

COMPSCI 111 / 111G

*Mastering Cyberspace:
An introduction to practical computing*

Programming with Python

Revision

Ask the user to enter a number of centimetres. The program should print out the number of metres and centimeters that are equal to the input.

```
#Author: Andrew Luxton-Reilly
#Date: 7/05/06

#Ask the user to enter the number of centimetres
cm = input("Please enter the number of centimetres: ")

#Calculate the number of metres and centimetres
m = cm / 100
cm = cm % 100

#Print the output to the user
print m, "metres and", cm, "centimeters"
```

Exercise

Write a program that asks the user to enter a number of feet and a number of inches. The program should convert the length to a number of meters (with decimal points) the length using metres as the units.

Note: 1 inch = 2.54 centimetres
1 foot = 12 inches

if statements

```
if <condition>:
    statements to execute if condition is true
else:
    statements to execute if condition is false
```

A colon appears in two places

- after the <condition>
- after the keyword `else`

The <condition> should be a boolean expression

- Evaluates to either true or false

Statements to be executed are organised as blocks

- Statements should be indented to the same amount
- Any number of statements can be executed in a block

Comparison operators

Used to compare two things

- Evaluates to a boolean value
- Commonly used in a condition

Meaning	Operator	Example
Less than	<	a < b
Less than or equal to	<=	a <= b
Greater than	>	a > b
Greater than or equal to	>=	a >= b
Equal to	==	a == b
Not equal to	!=	a != b

Example

Write a program that asks the user to enter a number. The program should determine if the number is odd or even, and print out an appropriate message.

```
#Author: Andrew Luxton-Reilly
#Date: 7/05/06

#Ask the user to enter a number
number = input("Please enter a number: ")

#Determine if the number is odd or even
if (number % 2 == 0):
    print "You entered the number", number
    print "That number is even"
else:
    print "You entered the number", number
    print "That number is odd"
```

Exercise

Write a program that asks the user to enter their age, and then prints out a ticket based on their age:

- Child tickets (below the age of 12) cost \$5.00
- Adult tickets (age 12 or above) cost \$9.99

The tickets should look like:

```
*****
Child Ticket
Price: $5.00
*****
```

```
*****
Adult Ticket
Price: $10.00
*****
```

Logical operators

Used to combine boolean expressions

- And
- Or
- Not

Meaning	Operator	Example
Logical AND	and	number >= 1 and number <= 2
Logical OR	or	number == 1 or number == 2
Logical NOT	not	not(number < 1 or number > 2)

Example

Write a program that asks the user to enter a number between 1 and 10 (inclusive). The program will print out "Correct" if the number is in the range and "Incorrect" if the number is outside the range.

```
#Author: Andrew Luxton-Reilly
#Date: 7/05/06

#Ask the user to enter a number
number = input("Please enter a number (1-10): ")

#Determine if the number is within the range
if (number >= 1 and number <= 10):
    print "Correct"
else:
    print "Incorrect"
```

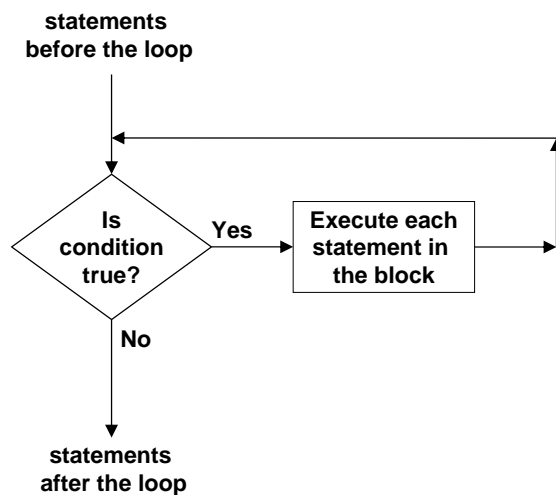
Loops

```
while <condition>:
    statements to execute while the condition is true
```

Loops are extremely common in programming

- If the condition is true, then the statements in the block are executed
- After the block of statements is executed, we return to the while keyword and check condition again
- If the condition is true, then the statements in the block are executed
- ...
- The process continues until the condition is false
- When the condition is false, the statements in the block are skipped and execution continues with any statements that follow the loop

Flow Diagram



Example

Keep asking the user to enter a number bigger than 0 until they enter a number bigger than 0.

```
#Author: Andrew Luxton-Reilly
#Date: 7/05/06

#Set the number to be -1 to start
number = -1

while (number <= 0):
    number = input("Please enter a positive number: ")

print "Well done"
```

Exercise

Write a program that repeatedly asks the user to enter a number between 1 and 10 until the number they enter falls inside that range.

Using a loop to count

Extremely common in programming

- Loop a specified number of times
- Use the same pattern of code

```
#Loop from start to end
start = 10
end = 20

#Use a variable to keep track of the loop
count = start

while (count <= end):
    #do something
    print count, "*", count, "=", count ** 2
    #increase value of count
    count = count + 1
```

Example

Print out all the numbers between 1 and 10

- Use a variable to keep track of the numbers
- Start the loop with number = 1
- Stop the loop after the number reaches 10

```
#Author: Andrew Luxton-Reilly
#Date: 7/05/06

#Use a variable to keep track of the loop
count = 1

#Keep doing the loop until we have reached 10
while (count <= 10):
    print count
    count = count + 1
```

Exercise

Write a program that repeatedly asks the user to enter a password until they enter the text "kiwi". The program will keep track of the number of incorrect attempts. Sample output is shown below:

```
Attempt 1 - Enter the password: one
Attempt 2 - Enter the password: fish
Attempt 3 - Enter the password: two
Attempt 4 - Enter the password: fish
Attempt 5 - Enter the password: red
Attempt 6 - Enter the password: fish
Attempt 7 - Enter the password: blue
Attempt 8 - Enter the password: fish
Attempt 9 - Enter the password: kiwi
Correct.
```