low
1	Insignificant



# Richter Scale

---

Value	Effect
8	Most structures fall
7	Many buildings destroyed
6	Many buildings considerably damaged, some collapse
4.5	Damage to poorly constructed buildings

Write a program to print out the Richter scale using if/else statements.



# The if-else Statement

```
if richter >= 8.0 :
    print("Most structures fall")
else:
    if richter >= 7.0 :
        print("Many buildings destroyed")
    else:
        if richter >= 6.0 :
            print("Many buildings considerably damaged")
        else:
            if richter >= 4.5 :
                print("Damage to poorly constructed buildings")
            else :
                print("No destruction of buildings")
```

Value	Effect
8	Most structures fall
7	Many buildings destroyed
6	Many buildings considerably damaged, some collapse
4.5	Damage to poorly constructed buildings



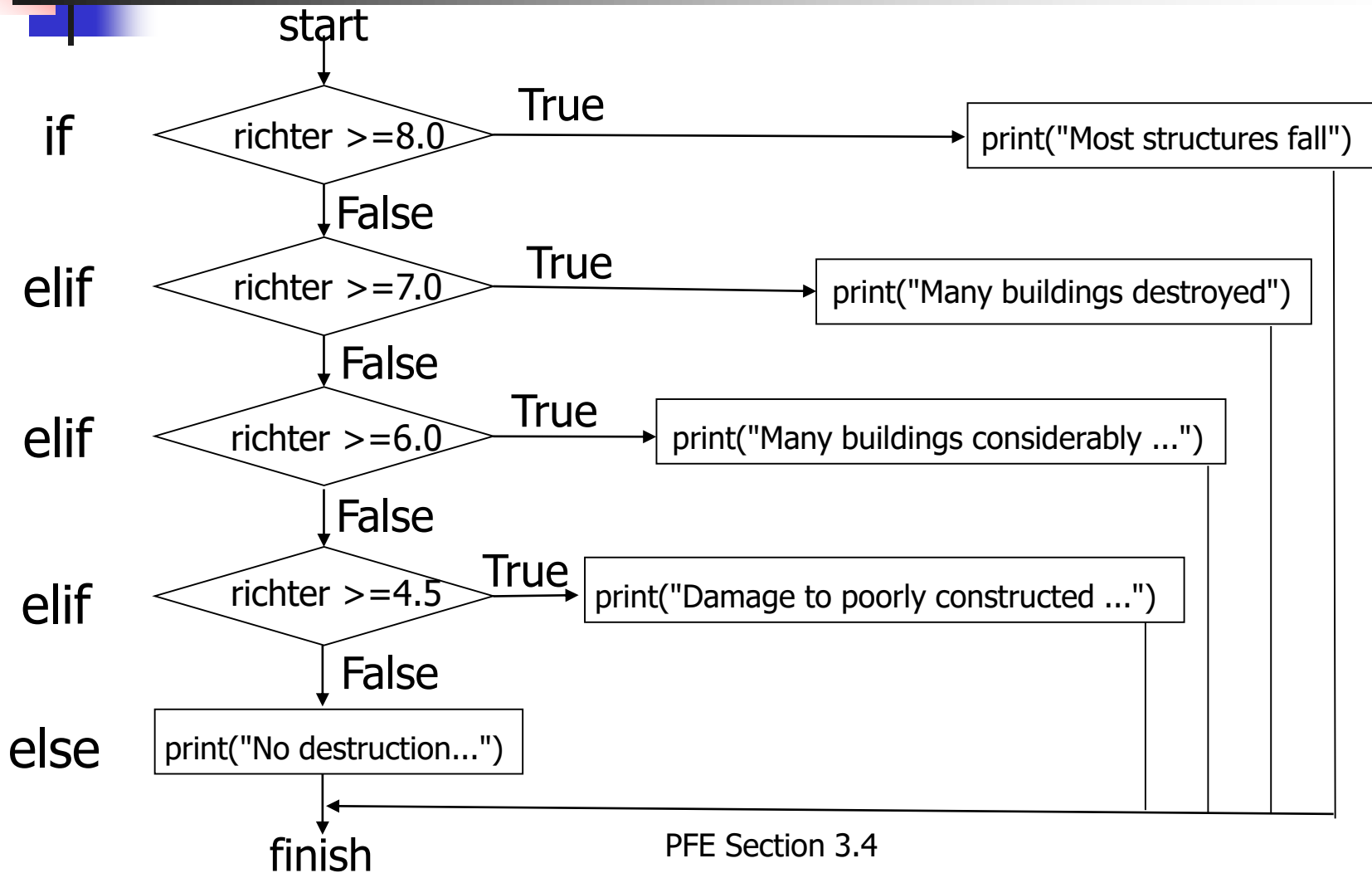
# The elif Statement

---

```
if richter >= 8.0 :  
    print("Most structures fall")  
elif richter >= 7.0 :  
    print("Many buildings destroyed")  
elif richter >= 6.0 :  
    print("Many buildings considerably damaged, some collapse")  
elif richter >= 4.5 :  
    print("Damage to poorly constructed buildings")  
else :  
    print("No destruction of buildings")
```



# The elif Statement Flowchart





# Questions

---

- What happens if the order of the tests is reversed? Is this correct?

```
if richter >= 4.5 :
```

```
    print("Damage to poorly constructed buildings")
```

```
elif richter >= 6.0 :
```

```
    print("Many buildings considerably damaged, some collapse")
```

```
elif richter >= 7.0 :
```

```
    print("Many buildings destroyed")
```

```
elif richter >= 8.0 :
```

```
    print("Most structures fall")
```

```
else :
```

```
    print("No destruction of buildings")
```

Value	Effect
8	Most structures fall
7	Many buildings destroyed
6	Many buildings considerably damaged, some collapse
4.5	Damage to poorly constructed buildings



# Questions

---

- Change the order, change the comparison

```
if richter < 4.5 :
```

```
    print("No destruction of buildings")
```

```
elif richter < 6.0 :
```

```
    print("Damage to poorly constructed buildings")
```

```
