

Lecture 3

Topics to be covered:

- Operating Systems
- Basic Input and Output

Lecture 3

Let's refresh our memory with Python IDLE:

- 1) What is Python Shell?
- 2) What is Python Editor?
- 3) What command to use to display something on the screen?

Lecture 3

Let's refresh our memory with Python IDLE:

- 1) What is Python Shell?
- 2) What is Python Editor?
- 3) What command to use to display something on the screen?

In **Python shell**:

Write a print statement to display your name!

```
print(
```

Operating Systems: interfacing with hardware

An **application** (or just **app**) is a program that a user runs to carry out a user task like listening to music or surfing the web.

To carry out a task, an application may have to interact with a keyboard, mouse, screen, USB device, WiFi, etc.

The details of such interactions may be complex, involving specific writing/reading sequences of particular memory locations.

Operating Systems: interfacing with hardware

An *operating system* (or *OS*) eases this process by providing a more abstract interface like "Display '6' to the screen" for the apps.

With such abstraction, the same application program can run without modification on different kinds of computers, a feature known as *portability*.

Common operating systems include [Windows](#), [Mac OS X](#), and [Linux](#) for desktop computers, and [iOS](#) and [Android](#) for tablets and smartphones.

Operating Systems: file management

A user usually wishes to maintain *files*.

A *file* is a collection of information stored somewhere on a drive, such as a music file, movie file, word processing file, or an application.

A user may wish to *organize files* using *folders*.

A *folder* is a grouping of files or other folders, that helps a user organize items.

Operating Systems: file management

A **role of an OS** is to manage files and folders, keeping track of such items' locations on a drive, making such items visible to a user, and allowing a user to open files or run apps.

Most operating systems provide some way for a user see what folder/file/app items exist on a drive, as well as to move, delete, or create new items, or to find items whose name or contents match certain words.

Finder (Mac OS)

Windows Explorer (Windows)

Operating Systems: file management

Some common file types and extensions:

File type	Extension	Description
PDF	*.pdf	A PDF file, short for <i>Portable Document Format</i> , describes formatted text/graphics suitable for viewing and printing, independently of apps like a word processor, web browser, etc.
Microsoft Office document	*.docx *.pptx *.xlsx	A Microsoft Office document file is used by the <i>Microsoft Word</i> (.docx), <i>PowerPoint</i> (.pptx), and <i>Excel</i> (.xlsx)
Executable file	*.exe (Windows)	An executable file is an application; executables for Windows typically end with .exe.
Media file	*.jpg *.png *.mp4 *.mov *.mp3 ...	A media file stores a picture (jpg/png), video (mp4, mov), audio (mp3), or other media item.

Operating Systems: multitasking

A *multitasking OS* supports running multiple applications seemingly simultaneously.

To a user, multiple applications like a *music player* and a *web browser* appear to be running simultaneously, but in fact the OS is running a little bit of each application at a time, switching thousands of times per second.

Operating Systems: multitasking

Some computers have *multiple simultaneous users*, such as a server computer.

A *multi-user OS* is a multitasking OS that runs multiple users' applications, keeping each user's applications and files separate, and switching among applications fast enough (typically thousands of times per second) to give the user the illusion of being the only user on the computer.

Operating Systems: summary

Operating system is a special program with various duties, such as

- *interfacing* with hardware (providing a more abstract view to applications),
- *managing* files and folders, and
- running multiple applications seemingly simultaneously (*multitasking*) for one or more users.

Common Operating Systems

Operating System	Description
Unix	<p>Unix is an OS developed in the late 1960's by AT&T Bell Laboratories, focusing on multi-user support, and written in the <i>C high-level language</i> rather than <i>assembly</i> to ease implementations on different computers.</p> <p>Unix is text based, requiring typing to view, copy, move, or delete files or folders, or to run programs.</p>
MS DOS	<p>MS DOS was introduced in 1981 by Microsoft as a relatively simple OS for PCs, which were new at that time. MS DOS was text based.</p>

Common Operating Systems

Operating System	Description
Linux	<p>Linux is a version of Unix developed in the early 1990's by Finnish college student Linus Torvalds, with Linux popular today due in part to being free and open-source, and with people around the world continuing development.</p> <p>Linux was originally text based; popular Linux versions today like Ubuntu have a graphical interface.</p>
Windows	<p>Windows is a graphical OS developed by Microsoft in the mid-to-late 1980s.</p> <p>Windows was originally a simple graphical interface to MS DOS, but later versions were developed new without MS DOS and with powerful graphical features.</p>

Common Operating Systems

Operating System	Description
Mac OS X	Mac OS X is a graphical OS developed by Apple in the early 2000's, based on the Unix OS. Mac OS X was preceded by early Mac OS versions dating back to 1984, which were not Unix-based.
iOS	iOS is a graphical OS developed by Apple in 2007 for Apple smartphones and tablets. A key iOS feature is that most user operations involve touching the screen, using multiple fingers and gestures (like swiping or pinching), rather than typing or using a mouse. iOS has some Unix basis.
Android	Android is a graphical OS released by Google in 2007 for smartphones and tablets, based on the Linux OS. Android also emphasizes interactions via screen touches.

Computational thinking

Mathematical thinking became increasingly important throughout the industrial age, to enable people to successfully live and work.

In the information age, many people believe *computational thinking*, or creating a sequence of instructions to solve a problem, will become increasingly important for work and everyday life.

A sequence of instructions that solves a problem is called an *algorithm*.

Examples of algorithms:

sorting algorithms, searching algorithms, etc.

My third program

Create a new file ([File](#) → [New File](#)) and type in the following:

```
# this is my third program!

name = input("Enter your name, please:")

print("*"*40)
print("Nice to meet you,", name, "!")
print("The weather is wonderful today, isn't it?")
print("*"*40)
```

Save the program ([File](#) → [Save](#)) as `myThirdProgram.py`
Then press [F5](#) or go to [Run](#) → [Run Module](#)
Then check what you see in Python shell...

My third program

Create a new file (**File** → **New File**) and type in the following:

```
# this is my third program  
  
name = input("Enter your name, please:")  
  
print(name)  
print("Nice to meet you, " + name + "!")  
print("Today is a beautiful day, isn't it?")  
print("*" * 40)
```

`input()` function/method will read text entered by the user, and assign the entered text to the `name` variable

variable
(named reference where the information is stored)

me, "I")
ful today,

Save the program (**File** → **Save**) as `myThirdProgram.py`
Then press **F5** or go to **Run** → **Run Module**
Then check what you see in Python shell...

Programs and terminology

A *computer program* mostly consists of a series of commands/instructions, called *statements*.

Each statement usually appears on its own line.

In a program we can see:

- *expressions* (code that return a value when evaluated)
`x * 5`
- *assignment statements* (using the = symbol)
`y = x * 5`
- *print() statements* (displays variables, or expression values, or string literals)
`print("My name is", name)`
- many other things we will learn later

My fourth program

Grab the file `myFourthProgram.py` from our web-site.
Open it in Python Editor.