Python 程式設計

陳建良

What will we cover in this course?

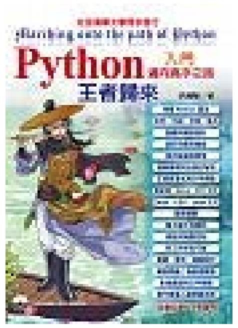
- Input, Processing, and Output
- Decision Structures and Boolean Logic
- Repetition Structures
- Functions
- Files and Exceptions
- Lists and Tuples
- More About Strings
- Dictionaries and Sets
- Classes and Object-Oriented Programming
- Inheritance
- Recursion
- GUI Programming
- Numpy, Pandas, Matplotlib, Seaborn
- Web scraping

真理大學 Aletheia University 資訊工程學系

評分標準

- 期中考筆試與上機考試：30%
- 期末Project：30%
- 課堂作業：20%
- 課後作業：10%
- 出缺席：10%

教材



- Text Book
  - Python入門邁向高手之路王者歸來，洪錦魁，上奇資訊
  - 講義下載處: https://ilms.au.edu.tw
- Reference Book
  - Python 技術者們：實踐！帶你一步一腳印由初學到精通， 施威銘研究室，旗標
  - 用Python學程式設計運算思維，李啟龍，碁峰

WHICH PROGRAMMING LANGUAGE

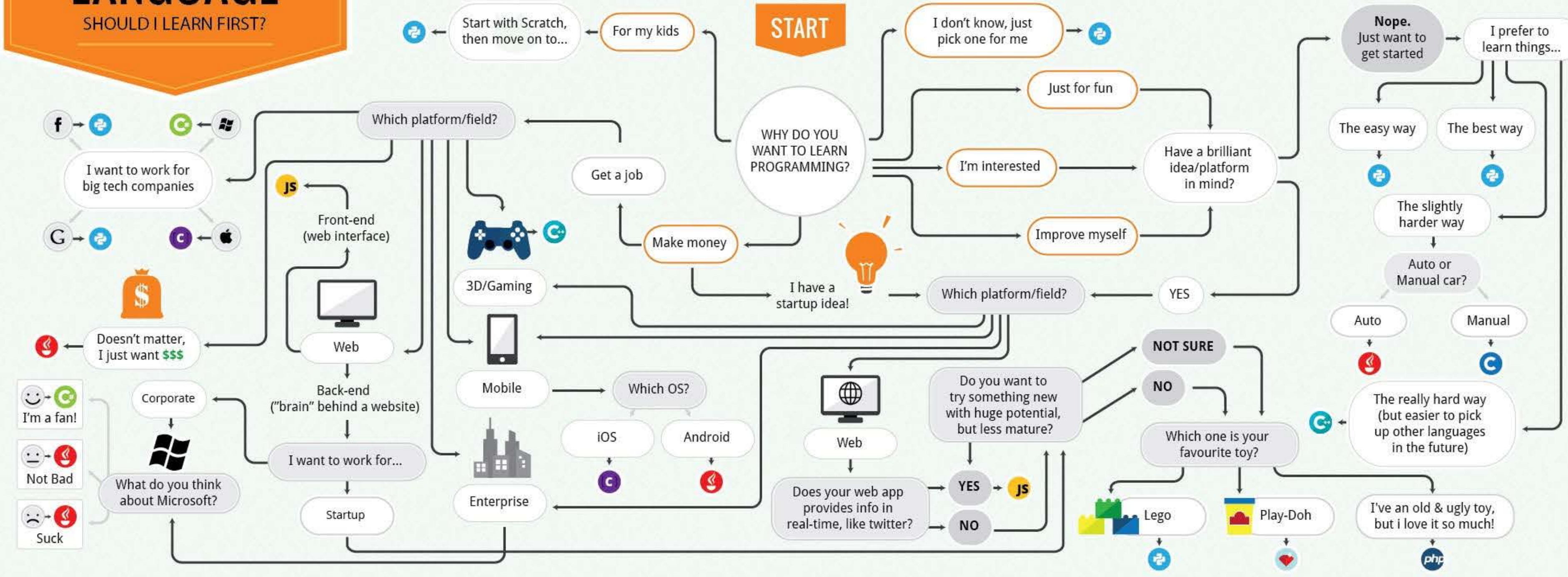
SHOULD I LEARN FIRST?

WHAT IS PROGRAMMING?

Writing very specific instructions to a very dumb, yet obedient machine.

LANGUAGES

- PYTHON
- JAVA
- C
- PHP
- C++
- JAVASCRIPT
- C#
- RUBY
- OBJECTIVE-C



START
WHY DO YOU WANT TO LEARN PROGRAMMING?
For my kids
Start with Scratch, then move on to...
I don't know, just pick one for me
Just for fun
I'm interested
Improve myself
Have a brilliant idea/platform in mind?
Nope. Just want to get started
I prefer to learn things...
The easy way
The best way
The slightly harder way
Auto or Manual car?
Auto
Manual
The really hard way (but easier to pick up other languages in the future)
YES
Which platform/field?
I have a startup idea!
Make money
Get a job
Which platform/field?
I want to work for big tech companies
Doesn't matter, I just want $$$
Front-end (web interface)
Web
Back-end ("brain" behind a website)
I want to work for...
Corporate
Startup
What do you think about Microsoft?
I'm a fan!
Not Bad
Suck
3D/Gaming
Mobile
Which OS?
iOS
Android
Enterprise
Web
Does your web app provides info in real-time, like twitter?
YES
NO
Do you want to try something new with huge potential, but less mature?
NOT SURE
NO
Which one is your favourite toy?
Lego
Play-Doh
I've an old & ugly toy, but i love it so much!

THE LORD OF THE RINGS ANALOGY TO PROGRAMMING LANGUAGES

Python
The Ent

DIFFICULTY ★☆☆☆☆

Help little Hobbits (beginners) to understand programming concepts

Help Wizards (computer scientists) to conduct researches

Widely regarded as the best

Java
Gandalf

DIFFICULTY ★★★☆☆

Wants peace & works with everyone (portable)

Very popular on all platforms, OS, and devices due to its portability

One of the most in demand & highest

C
One Ring

DIFFICULTY ★★★☆☆

The power of C is known to them all

Everyone wants to get its Power

Lingua franca of programming language

C++
Saruman

DIFFICULTY ★★★★☆

Everyone thinks that he is the good guy

But once you get to know him, you will realize he wants the power, not good deeds

Complex version of C with a lot more

JavaScript
Hobbit

DIFFICULTY ★★☆☆☆

Frequently underestimated (powerful)

Well-known for the slow, gentle life of the Shire (web browsers)

"Java and Javascript are similar like Car and Carpet are similar" - Greg

C#
Elf

DIFFICULTY ★★★☆☆

Beautiful creature (language), used to stay in their land, Rivendell (Microsoft Platform), but recently started to open up to their neighbours (open source)

A popular choice for enterprise to create websites and Windows

Ruby
Man (Middle Earth)

DIFFICULTY ★★☆☆☆

Very emotional creature

They (some Ruby developers) feel they are superior & need to rule the Middle Earth

Mostly known for its popular web

PHP
Orc

DIFFICULTY ★★☆☆☆

Ugly guy (language) and doesn't respect the rules (inconsistent and unpredictable)

Big headache to those (developers) to manage them (codes)

