"Twitter based system has managed to detect the earthquake off the Philippines before any other advanced spotting systems being used by Seismologists. US Geological Survey uses the micro-blogging site to quickly gather information about earthquakes around the globe through the use of a system – Twitter Earthquake Detection (Ted) which put behind USGS' own sensors on Friday when it came to detecting 7.6 magnitude earthquake off the Philippine coast. The Ted system gathers earth-quake related messages (Tweets) in real-time from Twitter. The system takes into consideration various parameters like place, time, keywords, photographs of affected places where tremors have been detected. Online information posted by people, Tweets in this case, can be picked up faster by researchers as compared to scientific alerts that may take up to 20 minutes.”

• Talk about lab
  – Environments
  – Passwords
  – Idle3
Exercise 1, p. 23

1. `print("Hello, world!")`
2. `print("Hello", "world!")`
3. `print(3)`
4. `print(3.0)`
5. `print(2+3)`
6. `print(2.0 + 3.0)`
7. `print("2" + "3")`
8. `print("2 + 3 = ", 2+3)`
9. `print(2 * 3)`
10. `print(2 ** 3)`
11. `print(2 / 3)`
Elements of Programs

• Names
  – Given to
    • Variables (celsius, fahrenheit)
    • Modules (scripts: e.g., chaos or main)
  – Names called *identifiers*
  – Identifier must begin with a letter or underscore
    • followed by any sequence of letters, digits, or underscores
  – CaSe sEnSiTiVe
Elements of Programs

• Some identifiers are part of Python
  – *reserved words*
  – and, del, for, is, raise, assert, elif, in, print, etc.
  – See table 2.1
Elements of Programs

• Expressions
  – The fragments of code
  – produce or calculate new data values
  – *Literals* used to represent a specific value, e.g. 3.9
Elements of Programs

>>> x = 5
>>> x
5
>>> print(x)
5
>>> print(spam)

Traceback (most recent call last):
  File "<pyshell#15>", line 1, in toplevel-
    print spam
NameError: name 'spam' is not defined

• NameError
  – try to use a variable without a value assigned to it

Python Programming, 2/e
Elements of Programs

– Simpler expressions can be combined using operators
– +, -, *, /, **
– Spaces are irrelevant within an expression
– Normal mathematical precedence applies
– \((x_1 - x_2) / 2^n + (\text{spam} / k^{**3})\)
Elements of Programs

• Output Statements
  – `print` statement can print any number of expressions

  – Successive print statements will display on separate lines

  – A bare print will print a blank line
Elements of Programs

```python
print(3+4)
print(3, 4, 3+4)
print()
print(3, 4, end=" "),
print(3 + 4)
print("The answer is", 3+4)
```
Elements of Programs

print(3+4) 7
print(3, 4, 3+4) 3 4 7
print() 3 4 7
print(3, 4, end=" "),
print(3 + 4) 3 4 7
print("The answer is", 3+4) The answer is 7
Assignment Statements

• Simple assignment

• \(<variable> = <expr>\)
  variable is an identifier, expr is an expression

• RHS is evaluated
• Produced value then associated LHS variable
Assignment Statements

- \( x = 3.9 \times x \times (1-x) \)
- \( \text{fahrenheit} = \frac{9}{5} \times \text{celsius} + 32 \)
- \( x = 5 \)
Assignment Statements

• Variables can be reassigned

```python
>>> myVar = 0
>>> myVar
0
>>> myVar = 7
>>> myVar
7
>>> myVar = myVar + 1
>>> myVar
8
>>> 
```
Assignment Statements

- Technically, this model of assignment is simplistic for Python
- Python doesn't overwrite these memory locations (boxes)
- More like putting a “sticky note” on a value and saying, “this is x”
Assigning Input

• Input statement gets user info
• Stores into a variable.
• `<variable> = eval(input(<prompt>))`
Assigning Input

• First the prompt is printed
• `input` waits for the user to enter value and press `<enter>`
• Expression is evaluated to turn it
  – from a string of characters
  – into a Python value (a number)
• Value then assigned to the variable
Simultaneous Assignment

- Several values can be calculated at same time
- `<var>, <var>, … = <expr>, <expr>, …`

- Evaluate the expressions on the RHS
- Assign them to the variables on the LHS
Definite Loops

• Executes a definite number of times
  – at the time Python starts the loop it knows exactly how many iterations to do

• for <var> in <sequence>:
  <body>

• beginning & end of body indicated by indentation
Definite Loops

for <var> in <sequence>:
    <body>

• variable after the for is called the loop index
• takes on each successive value in sequence
• Sequences
  – range(10)
    • list(range(10))
    • range(5*2)
  – [2, 3, 5, 7, 11, 13]
  – [“a”, b, 5]
Definite Loops

- **for** loops alter the flow of program execution
- referred to as *control structures*
for i in range(5):
    print (i*i)

for i in range[3, 1, 4, 1, 5]:
    print (d, end=" ")

for i in range(4):
    print ("Hello")

for i in range(5):
    print (i, 2**i)