elif richter < 7.0 :
```

```
    print("Many buildings considerably damaged, some collapse")
```

```
elif richter < 8.0 :
```

```
    print("Many buildings destroyed")
```

```
else :
```

```
    print("Most structures fall")
```

Value	Effect
8	Most structures fall
7	Many buildings destroyed
6	Many buildings considerably damaged, some collapse
4.5	Damage to poorly constructed buildings



# Example

---

- In a game program, the scores of players A and B are stored in variables `scoreA` and `scoreB`.
- Assuming that the player with the larger score wins.
- Write an `if/elif` sequence that prints out "A won", "B won" or "Game tied".



# Example

---

```
scoreA = int(input(Please enter player A's score))
scoreB = int(input(Please enter player B's score))

if scoreA > scoreB :
    print("A won")
elif scoreB > scoreA :
    print("B won")
else:
    print("Game tied")
```



# Input Validation

---

- Check user supplied input to see if it has the correct form.
- Example: in the elevator simulation let `maxFloor` be the largest floor number. The following inputs are illegal.
  - i) 13
  - ii) 0 or a negative number (in the USA)
  - iii) any number  $> \text{maxFloor}$
  - iv) Any input not a sequence of digits



# Error Messages

---

```
floor = int(input("Floor: "))
```

```
# if the input is non digital then there is a run time exception
```

```
# and the program terminates.
```

```
if floor == 13 :
```

```
    print("Error: there is no 13th floor")
```

```
if floor <= 0 or floor > maxFloor :
```

```
    print("Error: the floor must be between 1 and", maxFloor)
```



# Example Code

---

```
floor = int(input("Floor: "))
```

← input

```
if floor == 13 :
```

```
    print("Error: there is no 13th floor")
```

← input validation

```
elif floor <= 0 or floor > maxFloor :
```

```
    print("Error: the floor must be between 0 and", maxFloor)
```

```
else :
```

```
    actualFloor = floor
```

```
    if floor > 13 :
```

```
        actualFloor = actualFloor - 1
```

← rest of the program





# Exercise

---

```
print("Buy pasta")
if weather == "sunny":
    print("Buy salad")
    print("Buy olives")
print("Buy wine")
```

What items will be bought if

- 1) it is cloudy?
- 2) it is sunny?