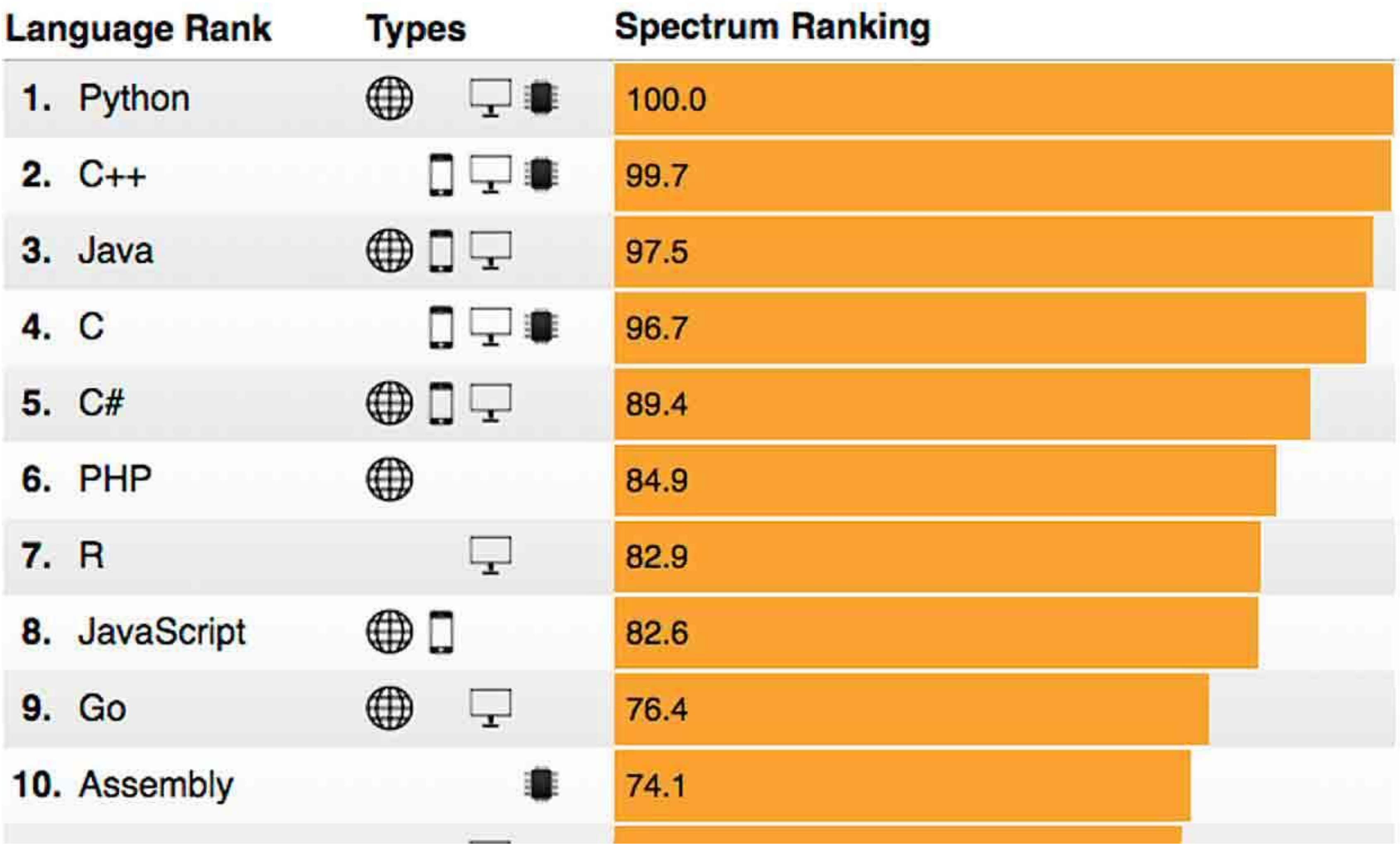
Objective-C
Smaug

DIFFICULTY ★★★☆☆

Lonely and loves gold

Primary language used by Apple for Mac OS X & iOS

Choose this if you want to focus on developing iOS or OS X apps only



Language Rank		Types	Spectrum Ranking
1.	Python	🌐 🖥 ▪	100.0
2.	C++	📱 🖥 ▪	99.7
3.	Java	🌐 📱 🖥	97.5
4.	C	📱 🖥 ▪	96.7
5.	C#	🌐 📱 🖥	89.4
6.	PHP	🌐	84.9
7.	R	🖥	82.9
8.	JavaScript	🌐 📱	82.6
9.	Go	🌐 🖥	76.4
10.	Assembly	▪	74.1

https://hackernoon.com/top-3-programming-language-to-watch-out-in-2019-95995e81ad2b

What is Python?

- Python is an **interpreted, high-level, general-purpose** programming language.
- Python has a design philosophy that emphasizes **code readability**, notably using **significant whitespace**.
- It provides constructs that enable clear programming on both small and large scales.
- Python features a **dynamic type** system and automatic **memory management**.
- It supports multiple programming paradigms, including **object-oriented**, **imperative**, **functional** and **procedural**, and has a large and comprehensive **standard library**.

真理大學 Aletheia University 資訊工程學系

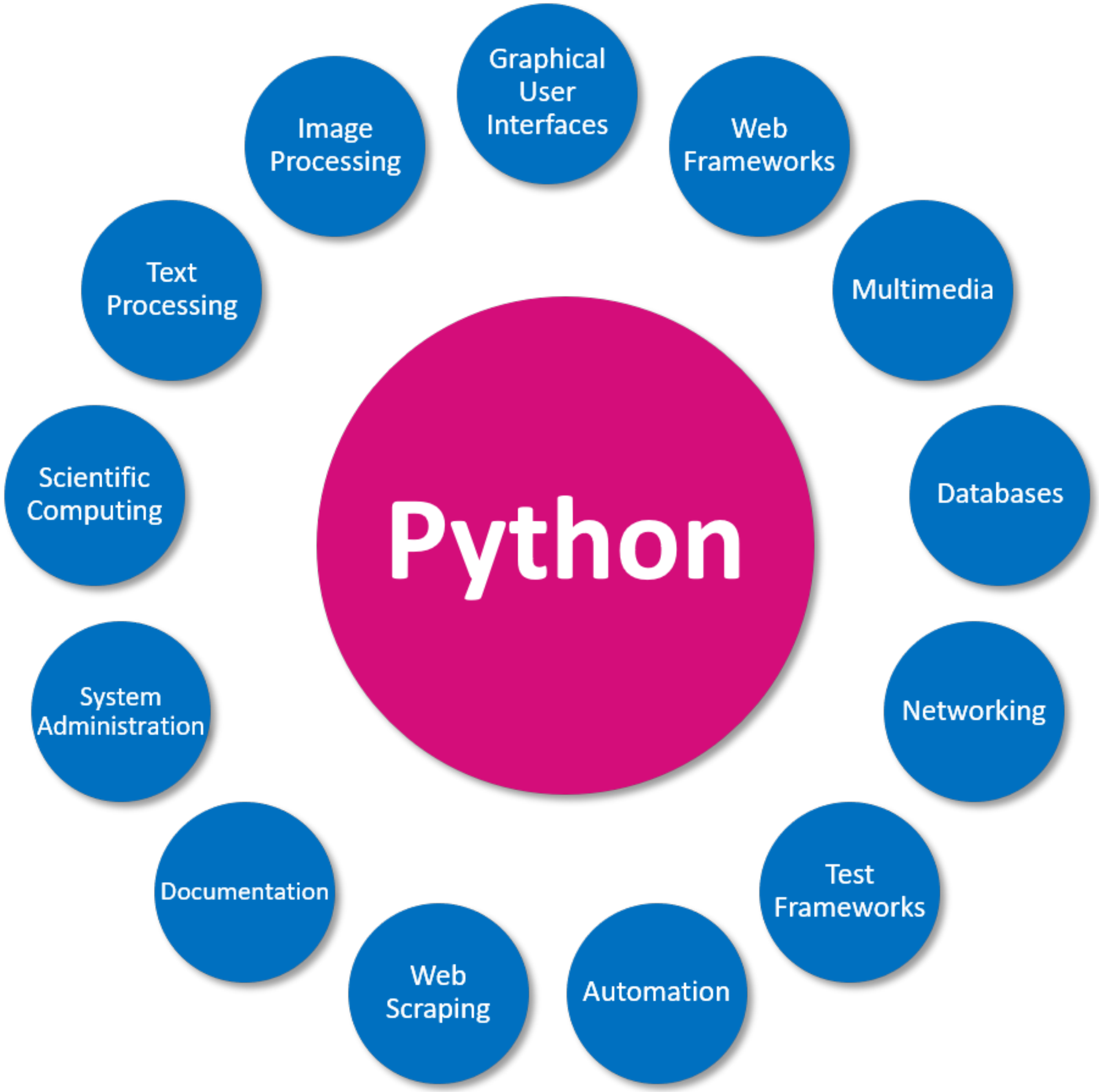
Why learn Python?

- Cross Platform – Windows, MAC, Linux
- Easy Learning
- Clear Syntax and Structure
- Extended and Embedded
- Open Source
- Powerful
- Widely used (Google, NASA, Yahoo, Electronic Arts, some UNIX scripts etc.)

