Announcements

- Homework 3 deadline extended to Wednesday 10/2 @ 11:59pm.
- Optional Hog strategy contest due Thursday 10/3 @ 11:59pm.
- Homework 4 due Tuesday 10/8 @ 11:59pm.
- Project 2 due Thursday 10/10 @ 11:59pm.
- Guerrilla Section 2 this Saturday 10/5 & Sunday 10/6 10am-1pm in Soda.
- Topics: Data abstraction, sequences, and non-local assignment.
- Please RSVP on Piazza!
- Guest lecture on Wednesday 10/9, Peter Norvig on Natural Language Processing in Python.

Strings

Strings are an Abstraction

Representing data:

- '200'
- '1.2e-5'
- 'False'
- '{1, 2}'

Representing language:

- """And, as imagination bodies forth
  The forms of things to unknown, and the poet's pen
  Turns them to shapes, and gives to airy nothing
  A local habitation and a name."

Representing programs:

- 'curry = lambda f: lambda x: lambda y: f(x, y)'

Strings are Sequences

Length. A sequence has a finite length.

Element selection. A sequence has an element corresponding to any non-negative integer index less than its length, starting at 0 for the first element.

String Literals Have Three Forms

```python
>>> 'I am string!
'I am string!'
>>> "I've got an apostrophe"
'I've got an apostrophe'
>>> '김영'
'김영'
>>> """The Zen of Python
claims, Readability counts.
Read more: import this.""
'The Zen of Python
claims, Readability counts.
Read more: import this.'
```

A backslash "escapes" the following character

"Line feed" character represents a new line

Single-quoted and double-quoted strings are equivalent
String Membership Differs from Other Sequence Types

The "in" and "not in" operators match substrings

>>> 'here' in 'Where's Waldo?'
True

>>> 234 in [1, 2, 3, 4, 5]
False

Why? Working with strings, we usually care about words more than characters

The count method also matches substrings

>>> 'Mississippi'.count('i')
4

>>> 'Mississippi'.count('issi')
1

The count method also matches substrings

the number of non-overlapping occurrences of a substring

Encoding Strings

Representing Strings: the ASCII Standard

American Standard Code for Information Interchange

- "Bell" (a) - ASCII Code Chart
- "Line feed" (n)

- 16 columns: 4 bits
- Rows indexed 2-5 are a useful 6-bit (64 element) subset
- Control characters were designed for transmission

(Demo)

Representing Strings: the Unicode Standard

- 109,880 characters
- 93 scripts (organized)
- Enumeration of character properties, such as case
- Supports bidirectional display order
- A canonical name for every character

U+0058 LATIN CAPITAL LETTER X
U+263a WHITE SMILING FACE
U+2639 WHITE FROWNING FACE

(Demo)

Representing Strings: UTF-8 Encoding

UTF (UCS (Universal Character Set) Transformation Format)

Unicode: Correspondence between characters and integers
UTF-8: Correspondence between those integers and bytes

A byte is 8 bits and can encode any integer 0-255.

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Integers</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>0</td>
</tr>
<tr>
<td>00000001</td>
<td>1</td>
</tr>
<tr>
<td>00000010</td>
<td>2</td>
</tr>
<tr>
<td>00000011</td>
<td>3</td>
</tr>
</tbody>
</table>

Variable-length encoding: integers vary in the number of bytes required to encode them.

In Python: string length is measured in characters, bytes length in bytes.

(Demo)
Consider two problems:

- Sum the even members of the first n Fibonacci numbers.
- List the letters in the acronym for a name, which includes the first letter of each capitalized word.

```python
enumerate naturals:
map fib:
filter even:
accumulate sum:
```

```python
enumerate words:
filter capitalized:
map first:
accumulate tuple:
```

### Mapping a Function over a Sequence

**Apply a function to each element of the sequence**

```python
>>> alternates = (-1, 2, -3, 4, -5)
>>> tuple(map(abs, alternates))
(1, 2, 3, 4, 5)
```

The returned value of `map` is an iterable map object

A constructor for the built-in map type

The returned value of `filter` is an iterable filter object

### Iterable Values and Accumulation

**Iterable objects give access to their elements in order.**

Similar to a sequence, but does not always allow element selection or have finite length.

Many built-in functions take iterable objects as argument.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tuple</code></td>
<td>Return a tuple containing the elements</td>
</tr>
<tr>
<td><code>sum</code></td>
<td>Return the sum of the elements</td>
</tr>
<tr>
<td><code>min</code></td>
<td>Return the minimum of the elements</td>
</tr>
<tr>
<td><code>max</code></td>
<td>Return the maximum of the elements</td>
</tr>
</tbody>
</table>

For statements also operate on iterable values.

### Reducing a Sequence

Reduce is a higher-order generalization of `max`, `min`, & `sum`.

```python
>>> from operator import mul
>>> from functools import reduce
>>> reduce(mul, (1, 2, 3, 4, 5))
120
```

First argument: A two-argument function

Second argument: an iterable object

Similar to accumulate from Homework 2, but with iterable objects.
Generator Expressions

One large expression that evaluates to an iterable object

\[(\text{map exp} \text{ for } \text{name} \text{ in } \text{iter exp} \text{ if } \text{filter exp})\]

- Evaluates to an iterable object.
- \text{iter exp} is evaluated when the generator expression is evaluated.
- Remaining expressions are evaluated when elements are accessed.

Short version: \((\text{map exp} \text{ for } \text{name} \text{ in } \text{iter exp})\)

(Deeo)