More on sequence types
The `range` function is useful in `for`-loops

```
for i in range(1, 10, 2):
    print i*i
```

- Repeats the execution of the body of the `for`-loop for each value of $i = 1, 3, 5, 7, \text{and} \ 9$.

- Equivalent to
  ```
i = 1
while i < 10:
    print i*i
    i = i + 2
  ```

- But more convenient for simple loops because no need to initialize before loop and no need to update within loop.
More examples of for-loops

L = ["hello", "hi", "bye"]
for e in L:
    print e + e

s = "What is this sentence?"
for ch in s:
    print ch
The map function

- **map(f, [a, b, c, d, e])** returns the list \([f(a), f(b), f(c), f(d), f(e)]\)
- The first argument of **map** is a function \(f\) and the second argument is a list \(L\); it returns a new list obtained by applying \(f\) onto every element of \(L\).

**Examples:**
- **map(round, [4.57, -9.876, math.pi])** returns \([5.0, -10.0, 3.0]\)
- **map(str, range(0, 6))** returns \(['0', '1', '2', '3', '4', '5']\)

- The **map** function allows us to construct new lists from old ones.
The **filter** function

- \texttt{filter(f, L)} returns a sublist of \texttt{L} consisting of those elements in \texttt{L} (in the same order as they appear in \texttt{L}) for which the boolean function \texttt{f} evaluates to \texttt{True}.

**Examples:**
- \texttt{filter(bool, [0, -10, 0.0, None, "hello"])} returns \texttt{[-10, 'hello']}
- \texttt{filter(containsSeven, map(str, range(1001))]} returns a list containing all of the numbers in the range 0 through 1000 that contain 7.
Operations that work on strings and lists

1. `x in s`, `x not in s`

2. `s + t`, `s*n`, `n*s`

3. `s[i]`, `s[i:j]`, `s[i:j:k]`

4. `len(s)`, `min(s)`, `max(s)`

5. `s.index(i)`, `s.count(i)`
Problem 1

- A positive integer $n$ is *perfect* if the sum of its factors (excluding itself) is equal to $n$.

**Example**: 6 is perfect because $1 + 2 + 3 = 6$.

- Write a program that finds all perfect numbers between 1 and 10,000.
Useful string operations

1. `str.find(s)`

2. `str.isalnum()`, `str.isalpha()`, `str.isdigit()`, `str.islower()`, `str.isupper()`, etc.

3. `str.upper()`, `str.lower()`

4. `str.split()`

5. `str.replace(old, new)`
Problem 2

- You are given a list of words. You are required to write a program that counts the number of times each word occurs in some input text.