COMPSCI 111 / 111G

Mastering Cyberspace:
An introduction to practical computing

Programming with Python

Programming Languages

A programming language is a formal language that specifies how to perform a computational task

Many programming languages exist

- Fortran
- Visual Basic
- C, C++, C#
- Java
- Python

Translating Code

High-level languages

- Easy for humans to understand
- Too complex for a CPU
- Need to translate to a lower level language

Compiler

- Translates entire file at once
- Entire file can be understood by the CPU

Interpreter

- Runs the code while it is being translated
- Translates a line, runs the code, translates a line, etc.

Statements

A program consists of a series of instructions

- Instructions in a program are called statements
- Computer executes them in the order they appear

Must be precise

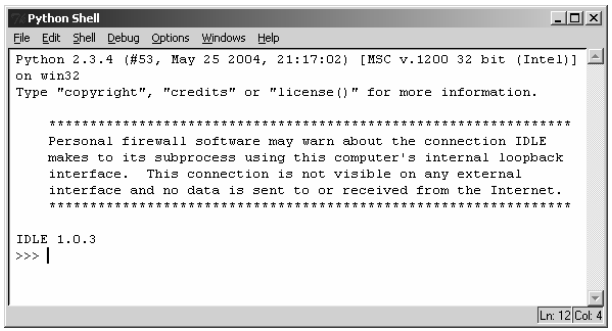
- Computer does what you say, not what you mean

Using IDLE

Integrated Development Environment

- Text Editor
- Interpreter to run the code
- May include other tools to help a programmer

IDLE is an IDE for Python



Running Python instructions

Interactive Interpreter

- Allows you to type statements directly at the prompt
- Statement is executed when you hit <Enter>
- Very useful for experimentation
- Good for learning

Running a Script

- Type a sequence of statements into a file
- Save the file with the file extension .py
- Running the program executes each statement in turn

Example: Hello World

The first program you normally write

- Traditional
- Prints out the message "Hello World"

Python

```
print "Hello World"
```

Java

```
public class MyProgram{
   public static void main(String[] args){
      System.out.println("Hello World");
   }
}
```


Comments

Comments

- Used to make comments to human readers
- Ignored by the computer
- Start with hash sign (#), ignores everything until end of line
- Always start a program with comments describing the Author and Date.

```
#Author: Andrew Luxton-Reilly
#Date: 7/05/06
#Purpose: Show the use of comments

print "Hello"       #Hello, Hello
print "Is there anybody in there"
print "Just nod if you can hear me"
print "Is there anyone at home"
```

Types of information

Strings

- Sequence of characters
- Plain text (ASCII or Unicode)
- Enclosed in quote marks
- E.g. "Hello", "Goodbye"

Integers

- Numbers without a decimal point
- E.g. -100, 0, 45

Floating-point numbers

- Numbers with a decimal point
- E.g. -1.00002, 0.0, 4.5, 45.0,

Printing output

Use the `print` statement

```
print "Hello"
print 34.9
print 2
```

Printing more than one thing on a single line

- Separate each thing with a comma
- Single space used between different things in the output

```
print "Hello", "World"
print "The year is 2006"
print "The year is", 2006
```


Exercise

What is the output produced by the following program: (Use the ^ symbol to represent a space)

```
print "This", "is"
print "a program that has"
print 3, "lines"

print 1,2,3,4
print "1,2,3,4"
print "1234", 1,2
print "1",2,3,"4"
```


Arithmetic operators

Used to perform calculations between numeric values

Operation	Symbol
Exponent	`**`
Multiply	`*`
Divide	`/`
Remainder	`%`
Add	`+`
Subtract	`-`

Operators

```
print 2 + 3
print 3 – 4
print 4 * 3
print 3 / 2
print 7 % 3
```

```
print 7 / 2
print 7.0 / 2
```

```
print 7 % 2
print 7 % 3
print 4 % 7
```


String operations

Concatenation

- Join two strings together
- +
- `"Hello" + " " + "World"`

Repetition

- Repeat a string multiple times
- *
- `"Hello World" * 3`

Exercise

What is the output produced the following program?

```
print 1 + 2 + 3
print "1" + "2" + "3"
print 1 * 2
print "1" * 2

print 2 / 3
print 1.0 + 3 / 2
print 1 + 3 / 2.0

print 2 % 3
print 20 % 7
print 20 / 7
```


Variables

Variable in mathematics

- Symbol that acts as a placeholder
- Same value used everywhere the symbol is used
- Impossible to have

```
x = 9
x = 5
```
← **impossible in mathematics**

Variable in computer science

- Name of a box
- Used to store different values at different times

```
x = 9
x = 5
```
← **valid in computer science**

Assignment statement

Storing a value in a variable

- Assigning a value to the variable
- Equals sign

```
age = 23
sizeOfFruitbat = 56
numberOfWeasels = 17
```

Valid name of a variable

- Start with a lower case letter
- Each subsequent word starts with upper case
- May contain numbers (or the underscore)
- May not be a keyword that means something special in Python

Using variables

```
height = 10
width = 20
area = height * width

print "Area =", area
```

```
age = 21
print "Age =", age
age = age + 1
print "Next year my age =", age
```


Exercise

What is the output produced by the following program?

```
a = 17
b = 5

print a
print b

temp = a
a = b
b = temp

print a
print b
```


Reading input from the user

Reading a number

- `input( prompt )`
- Store the input in a variable

```
age = input("Enter your age: ")
```

Reading a string

- `raw_input( prompt )`
- Store the input in a variable

```
name = raw_input("Enter your name: ")
```

Examples

Write a program that converts NZD to USD

- Currently 1 NZD = 0.6409 USD

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

#Ratio of NZD to USD
currencyRatio = 0.6409 # 1 NZD = 0.6409 USD

#Ask the user to enter the NZD value
nzd = input("Please enter the dollar value (NZD): ")

#Calculate the amount of USD
usd = nzd * currencyRatio

#Print the output to the user
print nzd, "NZ dollars is worth", usd, "US dollars"
```


Exercise

Write a program that converts a temperature from Fahrenheit to Celsius.

The formula to convert from Fahrenheit to Celsius is:

Celsius = (5 / 9) * (Fahrenheit - 32)