ITE315 Module 1 Part A - Introduction

Athens State University

Contents

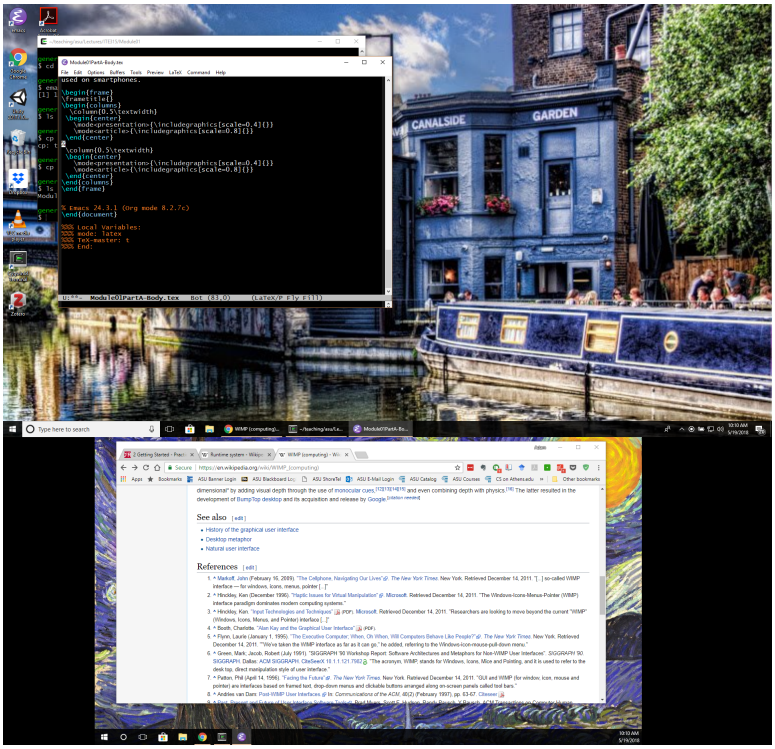
1 Scripting languages and automation

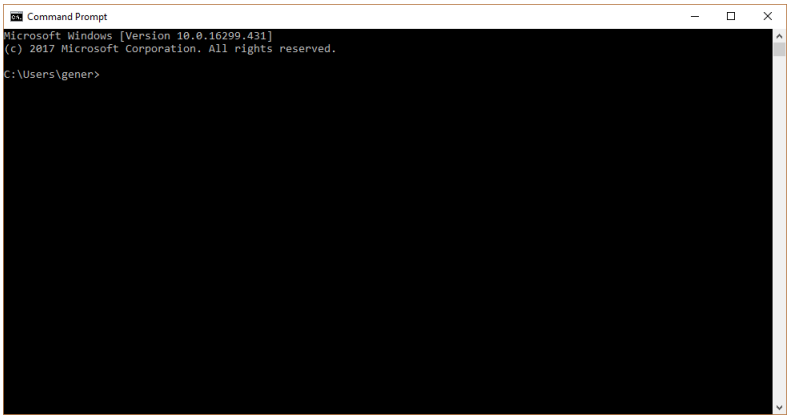
Scripting Language

- **Scripting language**: A programming language optimized for writing programs in a special run-time environment that automate the execution of tasks.
- **Run-time Environment**: The entities and variables available to a program that can be manipulated to complete a task
  - Example: Environment variables provided by the operating system, hardware devices containing data objects

The operating system on a computer provides a set of services that we use to perform useful work. These services are exposed to us via an interface that we call a run-time environment. Classic examples of such environments are the graphical user interfaces provided to us by operating systems like Windows, macOS, Linux, iOS, or Android.

The User Interface





The environment provides a set of tools that we use to complete work. Everyone is familiar with the classic WIMP model: Windows, Icons, Menus, and Pointers. Contrast this with the touchscreen-based models used on smartphones.

There is an older paradigm. It dates back to dark ages of computing when old people like your Glorious Instructor (think what Dr. Evil would have been if he had decided to be an Evil Professor rather than Evil Doctor) were learning how to compute. This model revovled around the entry of commands into devices like Teletypes and Data Entry Terminals.

In modern operating systems, the command-line interface is buried within "Terminal applications" which integrate the command-line environment into the GUI. These applications serve as containers for the classic command-line interpreters.

1.1 A Short Digression About Command-Line Shells

BTW, it's still possible to configure operating systems to use a traditional command-line based user interface. For example, one can configure a Linux system to boot to the command-line rather than the GUI. This is a very common in embedded system environments.

The Command-Line vs. the GUI

- There are a lot of tasks that can be done more efficiently at the command-line than using the GUI
- Task automation: command-line interpreters provide a run-time environment that we can use to "script" repetitive tasks

2 Scripting Languages

Types of Scripting Languages

- **Extension Languages**: a domain-specific language for a particular environment
  - Example: Editor macro-languages such as Emacs Lisp or Office automation languages such as Visual Basic for Applications
  - Example: R, SPSS, or SAS for statistical analysis
  - **Job Control Languages**: a domain-specific language targeted at job control on operating systems
    - Examples: The Bash shell, Windows Powershell, IBM z/OS JCL
  - **Dynamic High-Level Programming Languages**: Portable languages intended for general-purpose use adapted to scripting
    - Examples: Perl, Python, PHP
    - For these languages, "*script*" implies a small (less than 1000 lines of code) program intended for a single and very specific purpose

Characteristics of Scripting Languages

- Designed to be very fast to learn with simple syntax and semantics
- Interpreted rather than compiled
  - **REPL**: Most interpreters implement a *read-evaluate-print loop*
- Extensive use of data abstraction and data hiding
  - Often dynamically typed
  - **Duck typing**: "If walks like duck, quacks like a duck, and smells like a duck, then it's a duck."
  - Implies definiton of variables at point of use

Course Structure