COMP1021
Introduction to Computer Science

Using Logic

David Rossiter

Outcomes

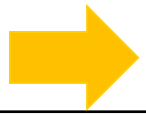
- After completing this presentation, you are expected to be able to:

  1. Understand how to use logic operators *and*, *or*, and *not*

Comparing Things

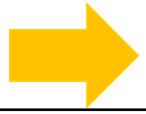
- When you do a comparison, the result is either True or False

```
x = 100
result = x > 50
print(result)
```



```
True
```

```
x = 10
result = x > 50
print(result)
```



```
False
```

Using Logical Operators

- You use the comparison operators (<, <=, >, >=, == and !=) to compare two values
- You can also use *logical operators*, also called *Boolean operators*:

a `and` *b*	if both condition *a* and condition *b* are True, the result is True; otherwise, it's False
a `or` *b*	if either condition *a* or condition *b* is True, the result is True; otherwise, it's False
`not` *a*	if *a* is True, then the result is False; if *a* is False, then the result is True

The opposite

Summary

- Here is a summary of the input and output:

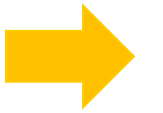
a	***b***	***a and b***	***a or b***	***not a***
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

And

- `and` – the result is True if both inputs are True
  otherwise the result is False
  - Let's use Python to check whether someone is a suitable girlfriend/boyfriend
  - In this example, we need **both** of the two inputs to be true for the person to be suitable

```
funny = False
friendly = False
suitable_partner = funny and friendly
print(suitable_partner)
```

The inputs



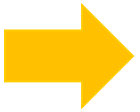
`False`

The result

```
funny = False
friendly = True
suitable_partner = funny and friendly
print(suitable_partner)
```

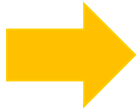
False

```
funny = True
friendly = False
suitable_partner = funny and friendly
print(suitable_partner)
```



False

```
funny = True
friendly = True
suitable_partner = funny and friendly
print(suitable_partner)
```



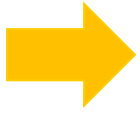
True

Or

- `or` – the result is False if both inputs are False
  otherwise the result is True

- Let's revise the previous code so only one of the two inputs has to be true for the person to be suitable

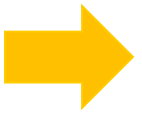
```
funny = False
friendly = False
suitable_partner = funny or friendly
print(suitable_partner)
```

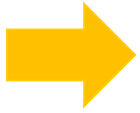


```
False
```

```
funny = False
friendly = True
suitable_partner = funny or friendly
print(suitable_partner)
```

```
True
```

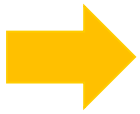
```
funny = True
friendly = False
suitable_partner = funny or friendly
print(suitable_partner)
```

```
True
```

```
funny = True
friendly = True
suitable_partner = funny or friendly
print(suitable_partner)
```

```
True
```

Not

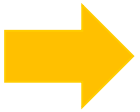
- `not` – the output is the opposite of the input

```
very_clean = False
need_to_shower = not very_clean
print(need_to_shower)
```



```
True
```

```
very_clean = True
need_to_shower = not very_clean
print(need_to_shower)
```



```
False
```

Simpler Code

```
if funny == True and friendly == True:
    suitable_partner = True
else:
    suitable_partner = False
```

- The code shown above works fine
  but a good programmer would write this,
  which does the same:

```
suitable_partner = funny and friendly
```

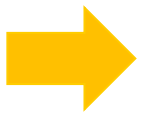
Multiple Inputs

- Here's an example of multiple inputs

```
funny = True
friendly = False
wealthy = True
has_car = True
cute = False
suitable_partner = funny and friendly and \
    wealthy and has_car and cute
print(suitable_partner)
```

In this example all of these have to be True for the result to be True

This tells Python the code continues on the following line

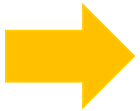


```
False
```

Multiple Inputs

- Here's another example

```
scary_virus = True
need_internet = True
live_on_campus = False
go_to_HKUST = (live_on_campus or need_internet) \
        and not scary_virus
print(go_to_HKUST)
```



```
False
```

- The logic is: go to HKUST if you live on campus
  or you need internet
  but if there is a scary virus don't go

Converting Inputs into True or False

- Sometimes the inputs are not True or False, they are something else
- You may have to ‘convert’ the inputs into True or False before you can use logical operators
- The example on the next slide ‘converts’ input from the user into True or False, then uses a logical operator

if the user enters yes then `response` *contains True*

if the user enters anything except yes then `response` *contains False*

```
response = input("Are you alive? (yes/no)")
response = response == "yes"
print("response =", response)
print("Are you dead?")
print("The answer is:", not response)
```

```
Are you alive? (yes/no) yes
response = True
Are you dead?
The answer is: False
```

```
Are you alive? (yes/no) no
response = False
Are you dead?
The answer is: True
```