

# Lecture 13

Topics to be covered:

Chapter 9:

- Section 9.4 More if-else
- Section 9.5 Boolean operators and expressions
- Section 9.6 Membership and identity operators

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade >= 90:  
    if grade < 93:  
        print("that's an A-")  
    elif grade >= 97:  
        print("that's an A+")  
    else:  
        print("that's an A")  
else:  
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade = 78
if grade >= 90:
    if grade < 93:
        print("that's an A-")
    elif grade >= 97:
        print("that's an A+")
    else:
        print("that's an A")
else:
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

if grade = 78

```
→ if grade >= 90:  
    if grade < 93:  
        print("that's an A-")  
    elif grade >= 97:  
        print("that's an A+")  
    else:  
        print("that's an A")  
else:  
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

if grade = 78

```
if grade >= 90:  
    if grade < 93:  
        print("that's an A-")  
    elif grade >= 97:  
        print("that's an A+")  
    else:  
        print("that's an A")  
→else:  
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade = 78
if grade >= 90:
    if grade < 93:
        print("that's an A-")
    elif grade >= 97:
        print("that's an A+")
    else:
        print("that's an A")
else:
    →print("not an A grade")
```

if grade = 78

not an A grade

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade = 95
if grade >= 90:
    if grade < 93:
        print("that's an A-")
    elif grade >= 97:
        print("that's an A+")
    else:
        print("that's an A")
else:
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

if grade = 95

```
→ if grade >= 90:  
    if grade < 93:  
        print("that's an A-")  
    elif grade >= 97:  
        print("that's an A+")  
    else:  
        print("that's an A")  
else:  
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade = 95
if grade >= 90:
→if grade < 93:
    print("that's an A-")
elif grade >= 97:
    print("that's an A+")
else:
    print("that's an A")
else:
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade = 95
if grade >= 90:
    if grade < 93:
        print("that's an A-")
→elif grade >= 97:
    print("that's an A+")
else:
    print("that's an A")
else:
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade = 95
if grade >= 90:
    if grade < 93:
        print("that's an A-")
    elif grade >= 97:
        print("that's an A+")
    else:
        print("that's an A")
else:
    print("not an A grade")
```

## 9.4 More if-else

### Nested if-else statements

A branch's statements can include any valid statements, including another if-else statement, which are known as *nested if-else statements*.

```
if grade = 95  
if grade >= 90:  
    if grade < 93:  
        print("that's an A-")  
    elif grade >= 97:  
        print("that's an A+")  
    else:  
        → print("that's an A")  
else:  
    print("not an A grade")
```

that's an A

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if num = 12?

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if `num = 12`?

A  
B

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if num = 1?

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if num = 1?

B

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if num = -1?

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if num = -1?

C

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if num = -12?

## 9.4 More if-else

### Multiple if statements

Consider the following code fragment:

```
if num >= 10:  
    print("A")  
  
if num >= 0:  
    print("B")  
  
if num < 0:  
    print("C")  
  
if num < -10:  
    print("D")
```

What would the program output if num = -12?

C  
D

## 9.4 More if-else

In-class Activity

See exercises 1-2

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

A **Boolean** refers to a value that is either **True** or **False**.  
These two are constants in Python.

- we can assign a **Boolean value** by specifying **True** or **False**,  
`x = True`
- an expression can evaluate to a **Boolean value**  
`y > 10`

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Recall our logical operators/connectives!

^ conjunction and

∨ disjunction or

¬ negation not

→ implication decision structure!

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Recall our logical operators/connectives!

- ^ conjunction      and
- ∨ disjunction      or
- ¬ negation          not

**Examples:** assume that  $a = 8$  and  $b = 3$

$(a > 10) \text{ and } (b < 5)$

$(a > 10) \text{ or } (b < 5)$

not  $(a > 10)$

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Recall our logical operators/connectives!

- ^ conjunction and
- ∨ disjunction or
- ¬ negation not

**Examples:** assume that  $a = 8$  and  $b = 3$

$(a > 10)$  and  $(b < 5)$  is **False**

$(a > 10)$  or  $(b < 5)$  is **True**

not  $(a > 10)$  is **True**

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Consider the following code fragment:

```
if letter == 'a' or letter == 'b':  
    print("Help!")  
  
elif letter == 'c' or letter == 'd':  
    print("We are in trouble!")  
  
else:  
    print("We are good!")
```

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Consider the following code fragment:

```
if letter == 'a' or letter == 'b':  
    print("Help!")  
  
elif letter == 'c' or letter == 'd':  
    print("We are in trouble!")  
  
else:  
    print("We are good!")
```

if letter = ‘a’, then we will get:

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Consider the following code fragment:

```
if letter == 'a' or letter == 'b':  
    print("Help!")  
  
elif letter == 'c' or letter == 'd':  
    print("We are in trouble!")  
  
else:  
    print("We are good!")
```

if letter = ‘a’, then we will get:

Help!

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Consider the following code fragment:

```
if letter == 'a' or letter == 'b':  
    print("Help!")  
  
elif letter == 'c' or letter == 'd':  
    print("We are in trouble!")  
  
else:  
    print("We are good!")
```

if `letter = 'c'`, then we will get:

## 9.5 Boolean operators and expressions

### Booleans and Boolean operators

Consider the following code fragment:

```
if letter == 'a' or letter == 'b':  
    print("Help!")  
  
elif letter == 'c' or letter == 'd':  
    print("We are in trouble!")  
  
else:  
    print("We are good!")
```

if letter = ‘c’, then we will get:

We are in trouble!

## 9.5 Boolean operators and expressions

In-class work

Exercises 2-5

## 9.6 Membership and identity operators

Membership operators: `in/not in`

Quite often we need to check if a value can be or cannot be found within a container, such as a list or dictionary.

`in` and `not in` operators, known as *membership operators*, can help us!

## 9.6 Membership and identity operators

### Membership operators: `in/not in`

Quite often we need to check if a value can be or cannot be found within a container, such as a list or dictionary.

`in` and `not in` operators, known as *membership operators*, can help us!

#### Example:

```
num = int(input("Enter an integer:"))
myContainer = [1,2,3,4,5,6,7]
```

```
if num in myContainer:
    print("Found it! It is in myContainer!")
```

```
else: print("Nope. It is not in myContainer.")
```

## 9.6 Membership and identity operators

Membership operators: `in/not in`

### Example:

```
name = int(input("Enter a name:"))
MyNamesContainer = {
    "Maria": 23,
    "Anna": 19,
    "Jack": 5,
    "Alex": 12,
    "John": 18}

if name in myNamesContainer:
    print("Found it! It is corresponds to",
        MyNamesContainer[name])
else: print("No such name in the container.")
```

## 9.6 Membership and identity operators

Membership operators: `in/not in`

### Example:

```
name = int(input("Enter a name:"))
MyNamesContainer = {
    "Maria": 23,
    "Anna": 19,
    "Jack": 5,
    "Alex": 12,
    "John": 18}
```

```
if name in myNamesContainer:
    print("Found it! It is corresponds to",
          MyNamesContainer[name])
else: print("No such name in the container.")
```

Note that the keys are  
matched, not the values!

## 9.6 Membership and identity operators

### Identity operators: `is`/`is not`

Sometimes we want to determine whether two variables are the same object.

`is` and `is not` operators, known as *identity operators*, can help us out!

Identity operators return `True` only if the operands reference the same object (they do not compare object's values).

## 9.6 Membership and identity operators

Identity operators: `is/is not`

### Example:

```
myContainer = [1,2,3,4,5,6,7]
otherContainer = [9,8,7,6,5,4,3,2,1]
```

```
a = myContainer
b = otherContainer
a = b
```

```
if a is myContainer:
    print("a is myContainer!")
elif a is otherContainer:
    print("a is otherContainer!")
else: print("I have no idea that is a!")
```

## 9.6 Membership and identity operators

In-class work

Exercises 6-8