Creating and Modifying Text, part 1

Lecture 4

Strings: Making Human Text in a Computer

- Programming concept: representations of words (*char*acters) in computers
- Character: "H"
  - Unicode: U+0042
  - ASCII: 72
- Literal string: "Hello" …five *mailboxes* with five characters
- Quotes: `""` (double quotes) and `''` (single quotes)

```
"It's okay to do this."
"However, "This is not okay," he said."
"But this is \"another way\" around this problem."
```

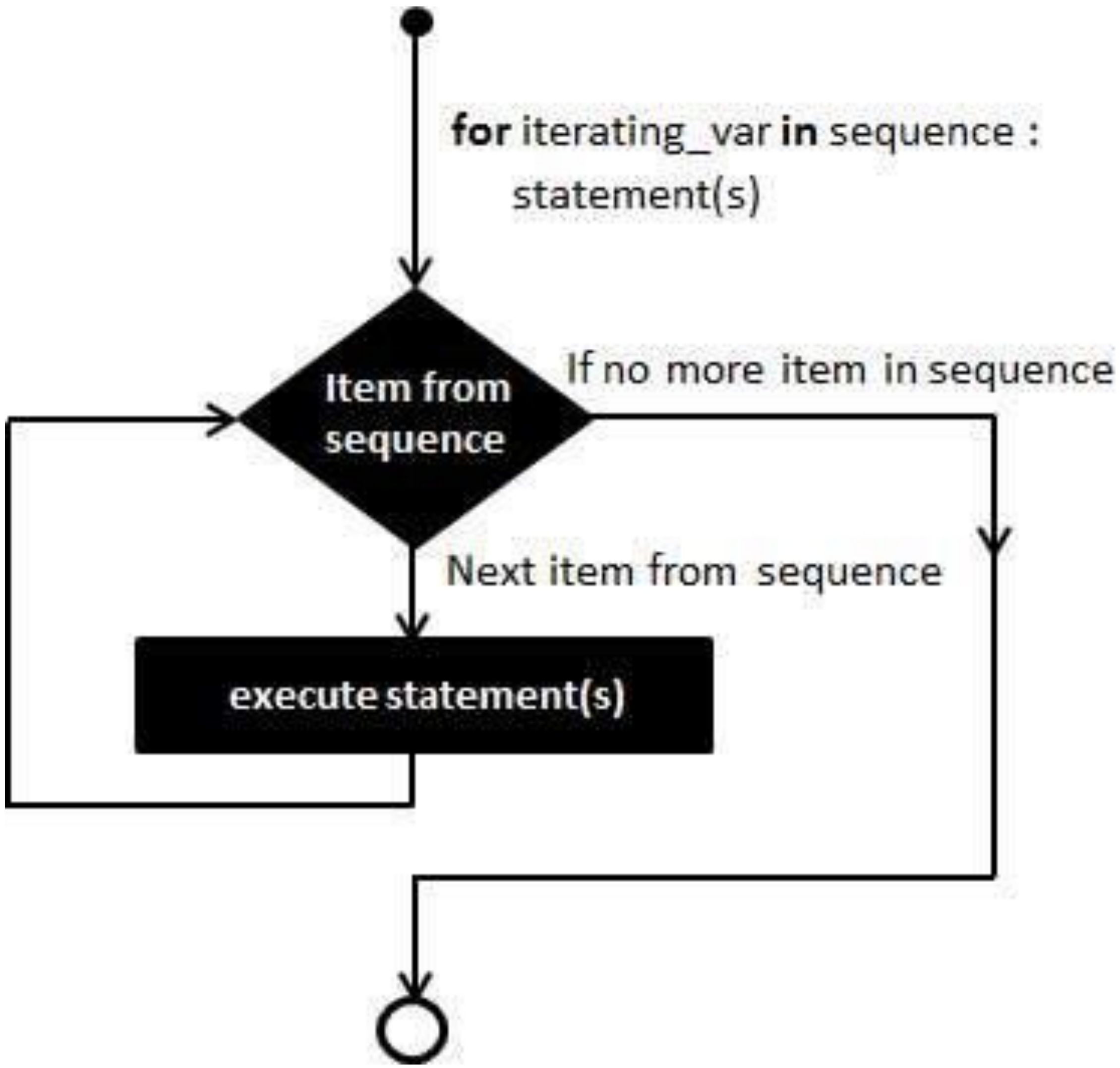
- Type casting
  - Notice the difference: `"4"+"5"` *vs.* `4+5`
  - `str()` and `int()` and `float()`

