Introduction to Python

Object Oriented System Design

Otago Polytechnic
Dunedin, New Zealand

Overview

Introduction

Variables, types, and values

Control flow

Functions and classes

Python

- Python is a widely used, object-oriented high level language.
- It is an expressive language that facilitates rapid development.
- The language syntax supports the Python community's stated goal of producing readable code.
- Python is supported on many operating systems.
- Python is used at top-tier technology organisations like Google, NASA, Red Hat, and Rackspace.

Python style

- The Python language has a set of universal language-wide style guidelines.
- The style guidelines are published in the *Python Enhancement Proposal* (PEP) 8 at `https://www.python.org/dev/peps/pep-0008/`.
- In this class we will follow the PEP 8 style guidelines.

THE PYTHON INTERPRETER

- Python is an interpreted language.
- You can intereact with the interpreter directly.
- This can be really useful when you are developing because you can test little segments of code directly.

Overview

Introduction

Variables, types, and values

Control flow

Functions and classes

Variables

- Python variable do not have types associated with them.
- They are not declared before use.
- To create a variable, just assign it a value.
- Python variable are references to objects.

Types

- Python values are strongly types and they are all objects.
- Numeric, string, and boolean types behave in the ways you would generally expect.
- There is no character type. A character is just a short string.
- Python's collection types are powerful and flexible. Many problems that might require a class in other languages can be solved with a basic collection in Python.
    - List
    - Tuple
    - Dictionary
    - Set, Frozenset

Overview

Introduction

Variables, types, and values

Control flow

Functions and classes

White space

- People who are new to Python are surprised by the significant white space.
- The surprise wears off quickly.
- Statements are generally terminated by a line ending.
- Code blocks are denoted by indentation.

Conditionals

```
if(a > 0):
    print("positive")
elif(a < 0):
    print("negative")
else:
    print("zero")
```

For loops

`for` loops over anything that implements a sequence.

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

for char in "spam":
    print(char)
```

While loops

```
while(stop != "y"):
    print("One more time.")
    stop = input("Go again? ")
```

Overview

Introduction

Variables, types, and values

Control flow

Functions and classes

Functions

- Define functions with the `def` keyword.
- Functions can return one or more values.
- Functions lacking an explicit return return the special value `None`.
- Functions in Python are first-class.

Functions

```
def times_two(x):
    return 2 * x
```

Classes

- Define classes with the `class` keyword.
- Classes support inheritance, including multiple inheritance.
- Classes include class variables, instance variables, and methods.