Computers and programming
in Python

Computer & Information Sciences

W. H. Bell

Objectives

- Introduce programming languages.
- Introduce variables and functions.
- Discuss evaluation order.
- Summary

How a computer works

- A computer has a processor.
- Accesses memory and storage.
- Adds and subtracts numbers.
- Performs low-level operations.
- The processor is controlled with instructions.
- Many instructions are needed to achieve basic tasks.
- Time-consuming and difficult for humans to understand when creating large programs.

Computer and programmers

Implement a programming language to control a computer.
Low-level and high-level programming languages exist.
Many programming languages, as needed to solve problems.

Programming languages

- Mostly written as text files.
- Text files need to be turned into instructions.
- Can compile them into instructions. - C or C++
- Can interpret them and perform instructions. - Python
- Can partially compile them and then interpret the results. - Java

Variables

- Associated with a space in the computer's memory.
- The size of the memory space depends on the type of the variable.
- The order of the storage depends on the type of the variable.
- Variables are used to store values.
- Pass variables around a program.
- Operate on variables with functions.

Functions

- Perform actions.
- Can have zero or more input arguments.
- Can have zero or more return values.
- Can create variables within "scope" of function.
- Can pass in variables that are outside "scope" of function.

Python

- It is an interpreted programming language.
- The text file (.py) is interpreted by the Python program.
- Interpreter also allows user to type commands – interactive.
- Provides high-level features to help programmer.
- Manages memory for programmer.

Creating a variable

- Created by first assignment.
- Given a type by first assignment.
- Given a memory space, following type of variable.

```
x = 10
s = "This is a text string."
f = 3.1415
```

Integer or whole number (int)

Text string (str)

Floating point number (float)

Using variables

- Can only use a variable when it has been defined.
- Can only use memory within a data structure if allocated.
- Data structures many contain zero or more elements.
- Cannot access an element unless it exists first.

Variable names

- Can choose any name.
- Choose meaningful variable names.
- Avoid names that are very long.
- Cause lines of programs to be very long.
- Normally follow convention, camel case or underscores.

Function names

- Beginning with single or double "_" has special meaning.
- Can choose any name.
- Choose a name that expresses functionality of function.
- Normally follow convention, camel case or underscores.
- Avoid names that are very long.
- Cause lines of programs to be very long.

Evaluation order

- Expressions are evaluated sequentially.

```
x = x + 10
```

1. Add 10 to the value that is stored in `x`.
2. Assign the resulting value to the variable `x`.

Evaluation order

```
s = "Python is great"
s = s + "!"
s = s.replace("Python","Programming")
print(s)
```

1. Create a variable called `s` as a string (str).
2. Assign `"Python is great"` to `s`.
3. Append `"!"` to the string that is stored in s and assign the result to s.
4. Call the replace function, passing it `"Python"` and `"Programming"`.
5. Assign the result of replace to `s`.
6. Print the contents of `s` on the screen.

Evaluation order

```
for i in range(10):
    print(i)
```

1. Create a range from zero to 10, in steps of one.
2. For each number that is generated by the range loop.
   a. Create a variable called `i` as an integer (int).
   b. Is there another value in the range? (Yes – continue, No – exit the loop)
   c. Put the new value into `i`.
   d. Print the value of `i` on the screen.

Loop

Summary

- Introduced programming.
- Introduced variables and functions.
- Discussed evaluation order.