

CSC108H Lecture 2

Dan Zingaro

September 12, 2012

Python History

- ▶ Late 1970s: programming language called ABC
 - ▶ High-level, intended for teaching
 - ▶ Only five data types
 - ▶ Programs are supposedly one-quarter the size of the equivalent BASIC or Pascal program
 - ▶ Not a successful project
 - ▶ More ABC information:
<http://homepages.cwi.nl/~steven/abc/>

Python History...

- ▶ 1983: Guido van Rossum joined the ABC team
- ▶ Late 1980s: Guido needed a language for a different project; based it on ABC, removed warts (e.g. ABC wasn't extensible)
- ▶ Python, after Monty Python
- ▶ Guido: Benevolent Dictator for Life (BDFL)... but he's retiring!
- ▶ <http://www.artima.com/intv/> (search for Guido)

What is an Algorithm?

Four features of an algorithm:

- ▶ Finite number of instructions or steps
- ▶ Each instruction is well-defined (not ambiguous)
- ▶ Eventually halts
- ▶ Solves a general class of problems

Programming is how we tell the computer what algorithm it should carry out.

ConceptTest

- ▶ Step 1: write down the number 0
- ▶ Step 2: add 3
- ▶ Step 3: return to step 1

Is this an algorithm?

- ▶ A. Yes
- ▶ B. No, because it contains an infinite number of steps
- ▶ C. No, because it never halts
- ▶ D. No, because step 3 is not well-defined

Python as a Calculator

- ▶ The easiest way to experiment with Python is to enter statements into the Python shell
- ▶ Python supports many common mathematical operators
- ▶ Operators include + (addition), - (subtraction), * (multiplication), ** (exponentiation), / (division), % (remainder)
- ▶ e.g. `11+56` at the shell gives `67`

Python as a Calculator...

- ▶ Use parentheses to change precedence (e.g. $4 + 5 * 3$ compared to $(4 + 5) * 3$)
- ▶ There are two division operators
 - ▶ `/`: exact division
 - ▶ `//`: integer division
 - ▶ e.g. compare $5 / 2$ and $5 // 2$

ConceptTest

What is the difference between the literals 1 and 1.0?

- ▶ A. Nothing. They are both 1
- ▶ B. Nothing. They might be of different types, but types don't matter in programming
- ▶ C. 1 is an integer and 1.0 is a string, because of the . character
- ▶ D. 1 is an integer and 1.0 is a floating-point number
- ▶ E. 1.0 is an integer and 1 is a floating-point number

Checking and Changing Types

Python has a useful function called `type` that tells you the type of the argument.

```
type (1)
type (1.0)
type ("1")
type ("1.0")
```

Sometimes, you'll have a value of one type and want to convert it to another type:

- ▶ `int (value)`: convert to int
- ▶ `float (value)`: convert to float

Variables

- ▶ A variable is a name that refers to a value
- ▶ Variables let us store (“remember”) values so we can use them in several places
- ▶ In some ways, variables in programming are like math variables
 - ▶ e.g. in math, you could say “let $x = 5$ ”
 - ▶ Then, what is the value of $x * 3$?
- ▶ But, they are also different
 - ▶ e.g. in math, variables’ values cannot change once they are given a value
 - ▶ In programming, we can change a variable as often as we like

Assignment Statement

- ▶ The assignment statement lets us give a value to a variable
- ▶ Form: `variable = expression`
- ▶ Two steps:
 1. Evaluate the expression on the right-hand side to get a result
 2. Make the variable on the left-hand side refer to that result

ConcepTest

What is the value of y after the execution of this code?

```
x = 37
```

```
y = x + 2
```

```
x = 20
```

- ▶ A. 39
- ▶ B. 22
- ▶ C. 35
- ▶ D. 20
- ▶ E. 18

ConcepTest

What is the value of y after the execution of this code?

```
x = 37
```

```
y = x - 2
```

```
x = 20
```

- ▶ A. 39
- ▶ B. 22
- ▶ C. 35
- ▶ D. 20
- ▶ E. 18