End User Programming

- Customization of application performance by *end users*: regular (typically “non programmer”) users
- Emphasis on application behavior that is conventionally accomplished by programming new or modified “commands”
End User Programming Approaches

- Preferences
  - User chooses among alternative behaviors
  - Choices are limited and predefined
  - Often used for defining layout (e.g., windows in an IDE)

- Scripting languages
  - User writes code in special-purpose language customized for the application (e.g., shell script, elisp)
  - User needs to program

End User Programming Approaches

- Macro recorders
  - User turns on macro recording, interacts with the application, and stops recording
  - System records interactions and makes it possible to reexecute them (e.g., GNU Emacs macros, Microsoft Visual Basic macros)
  - Too literal: Use exact positions, keystrokes
  - Generalizing or fixing macros requires "real" programming

- Programming by demonstration
  - Generalizes macro recorders
  - www.acypher.com/wwid

"Before you begin a VBA project, ensure that you have the time to work with VBA. Programming requires focus and can be unpredictable. Especially as a beginner, never turn to programming unless you have time to work carefully. Trying to write a "quick script" to solve a problem when a deadline looms can result in a very stressful situation. If you are in a rush, you might want to use conventional methods, even if they are monotonous and repetitive."

—Getting Started with VBA in Office 2010