Guido van Rossum

2008	Python 3
2000	Python 2
1994	Python 1

https://medium.com/python4u/hello-python-509eabe5f5b1



Aletheia University 資訊工程學系

Who uses Python?

yelp

Massachusetts Institute of Technology

ebay

Los Angeles Times

cars.com

YouTube

redhat

FreeBSD

NOKIA
Connecting People

The Walt Disney Studios

Quora

NASA

reddit

LexisNexis

Google

Introduction

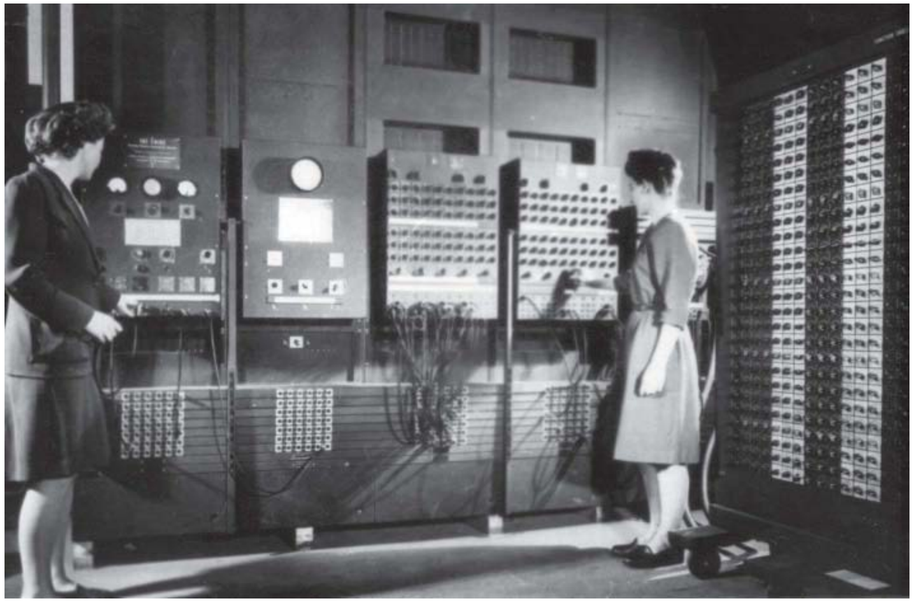
- A *program* is a set of instructions that a computer follows to perform a task.
- Programs are commonly referred to as *software*.
- Software is essential to a computer because it controls everything the computer does.
- All of the software is created by programmers or software developers.
- A *programmer*, or *software developer*, is a person with the training and skills necessary to design, create, and test computer programs.

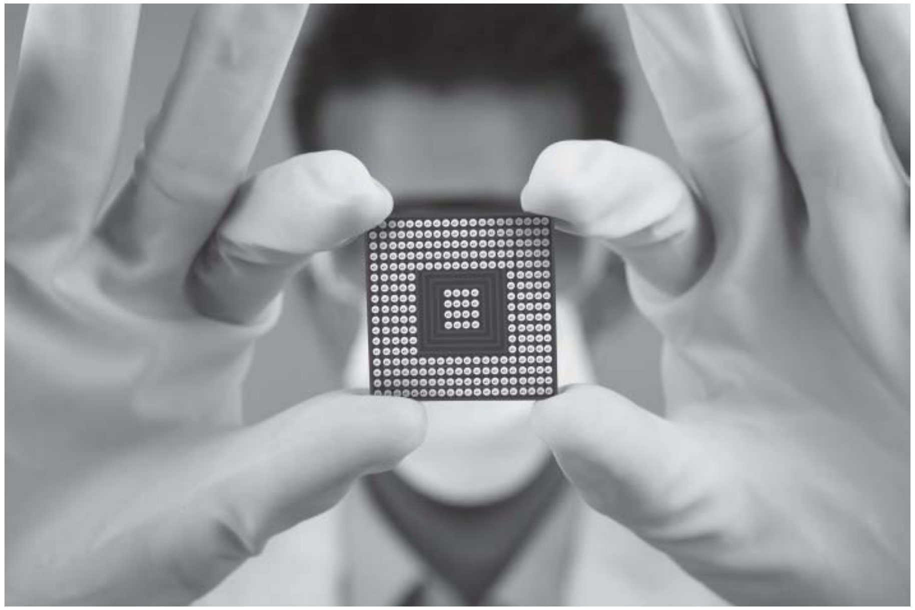
Hardware and Software

- The term ***hardware*** refers to all of the physical devices, or *components*, of which a computer is made.
- A typical computer system consists of the following major components:
  - The central processing unit (CPU)
  - Main memory
  - Secondary storage devices
  - Input devices
  - Output devices

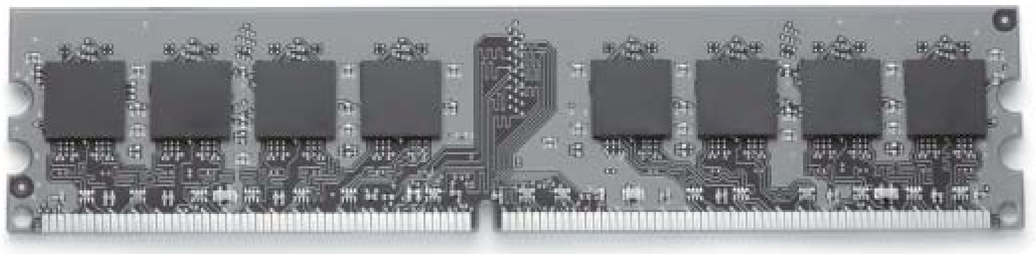
The CPU

- When a computer is performing the tasks that a program tells it to do, we say that the computer is *running* or *executing* the program.
- The *central processing unit*, or *CPU*, is the part of a computer that actually runs programs.





Main Memory



- You can think of ***main memory*** as the computer's work area.
- This is where the computer stores a program while the program is running, as well as the data that the program is working with.
- Main memory is commonly known as ***random-access memory***, or ***RAM***.
- It is called this because the CPU is able to quickly access data stored at any random location in RAM.
- RAM is usually a ***volatile*** type of memory that is used only for temporary storage while a program is running.
- When the computer is turned off, the contents of RAM are erased.

Aletheia University 資訊工程學系

Secondary Storage Devices

- *Secondary storage* is a type of memory that can hold data for long periods of time, even when there is no power to the computer.
- Programs are normally stored in secondary memory and loaded into main memory as needed.
- The most common type of secondary storage device is the *disk drive*.
- *Solid-state drives*, which store data in solid-state memory, are increasingly becoming popular.
- External storage devices, which connect to one of the computer's communication ports, are also available.
- External storage devices can be used to create backup copies of important data or to move data to another computer.
- For example, *USB drives* are small devices that plug into the computer's USB (universal serial bus) port.

Input Devices

- *Input* is any data the computer collects from people and from other devices.
- The component that collects the data and sends it to the computer is called an *input device*.
- Common input devices are the keyboard, mouse, touchscreen, scanner, microphone, and digital camera.

Output Devices

- *Output* is any data the computer produces for people or for other devices.
- The data is sent to an *output device*, which formats and presents it.
- Common output devices are video displays and printers.

真理大學 Aletheia University 資訊工程學系

Software

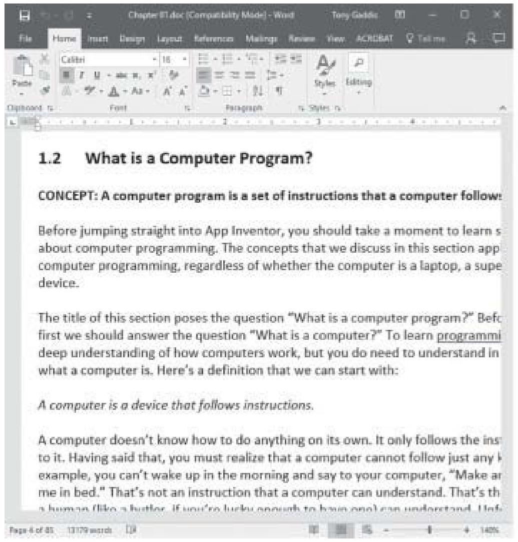
- If a computer is to function, software is not optional.
- Everything computer does, from the time you turn the power switch on until you shut the system down, is under the control of software.
- There are two general categories of software:
  - system software
  - application software.

