Strings are an Abstraction

Representing data:

'200'  '1.2e-5'  'False'  '(1, 2)'

Representing language:

"""O! methinks how slow
This old moon wanes; she lingers my desires ,
Like to a step dame, or a dowager
Long withering out a young man's revenue."""

Representing programs:

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

Representing Strings: the ASCII Standard

American Standard Code for Information Interchange

- The layout was chosen to support sorting by character code.
- Rows indexed 2-5 are a useful 6-bit (64 element) subset.
- Control characters were designed for transmission.

Representing Strings: the Unicode Standard

- 109,000 characters
- 93 scripts (organized)
- Enumeration of character properties, such as case
- Supports bidirectional display order
- 32 bits per character number
- A canonical name for every character

Representing Strings: UTF-8 Encoding

UTF: (UCS (Universal Character Set) Transformation Format)

Unicode: Correspondence between characters and 32 bit numbers

UTF-8: Correspondence between numbers 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>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</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 in characters, bytes length in bytes`

Strings are Sequences

```python
>>> city = 'Berkeley'
>>> len(city)
8
>>> city[3]  # An element of a string
'k'
```

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.

```python
>>> 'Berkeley' + ', CA'
'Berkeley, CA'
>>> 'Shabu' + 2
'Shabu Shabu'
```
String Membership Differs from Other Sequences

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

```python
>>> 'here' in "Where's Waldo?"
True
```

Why? Working with strings, we care about words, not characters

The "count" method also matches substrings

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

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

Methods on Strings

```python
>>> '1234'.isnumeric()
True
```

```python
>>> 'rOBERT dE nIRO'.swapcase()
'Robert De Niro'
```

```python
>>> 'snakeyes'.upper().endswith('YES')
True
```

String Literals Have Three Forms

```python
>>> 'I am string!'
'I am string!'
```

```python
>>> "I've got an apostrophe"
'I've got an apostrophe'
```

```python
>>> """The Zen of Python
claims, Readability counts.
Read more: import this.""
'The Zen of Python
claims, "Readability counts."
Read more: import this.'
```

```python
>>> '\n'
''
```

A backslash "escapes" the following character

"Line feed" character represents a new line

String Coercion

Any object can be "coerced" into a string.

Coercion doesn't change an object; it produces a corresponding object of a different type.

```python
>>> digits = (1, 8, 2, 8)
>>> 2 in digits
True
```

```python
>>> str(2) + ' is an element of ' + str(digits)
'2 is an element of (1, 8, 2, 8)'
```

How is string coercion implemented? October 10

```python
>>> enumerate naturals:
   0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.
```

```python
>>> map fib:
    0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.
```

```python
>>> filter iseven:
    0, 2, 4, 6, 8, 10, 12, 14, 16, 18.
```

```python
>>> accumulate sum:
    44
```

Sequences as Conventional Interfaces

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:
   0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.
```

```python
map fib:
    0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.
```

```python
filter iseven:
    0, 2, 4, 6, 8, 10, 12, 14, 16, 18.
```

```python
accumulate sum:
    44
```

Sequences as Conventional Interfaces

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 words:
    'University',  'of',  'California',  'Berkeley'
```

```python
filter iscap:
    'University',  'California',  'Berkeley'
```

```python
map first:
    'U',  'C',  'B'
```

```python
accumulate tuple:
    { 'U',  'C',  'B' }
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.

```
Demo
```

Accumulation and Iterable Values

Iterable objects give access to some elements in order.

Many built-in functions take iterable objects as argument.

```
tuple    Return a tuple containing the elements
sum      Return the sum of the elements
min      Return the minimum of the elements
max      Return the maximum of the elements
```

For statements also operate on iterable values.

```
Demo
```

Generator Expressions

One large expression that evaluates to an iterable object

```python
(map exp) for <name> in <iter exp> if <filter exp>)
```

• Evaluates to an iterable object.
• `<iter exp>` is evaluated when the generator expression is evaluated.
• Remaining expressions are evaluated when elements are accessed

```python
(map exp) for <name> in <iter exp>)
```

Precise evaluation rule introduced in Chapter 4.

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
```

Similar to accumulate from Homework 2

```
Demo
```