Making Strings from Strings: Telling Stories

- programming concept: *functions* (reusable blocks of code)
- Concatenation symbol: +
- Functions with parameters
  - Remember: parameters vs. arguments
- Lab assignment 2: Mad Libs

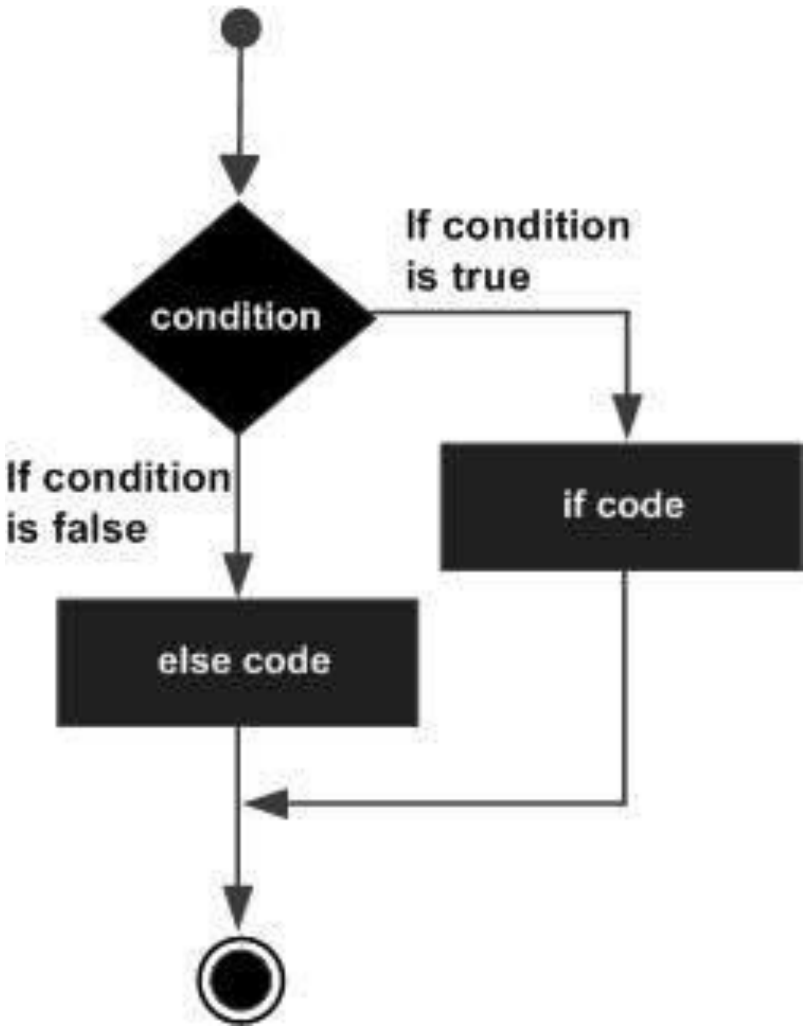
Taking Strings Apart with FOR

- Programming concept: *loop*
- For [each one of these] in [here]
- Don't forget:
  - colon (:)
  - indentation (block)



Testing the Pieces

- Programming concept: *IF* statements (decisions)
- If (some condition)…
- If not (some condition)…
- Don't forget:
  - colon (:)
  - indentation (block)
- Methods
  - Object-oriented programming
  - "dot" notation



Taking Strings Apart, and Putting Strings Together

- Remember: = (assignment operator) does not mean equals
- The "pile" concept
  - Initialize an empty variable
  - Then use the variable in a loop
- Lab assignment 3: Palindrome (homework)