Python List Operations

5 Oct 2009
CMPT140
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Quiz 2 20 pts, 10 min

- Describe (in words) each of the five program structure/flow abstractions covered in class [10]
  - For each one, name the Python language constructs we learned that implement that abstraction
  - (There is one abstraction for which we haven't learned the Python; write N/A)
- Write a Python function that takes a parameter n and returns n-factorial: n! = 1*2*3*...*(n-1)*n
  - Need docstring and pre/post-conditions
  - May not import anything [10]

Quiz 2: answers #1

- Describe each of the five program structure/flow abstractions covered in class [10]
  - Sequence: one command after another. (newline)
  - Selection: either/or (if/elif/else)
  - Repetition: looping (while, for)
  - Composition: functions, subroutines (def)
  - Parallelism: doing several things at the same time (N/A)

Quiz 2: answers #2

- Write a Python function that takes a parameter n and returns n-factorial: n! = 1*2*3*...*(n-1)*n
  - Iterative solution:
    - **def factorial(n):**
      - **“””Computes n-factorial.**
      - **Pre: n must be an integer ≥ 0.**
      - **Post: returns n! (integer).”””**
      - **prod = 1**
      - **for idx in range(1,n+1):**
        - **prod *= idx**
      - **return prod**

What's on today

- Python-specific list operations
  - Membership (in)
  - Concatenate (+), repeat (*)
  - Delete (del), slice ([s:e])
  - Aliasing vs. copying lists
- Example: Using lists: Sieve of Eratosthenes

Multidimensional arrays

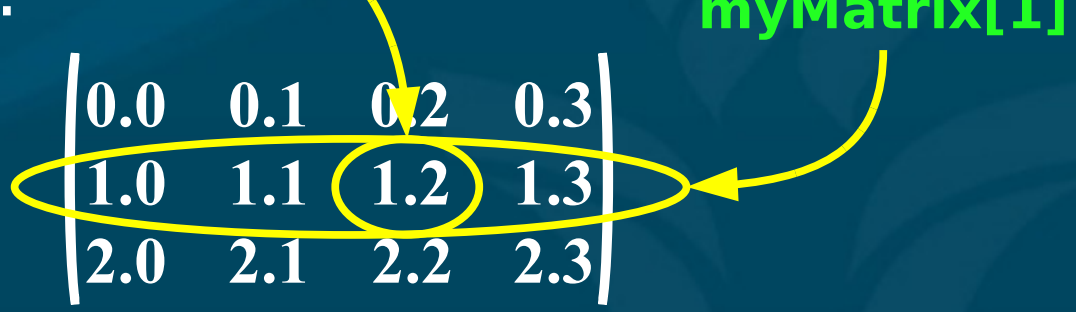
- Multidimensional arrays are simply arrays of arrays:

```
myMatrix = [ [0.0, 0.1, 0.2, 0.3],
             [1.0, 1.1, 1.2, 1.3],
             [2,0, 2.1, 2.2, 2.3] ]
```

- Accessing:

```
myMatrix[1][2] = 1.2
```

- Row-major convention:



Iterating in multidim arrays

```
def matrix_average(matrix):
    """Return the average value from the 2D
      matrix.
    Pre: matrix must be a non-empty 2D array of
      scalar values."""
    sum = 0
    num_entries = 0
    for row in range(len(matrix)):
        for col in range(len(matrix[row])):
            sum += matrix[row][col]
        num_entries += len(matrix[row])
    return sum / num_entries
```

- What if rows are not all equal length?

List operations (Python)

```
myApples = [ "Fuji", "Gala", "Red Delicious" ]
```

- Test for list membership:

```
if "Fuji" in myApples:                    # True
```

- Concatenate:

```
[ 'a', 'b', 'c' ] + [ 'd', 'e' ]
```

- Repeat:

```
[ 'a', 'b', 'c' ] * 2
```

- Modify list entries (mutable):

```
myApples[1] = "Braeburn"
```

- Convert a string to a list of characters:

```
list("Hello World!")      # ['H', 'e', 'l', 'l', 'o', ...]
```

More list operations

- Delete an element of the list:

```
del myApples[1]    # [ "Fuji", "Golden Delicious" ]
```

- List slice (start:end):

```
myApples[0:1]      # [ "Fuji" ]
```

- Assignment is aliasing:

```
yourApples = myApples      # points to same array
```

  - Changes to myAp... show up in yourAp...

- Use a whole-list slice to copy a list:

```
yourApples = myApples[:]
    # [:] is shorthand for [0:len(myApples)]
```