

# CIS 231

Python – Ch. 1

The Way of the program

# Python Programming Language

- First implemented in 1991 by Guido van Rossum
- An example of a *high-level language*
  - Portable between different OS
- An interpreted language
  - Differences between interpreted and compiled
  - Interactive mode and script mode
- We're using Python 3

# 1.1 What is a program?

- Common facets of a programming language:
  - Input
  - Output
  - Math
  - Conditional execution
  - Repetition
  - Repetition

# 1.2 Running Python

- Different environments to run Python
  - Web-based option mentioned by author
  - We won't be doing that
- Using the Python interpreter
  - Interactive mode – ad hoc code
  - Script mode – prepared scripts that we execute

# 1.3 The first program

- “Hello, world”, introduced by Kernighan and Ritchie in their book about the C language
- Produces user-specified output
- Displays text in the console
- Unchanging (isn’t variable)

# 1.4 Arithmetic Operators

- Operators denote a specific arithmetic operation -> +, -, \*, /, \*\*
- Operands are the values the operation is applied to – can be *binary* or *unary*
- The operand types determine the type of the result
  - Exception: In P3, dividing ints gives a float
    - To get an int (floor division), use //

# 1.5 Values and types

- Values are pieces of data that a program works with
- Values have a particular type:
  - Integer – whole numbers
  - Floating-point – fractional numbers
  - String – collections of characters
    - Can use single- or double-quotes

# 1.6 Formal and natural languages

- Programming languages have a set of rules, or syntax (grammar) that dictates how the program must be structured
- The interpreter converts, or parses, that code into a machine-readable form



# 1.7 Debugging

- The process of finding and correcting errors in your programs
- Error types:
  - *Syntax errors* – code could not be *parsed* because it didn't conform to the language's *syntax*
  - *Runtime errors* (aka *exceptions*) – syntax was ok but an unusual occurrence – e.g. divide by 0
  - *Semantic (logic) errors* – syntactically correct, but mistakes in specifics and/or order

# Next Up

- Ch. 2- Variables, expressions and statements
  - 2.1 – Assignment statements
  - 2.2 – Variable names
  - 2.3 – Expressions and statements
  - 2.4 – Script mode
  - 2.5 – Order of operations
  - 2.6 – String Operations
  - 2.7 – Comments
  - 2.8 - Debugging