System Software

- The programs that control and manage the basic operations of a computer are generally referred to as ***system software***.
- System software typically includes the following types of programs:
  - ***Operating Systems***
  - ***Utility Programs***

Application Software

- Programs that make a computer useful for everyday tasks are known as ***application software***.
- These are the programs that people normally spend most of their time running on their computers.



1.2 What is a Computer Program?

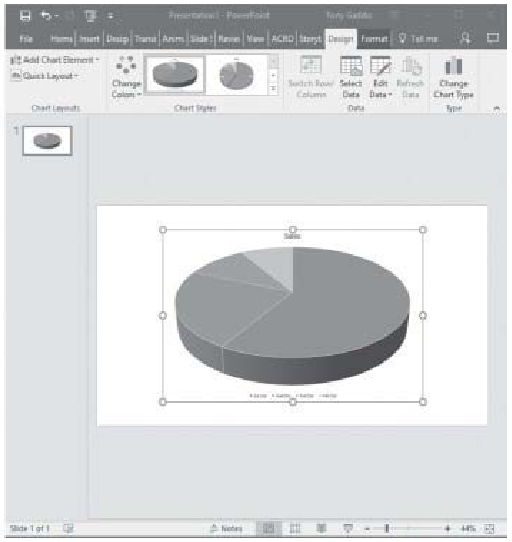
CONCEPT: A computer program is a set of instructions that a computer follow

Before jumping straight into App Inventor, you should take a moment to learn s
about computer programming. The concepts that we discuss in this section app
computer programming, regardless of whether the computer is a laptop, a supe
device.

The title of this section poses the question "What is a computer program?" Befc
first we should answer the question "What is a computer?" To learn programmi
deep understanding of how computers work, but you do need to understand in
what a computer is. Here's a definition that we can start with:

A computer is a device that follows instructions.

A computer doesn't know how to do anything on its own. It only follows the ins
to it. Having said that, you must realize that a computer cannot follow just any k
example, you can't wake up in the morning and say to your computer, "Make ar
me in bed." That's not an instruction that a computer can understand. That's th



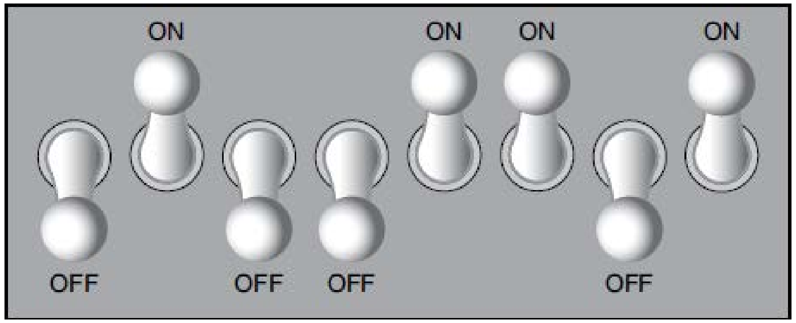
真理大學
Aletheia University
資訊工程學系

How Computers Store Data

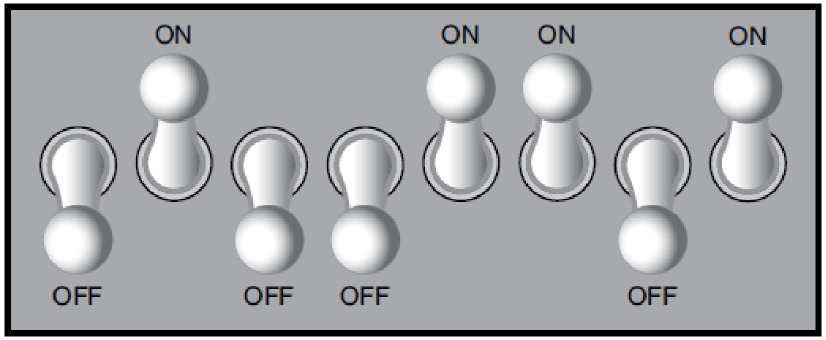
- All data that is stored in a computer is converted to sequences of 0s and 1s.
- A computer's memory is divided into tiny storage locations known as *bytes*.
- One byte is only enough memory to store a letter of the alphabet or a small number.
- In order to do anything meaningful, a computer has to have lots of bytes.
- Most computers today have millions, or even billions, of bytes of memory.
- Each byte is divided into eight smaller storage locations known as bits. The term *bit* stands for *binary digit*.

Aletheia University
資訊工程學系

Think of a byte as eight switches



- When a piece of data is stored in a byte, the computer sets the eight bits to an on/off pattern that represents the data.
- How the number 77 would be stored in a byte



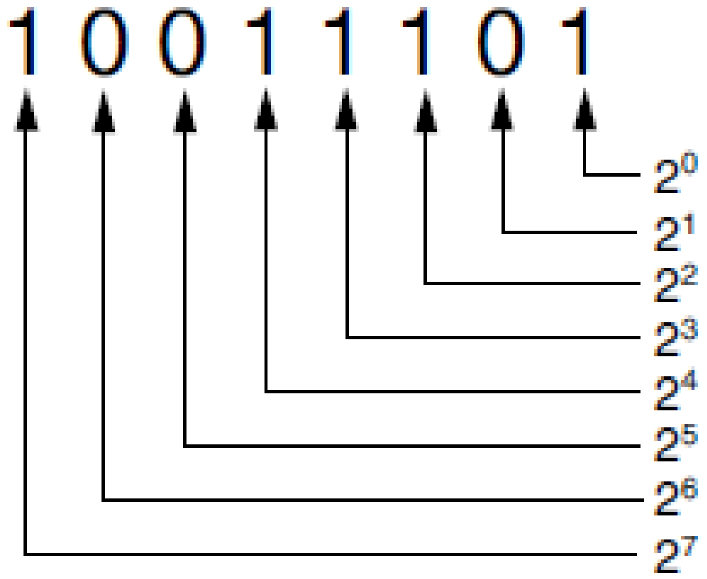
The number 77 stored in a byte.

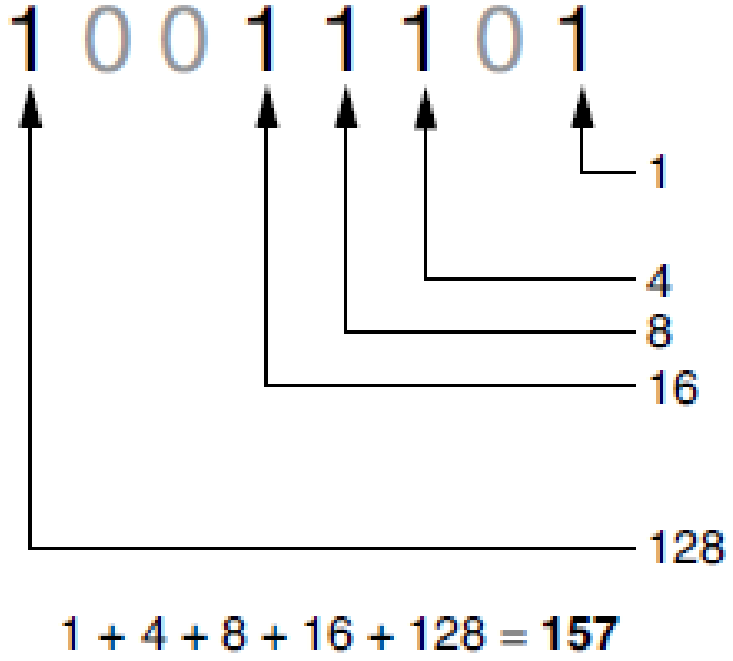
Storing Numbers

- A bit that is **turned off** represents the number **0**, and a bit that is **turned on** represents the number **1**.
- This corresponds perfectly to the ***binary numbering system***.
- In the binary numbering system (or ***binary***), all numeric values are written as sequences of 0s and 1s.
- For example: 10011101

