The Beauty and Joy of Computing

Beyond Blocks Python

Session 2: Data Structures

UC Berkeley EECS
Head TA
Michael Ball

UC Berkeley EECS
TA
Peter Sujan

bjc

Data Structures (Overview)

- **Review (and some new introductions)**
- **Sequences**
  - Operators
- **Sets**
  - Operators
- **Dictionaries**
- **Higher-Order Functions**
- **Let's Re-visit the midterm Exam!**

Review

- **Typing, Build-In Types**
  - Int, function, string, list, etc
- **Variables**
- **Looping and Conditionals**
  - for loops,
  - While loops
- **Functions**
  - Recursion
- **This week's content**
  - Sequences, APIs

Sequences

- **Contain an ORDERED set of data**
- `str` **– short for a "string of text"**
- `list` **- ['a', 'group' , 'of', 'items']**
- `range(start, stop, step)`
- `tuple` **– a list that can't be modified**
- **Supports very easy iteration:**
- ```
  for item in sequence:
      print(item)
  ```

Sequence (General) Operators

- **elem** `in` **&** `not in` **sequence**
- `+` **&** `*`
- **slice** `[START:END:STEP]`
- `len()`
- `min()` **&** `max()`
- **Even** `map() filter()` **&** `reduce()`!
- `count(item)`
- **Many, many more:**
  http://docs.python.org/library/stdtypes.html#typesseq

Strings and String Operators

- **Sequence (or "list" or "array") of chars**
- **Quoting**
  - Single Quotes, Double Quotes
  - Triple Quotes (this keeps formatting and line breaks)
- **Concentration, finding length, etc.**
  - help("string")
- **Slicing Supported** `[START:END:STEP]`
- **http://docs.python.org/library/stdtypes.html#string-methods**

Lists

- **Collection of any type**
  - Including itself!
- **Indexing (**`list[item]`**)**
  - Indexed from 0, **NOT** 1, unlike S*nap!*
- **Modifying (**`list[item] = new_item`**)**
- **Slicing and slicing notation (i.e. [::])**
  - Exactly the same as string notation!
- **Operators**
  - append(x), insert(i,x), count(x), sort(), etc.
- **http://docs.python.org/library/ stdtypes.html#mutable-sequence-types**

Dictionaries

- **Very fast access (by key, not number)**
- **"Map" from a key to a value**
- **Syntax**
  - { key1 : value1, key2 : value2, … }
- **Adding elements**
  - `dict[key] = value`
- **Accessing elements;** `dict[key]`
- **Keys**
  - Looking for specific keys (has_key() & "in")
  - Iterating over (iterkeys())

API (Application Programming Interface)

- **Set of agreements for sharing information**
- **Programming APIs:**
  - "Building Blocks" for common elements such as Open or Save prompts
- **Web APIs**
  - "Special" URLs for accessing data directly
- **Example: Open Weather Map API**
  - Map: http://openweathermap.org/Maps
  - Raw data: http://api.openweathermap.org/data/2.5/weather? q=Berkeley,CA

Demo (reference)

- **Code files are all on the website**
- **midterm.py**
  - Some problems from the midterm implemented in Python
- **fractals.py**
  - Some fractals in Turtle Graphics
- **ttt.py**
  - Tic-Tac-Toe in Python
  - Uses the Games Crafters API for getting information about best moves

More Information

- **Sequences & Methods**
  - http://docs.python.org/library/stdtypes.html
- **Coding Bat (*Great* practice!)**
  - http://codingbat.com/python
- **Google's Python Class**
  - http://code.google.com/edu/languages/google-python-class/
- **Exercises (More practice!)**
  - http://code.google.com/edu/languages/google-python-class/exercises/basic.html