

Computer Science 111

Fundamentals of Programming I

Overview of Programming

High-Level Programming Languages

- Code a program in a *high level language* that's close to English
- Run another program to translate it to instructions that the computer hardware can understand
- Such translator programs are called *compilers*

Why Python?

Real-world advantages

- Systems programming
- Graphical user interfaces
- Internet scripting
- Component integration
- Database programming
- Rapid prototyping

Pedagogical advantages

- Very simple syntax
- Highly interactive
- Easy to learn and use
- Extensive libraries of high-level resources
- Completely portable, free, and open source
- Widespread user (programmer) community

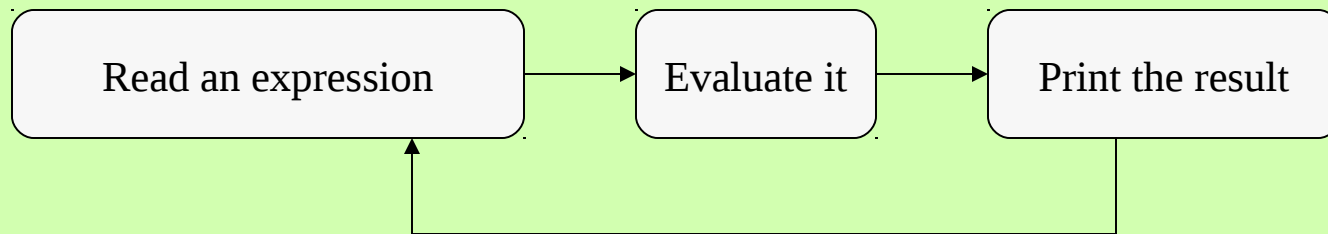
Obtaining Python

- Python was invented by Guido van Rossum in 1992
- Python comes with most Unix-based computer systems
- Python for Windows or any other operating system can be downloaded from <http://www.python.org/>

Developing Python Programs

- Can experiment with small pieces interactively in *shell mode*
- Can compose longer segments with an editor and run them in *script mode*

Evaluating Python Expressions in Shell Mode



Basic Elements: Data

- Numbers
 - Integers: 3, 77
 - Floats: 3.14, .22
- Strings: 'Hi there!', "Hi there!", '\n'
- Truth values (Booleans): True, False

Basic Operations: Arithmetic

Symbol	Meaning	Example
+	Addition or concatenation	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/ or //	Division	x / y or $x // y$
%	Remainder	$x \% y$
**	Exponentiation	$x ** y$

Built-In Functions

- A *function* is an operation that expects zero or more data values (arguments) from its user and computes and returns a single data value

- Examples:

`abs(-5)`

`max(33, 66)`

Library Functions

- A *library* is a collection of resources, including functions, that can be *imported* for use in a program
- Example:

```
import math
```

```
math.sqrt(2)
```

Variables and Assignment

- Variables are used to name data and other resources
- Instead of typing 3.14 for π , define a variable named **pi** to mean 3.14
- Assignment (=) is used to set (or reset) a variable to a value

```
pi = 3.14
```

```
34 * 34 * pi
```

- Variables make programs more readable and maintainable

Library Variables

- Typically define standard constants, such as **pi** and **e**
- Example:

```
import math
```

```
3 * math.pi
```

- Or:

```
from math import pi
```

```
3 * pi
```

Script Mode

- Longer programs can be edited and saved to a file and then run as scripts
- *IDLE* is a script-mode development environment that can be used on any computer system

For Tuesday

Lab meets in Parmly 405