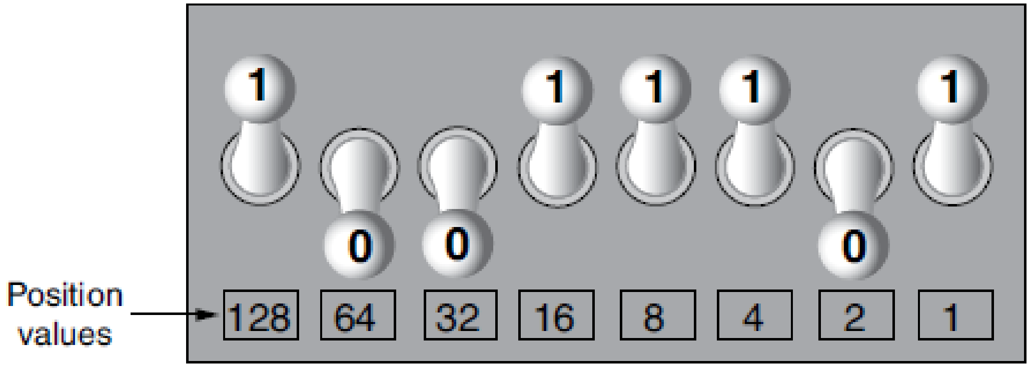
Aletheia University
真理大學
資訊工程學系

The values of binary digits as powers of 2



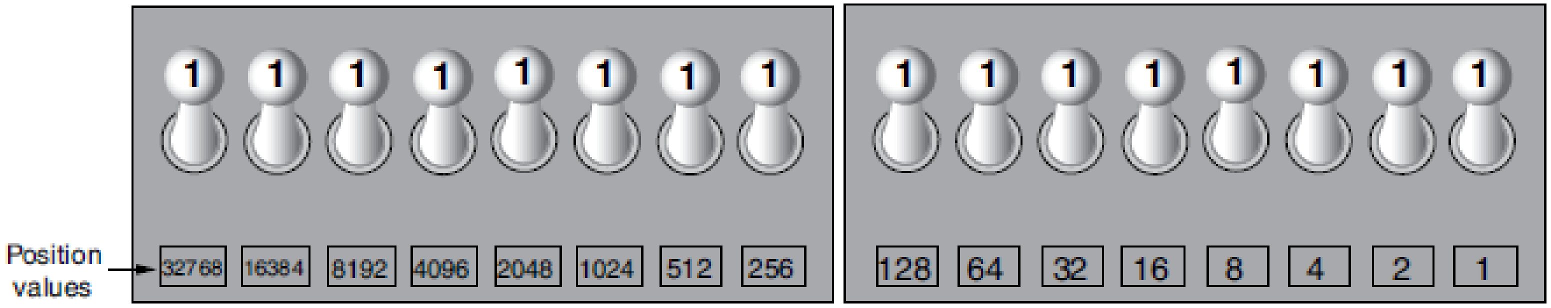


真理大學 Aletheia University 資訊工程學系



128 + 16 + 8 + 4 + 1 = **157**

Two bytes used for a large number:

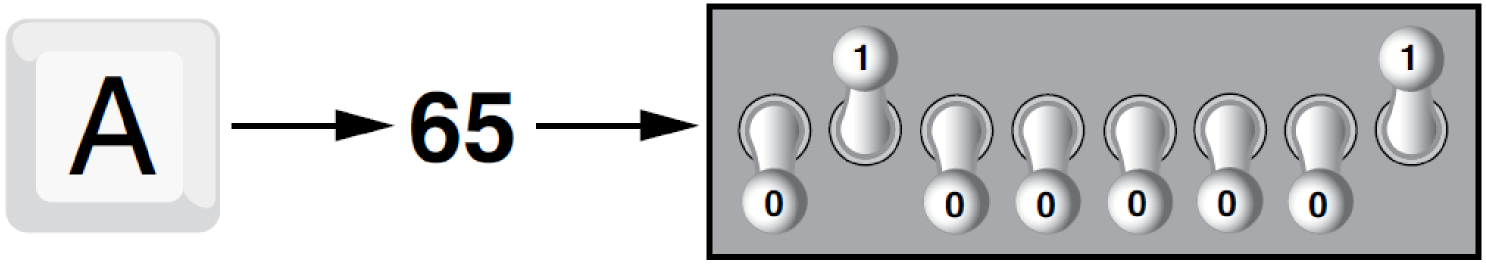


32768 + 16384 + 8192 + 4096 + 2048 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = **65535**

Storing Characters

- When a character is stored in memory, it is first converted to a numeric code.
- The numeric code is then stored in memory as a binary number.
- Different coding schemes have been developed to represent characters in computer memory.
- The most important of these coding schemes is *ASCII*, which stands for the *American Standard Code for Information Interchange*.
- ASCII is a set of 128 numeric codes that represent the English letters, various punctuation marks, and other characters.

The letter A is stored in memory as the number 65



- ASCII is limited, however, because it defines codes for only 128 characters. To remedy this, the Unicode character set was developed in the early 1990s.

Advanced Number Storage

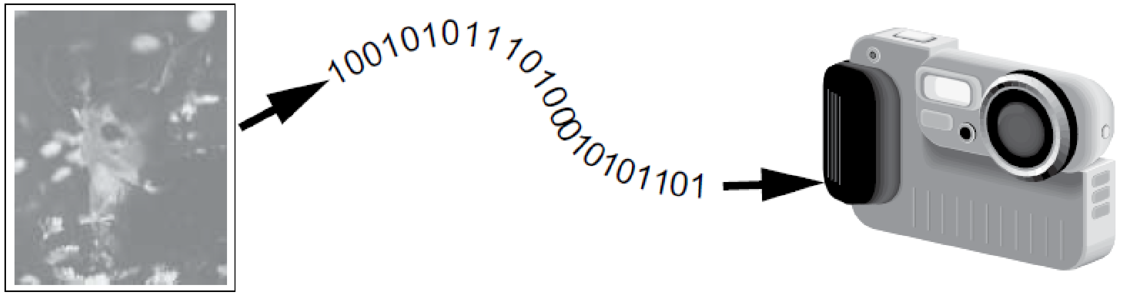
- It occurred to you that the binary numbering system can be used to represent only integer numbers, beginning with 0.
- Negative numbers and real numbers (such as 3.14159) cannot be represented using the simple binary numbering technique.
- But to do so they use encoding schemes along with the binary numbering system.
- Negative numbers are encoded using a technique known as ***two's complement***, and real numbers are encoded in ***floating-point notation***.
- You don't need to know how these encoding schemes work, only that they are used to convert negative numbers and real numbers to binary format.

Other Types of Data

- Computers are often referred to as digital devices. The term *digital* can be used to describe anything that uses binary numbers.
- *Digital data* is data that is stored in binary format, and a *digital device* is any device that works with binary data.
- For example, consider the pictures that you take with your digital camera. These images are composed of tiny dots of color known as *pixels*.

Aletheia University 資訊工程學系

A digital image is stored in binary format

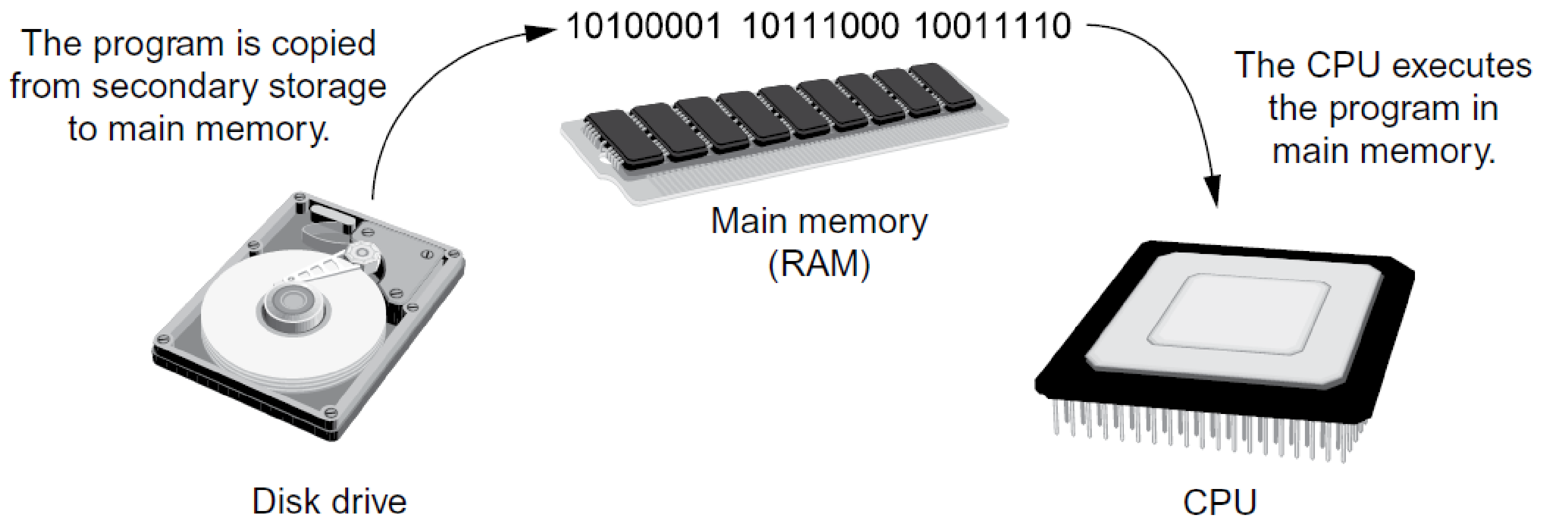


- Each pixel in an image is converted to a numeric code that represents the pixel's color.
- The numeric code is stored in memory as a binary number.

