Conditionals and booleans

Idea: Express different options

Real Life Examples:

If it’s cold, wear a sweater. Otherwise, wear a t-shirt

If you’re happy and you know it, clap your hands

Requires two language features

booleans - allow us to express whether something is true or false

conditionals - statements that do one thing or another
Booleans

bool data type (similar to int, str, and float)

possible values: True or False

booleans are often used as flags or flag variables

Metaphor: setting a ref flag if you see a problem

Ex. isHappy = True

   isCold = temp < 45 # isCold = True if temp < 45; False otherwise
Relational operators return booleans

a < b ← returns True if a is less than b; otherwise returns False

a <= b ← returns True if a is less than or equal to b; otherwise returns False

a > b ← returns True if a is greater than b; otherwise returns False

a >= b ← returns True if a is greater than or equal to b; otherwise returns False

a == b ← returns True if a is equal to b; otherwise returns False

WARNING: Don’t confuse == with assignment operator =

a != b ← returns True if a is not equal to b, otherwise returns False
Conditional statements

Idea: make a decision based on a boolean statement, e.g. conditional

if <condition>:

<body> runs if <condition> is true

<body>
Conditional statement example

temp = int(input("Enter a temperature: "))

if temp < 45:
    print("Wear a coat for goodness sakes!")
Conditional two-way statements

Idea: specify what happens when a condition is True and False

if <condition>:
    <body1>
else:
    <body2>

<body1> runs if <condition> is true
<body2> runs if <condition> is false
Conditional two-way statement example

temp = int(input(“Enter a temperature: ”))

if temp < 45:
    print(“Wear a coat for goodness sakes!”)
else:
    print(“Don’t wear a coat”)
Example - even or odd

Write a program that checks if a given integer is even or odd

Hint: Use % to check whether the number is even or odd

$ python3 evenodd.py
Enter an integer: 11
11 is odd

$ python3 evenodd.py
Enter an integer: 12
12 is even
Conditional multi-way statements

Idea: Handle multiple possibilities

if <condition1>:
    <body1>
else:
    <body3>

elif <condition2>:
    <body2>

<body1> runs if <condition1> is true
<body2> runs if <condition2> is true
<body3> runs if neither <condition1> nor <condition2> is True
Conditional multi-way statements

Idea: Handle multiple possibilities

if <condition1>:
    <body1>
elif <condition2>:
    <body2>
...
else:
    <body3>

You can have as many conditions as you like!

Each condition is checked in order

Only one body is ever executed
temp = int(input("Enter a temperature: "))

if temp < 45:
    print("Wear a coat for goodness sakes!")

elif temp < 65:
    print("Wear a light jacket")

else:
    print("Wear a t-shirt")
Exercise - Compute a grade

Write a program, grade.py, that outputs a grade given a value between 0 and 100.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90-100</td>
</tr>
<tr>
<td>B</td>
<td>80-89</td>
</tr>
<tr>
<td>C</td>
<td>70-79</td>
</tr>
<tr>
<td>D</td>
<td>60-69</td>
</tr>
<tr>
<td>F</td>
<td>0-59</td>
</tr>
</tbody>
</table>

$ python3 grade.py
Enter a value between 0 and 100: 46
Grade: F

$ python3 grade.py
Enter a value between 0 and 100: 78
Grade: C
Nested Blocks

Statements with the same indentation are called **blocks**

Blocks can be nested inside of other blocks

- if statements in other if statements
- if statements in a loop
- loops within if statements
- loops within loops
Exercise - Do these produce the same input?

if temp >= 60:
    print("No coat needed")
if temp >= 40:
    print("Spring jacket")

if temp >= 60:
    print("No coat needed")
elif temp >= 40:
    print("Spring jacket")

if temp >= 40:
    if temp >= 60:
        print("No coat needed")
    else:
        print("Spring jacket")
Logical operators

and - True if both operands are True

   Ex. If you are a citizen and over 18, you can vote

or - True if either operand is true

   Ex. If you have a ticket or you are a member, you can enter

not - Takes the opposite value

   Ex. You are not wrong
Logical operators examples

if citizen == “Yes” and age >= 18:
    print(“You can vote”)

if membership == “Yes” or hasTicket == True:
    print(“You may enter”)

isWrong = False
if not isWrong:
    print(“You are right!”)
For variables $x$ and $y$ of type bool

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x$ and $y$</th>
<th>$x$ or $y$</th>
<th>not $x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>
Logical operator precedence

()  

Relational operators (> , < , ==, etc)  

not  

and  

or  

Use () to make code clearer!!!