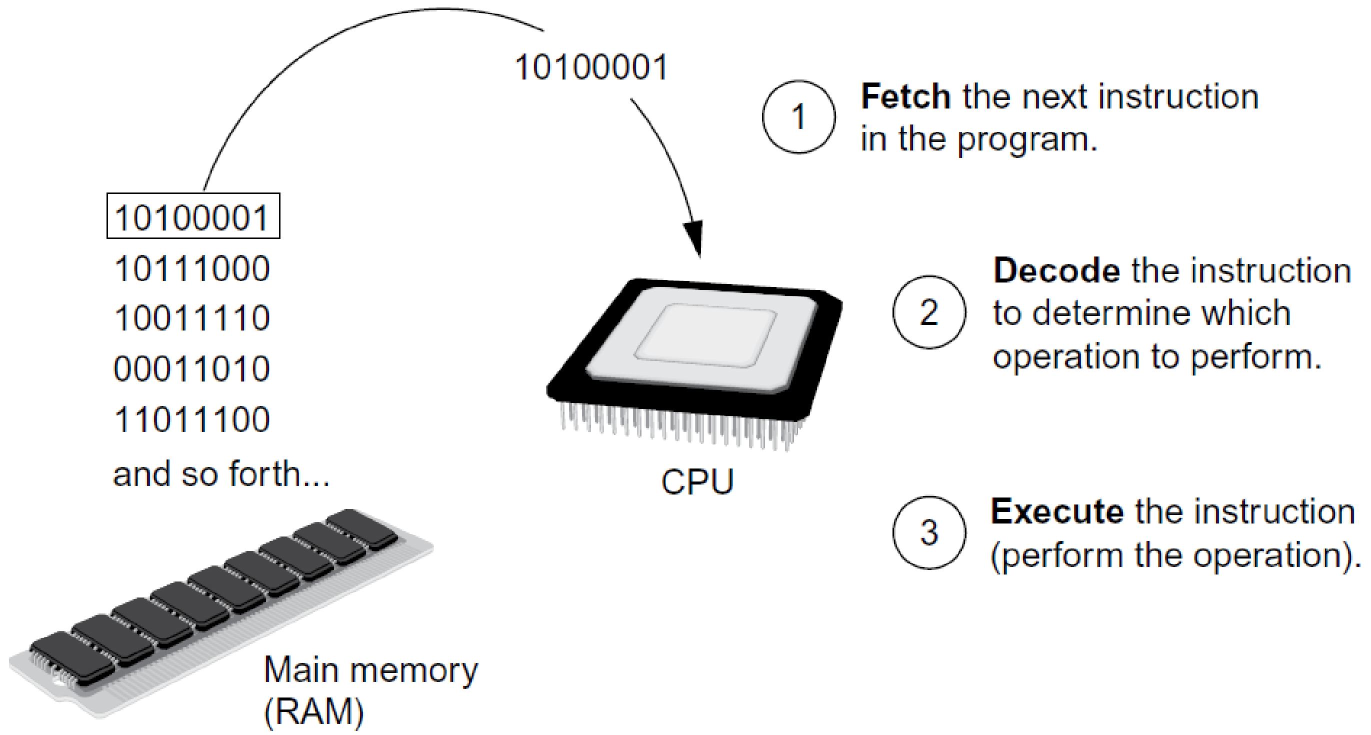
How a Program Works

- A computer's CPU can only understand instructions that are written in machine language.
- Because people find it very difficult to write entire programs in machine language, other programming languages have been invented.
- It is written in 0s and 1s because CPUs only understand instructions that are written in ***machine language***, and machine language instructions always have an underlying binary structure.
- A machine language instruction exists for each operation that a CPU is capable of performing.
- For example, there is an instruction for adding numbers, there is an instruction for subtracting one number from another, and so forth.
- The entire set of instructions that a CPU can execute is known as the **CPU's *instruction set***.

A program is copied into main memory and then executed


The program is copied from secondary storage to main memory.
10100001 10111000 10011110
The CPU executes the program in main memory.
Main memory (RAM)
Disk drive
CPU

- A CPU executes the instructions in a program, it is engaged in a process that is known as the ***fetch-decode-execute cycle***.


10100001
10100001
10111000
10011110
00011010
11011100
and so forth...
Main memory (RAM)
CPU
1 **Fetch** the next instruction in the program.
2 **Decode** the instruction to determine which operation to perform.
3 **Execute** the instruction (perform the operation).

Aletheia University 真理大學 資訊工程學系

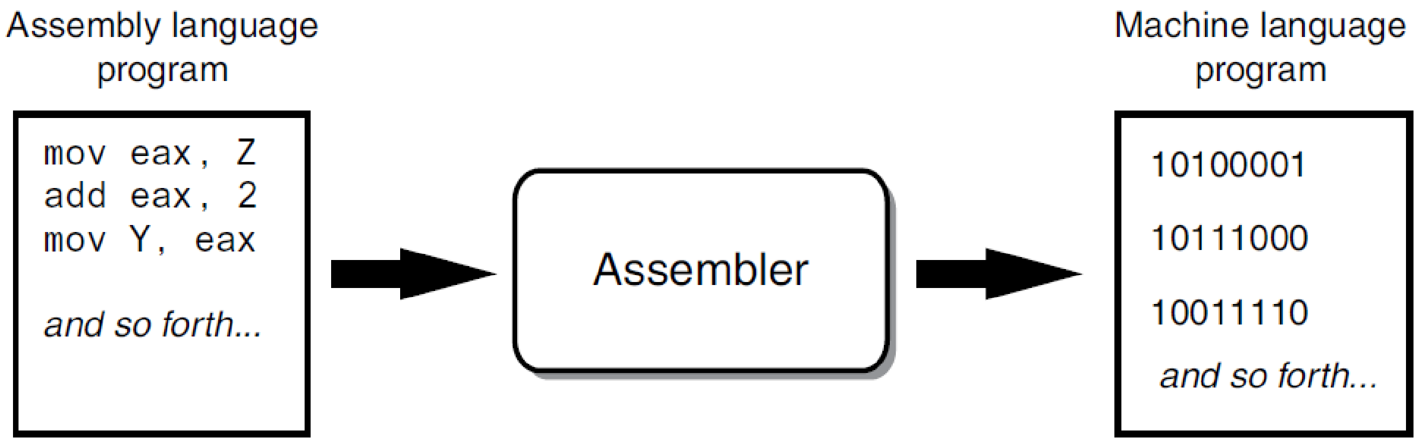
$$\underbrace{10110000}_{\text{ADD}}\ \underbrace{00000001}_{1}\ \underbrace{00000010}_{2}$$

From Machine Language To Assembly Language

- Programming in machine language would also be very difficult, because putting a 0 or a 1 in the wrong place will cause an error.
- Although a computer's CPU only understands machine language, it is impractical for people to write programs in machine language.
- *Assembly language* was created in the early days of computing as an alternative to machine language.
- In assembly language, the mnemonic add typically means to add numbers, mul typically means to multiply numbers, and mov typically means to move a value to a location in memory.
- Assembly language programs cannot be executed by the CPU.

真理大學 Aletheia University 資訊工程學系

- The CPU only understands machine language, so a special program known as an ***assembler*** is used to translate an assembly language program to a machine language program.



```
mov eax, Z
add eax, 2
mov Y, eax
```

```
10100001
10111000
10011110
```

High-Level Languages

- Assembly language is primarily a direct substitute for machine language.
- Like machine language, it requires that you know a lot about the CPU.
- Assembly language also requires that you write a large number of instructions for even the simplest program.
- Because assembly language is so close in nature to machine language, it is referred to as a *low-level language*.
- A high-level language allows you to create powerful and complex programs without knowing how the CPU works and without writing large numbers of low-level instructions.
- In addition, most high-level languages use words that are easy to understand.

Aletheia University 資訊工程學系

- 1950 COBOL，FORTRAN
- 1960 BASIC
- 1970 Ada，Pascal
- 1972 C
- 1983 C++
- 1990 Java，JavaScript，Python，Ruby，Visual Basic

High-Level Languages

COBOL

```
DISPLAY "Hello, World!"
```

Python

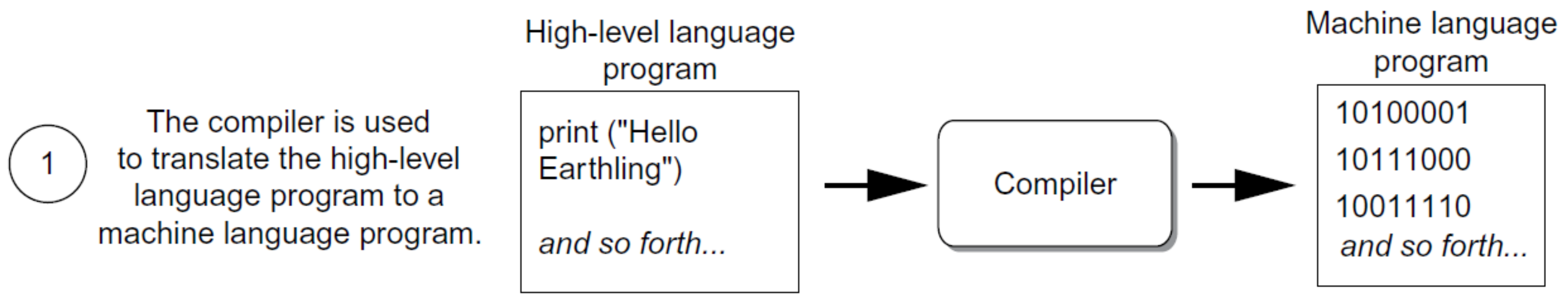
```
print ('Hello, World!')
```

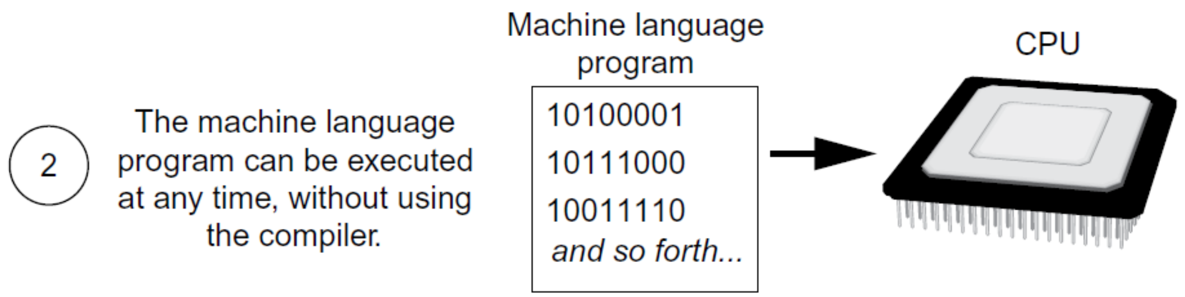
Aletheia University 資訊工程學系

Compilers and Interpreters

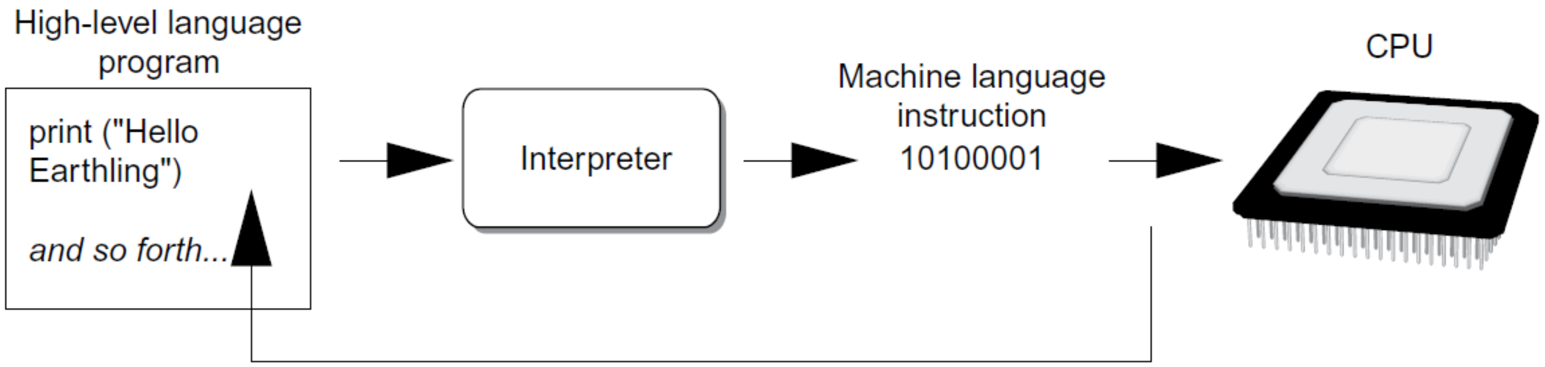
- Because the CPU understands only machine language instructions, programs that are written in a high-level language must be translated into machine language.
- Depending on the language in which a program has been written, the programmer will use either a **compiler** or an **interpreter** to make the translation.
- A *compiler* is a program that translates a high-level language program into a separate machine language program. The machine language program can then be executed any time it is needed.

Compiling A high-level program and executing it


1 The compiler is used to translate the high-level language program to a machine language program.


2 The machine language program can be executed at any time, without using the compiler.

- The Python language uses an *interpreter*, which is a program that both translates and executes the instructions in a high-level language program.


High-level language program
print ("Hello Earthling")
and so forth...
Interpreter
Machine language instruction
10100001
CPU

The interpreter translates each high-level instruction to its equivalent machine language instructions then immediately executes them.

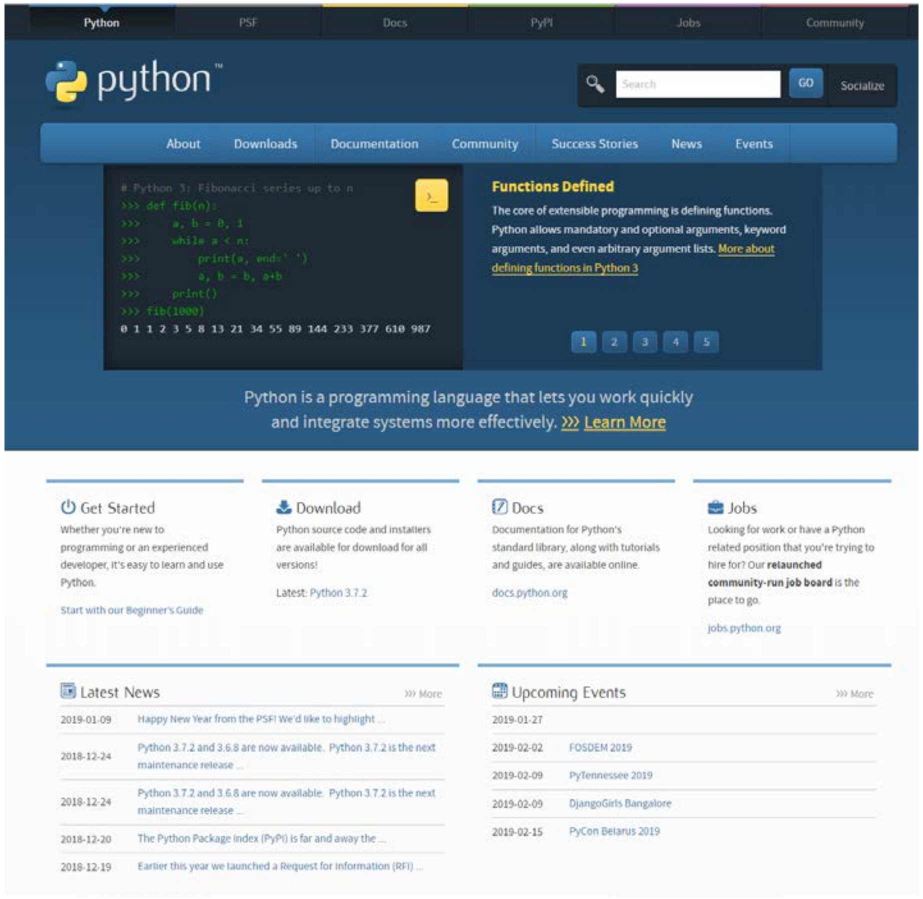
This process is repeated for each high-level instruction.

Aletheia University 資訊工程學系

- The statements that a programmer writes in a high-level language are called *source code*, or simply *code*.
- Typically, the programmer types a program's code into a text editor then saves the code in a file on the computer's disk.
- Next, the programmer uses a compiler to translate the code into a machine language program, or an interpreter to translate and execute the code.
- If the code contains a syntax error, however, it cannot be translated.
- A *syntax error* is a mistake such as a misspelled key word, a missing punctuation character, or the incorrect use of an operator.
- When this happens, the compiler or interpreter displays an error message indicating that the program contains a syntax error.

Installing Python

- Download Python from https://www.python.org



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Start with our Beginner's Guide

Download

Python source code and installers are available for download for all versions!

Latest: Python 3.7.2

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

docs.python.org

Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

jobs.python.org

Latest News >>> More

2019-01-09	Happy New Year from the PSF! We'd like to highlight ...
2018-12-24	Python 3.7.2 and 3.6.8 are now available. Python 3.7.2 is the next maintenance release ...
2018-12-24	Python 3.7.2 and 3.6.8 are now available. Python 3.7.2 is the next maintenance release ...
2018-12-20	The Python Package Index (PyPI) is far and away the ...
2018-12-19	Earlier this year we launched a Request for Information (RFI) ...

Upcoming Events >>> More

2019-01-27	
2019-02-02	FOSDEM 2019
2019-02-09	PyTennessee 2019
2019-02-09	DjangoGirls Bangalore
2019-02-15	PyCon Belarus 2019

真理大學 Aletheia University 資訊工程學系

Using Python

- Python must be installed and configured prior to use
  - One of the items installed is the Python interpreter
- Python interpreter can be used in two modes:
  - Interactive mode: enter statements on keyboard
  - Script mode: save statements in Python script

Interactive Mode

- When you start Python in interactive mode, you will see a prompt
  - Indicates the interpreter is waiting for a Python statement to be typed
  - Prompt reappears after previous statement is executed
  - Error message displayed If you incorrectly type a statement
- Good way to learn new parts of Python

The IDLE Programming Environment

- IDLE (Integrated Development Program): single program that provides tools to write, execute and test a program
  - Automatically installed when Python language is installed
  - Runs in interactive mode
  - Has built-in text editor with features designed to help write Python programs

真理大學 Aletheia University 資訊工程學系

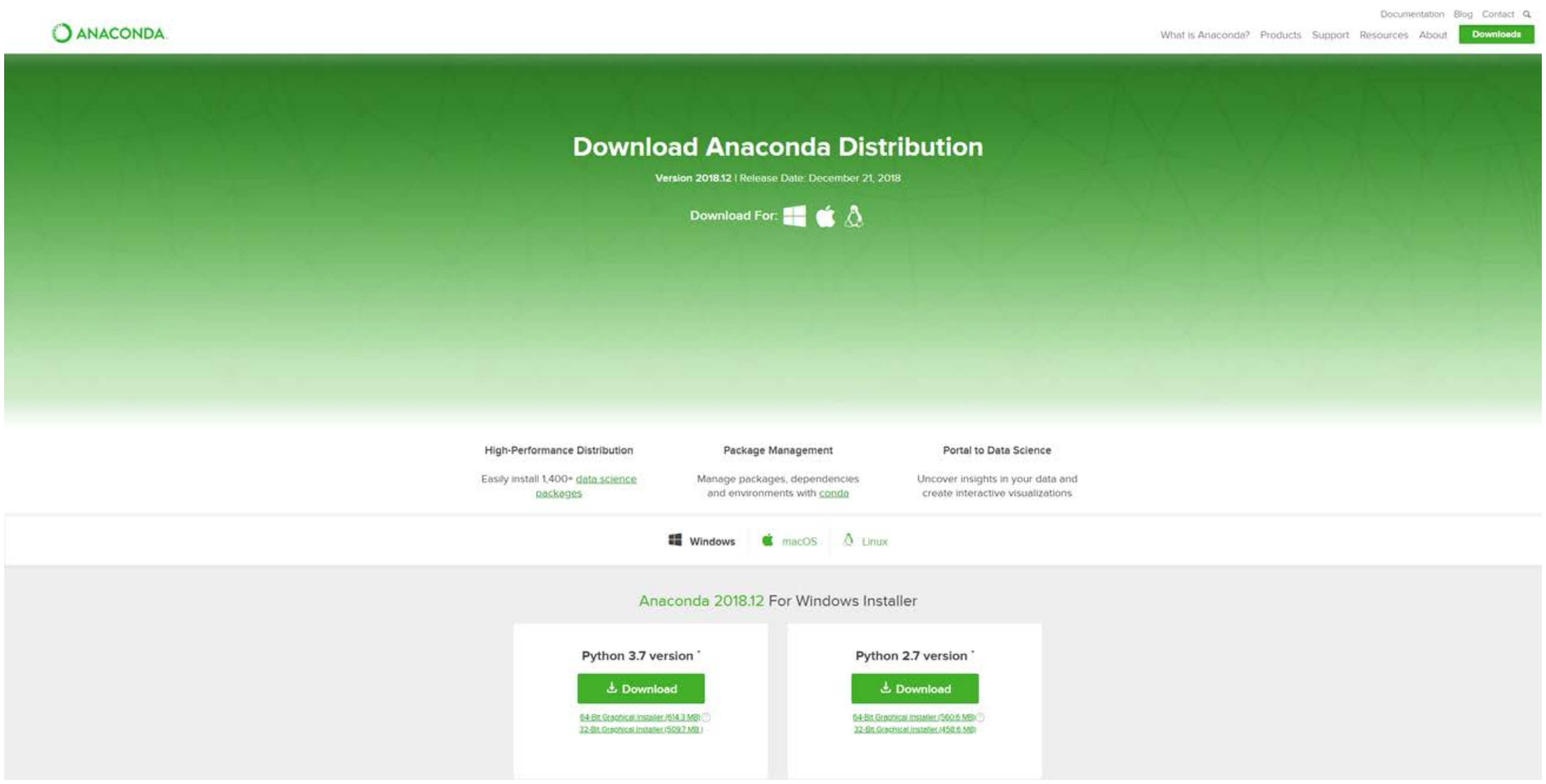
Writing Python Programs and Running Them in Script Mode

- Statements entered in interactive mode are not saved as a program
- To have a program use script mode
  - Save a set of Python statements in a file
  - The filename should have the .py extension
  - To run the file, or script, type

```
python filename
```

at the operating system command line

Recommendation : Anaconda


ANACONDA
Documentation Blog Contact
What is Anaconda? Products Support Resources About Downloads

Download Anaconda Distribution

Version 2018.12 | Release Date: December 21, 2018

Download For:

High-Performance Distribution

Easily install 1,400+ data science packages

Package Management

Manage packages, dependencies and environments with conda

Portal to Data Science

Uncover insights in your data and create interactive visualizations

Anaconda 2018.12 For Windows Installer

Python 3.7 version *

64-Bit Graphical Installer (614.3 MB)

32-Bit Graphical Installer (509.7 MB)

Python 2.7 version *

64-Bit Graphical Installer (560.6 MB)

32-Bit Graphical Installer (458.6 MB)