Storing Data

In Memory/CPU
- Transient (exists while program is running)
- Limited size

On Disk
- Persistent
- Larger capacity
- Text files, csv files, databases, etc

Diagram:
- Central Processing Unit (CPU)
- Random Access Memory (RAM)
- Disk
Storing Data

Text Files: Streams of characters

CSV Files: Comma separated values

Databases: Tables of data supporting highly efficient operations

(CSE 560 Data Models and Query Languages; CSE 562 Database Systems)
import sqlite3

conn = sqlite3.connect('test.db')
cur = conn.cursor()

# do things to database

conn.commit()
conn.close()
Example Commands

```python
cur.execute('CREATE TABLE IF NOT EXISTS movies (title, director, year)')

cur.execute('INSERT INTO movies VALUES ("Jaws", "Spielberg", 1975)')

results = cur.execute('SELECT * FROM movies')

results = cur.execute('SELECT * FROM movies WHERE year = 1975')
```
Parameterized Commands

How can we create commands from variables?
How can we create commands from variables?

We could build up a string using multiple concatenations...but that would be tedious (and cause other issues we will see in the future)
Parameterized Commands

How can we create commands from variables?

We could build up a string using multiple concatenations...but that would be tedious (and cause other issues we will see in the future)

SQLite let us parameterize our commands!
def insert(title, director, year):
    cur.execute('INSERT INTO movies VALUES (?,?,?)', (title, director, year))

def get_all_by_year(year):
    return cur.execute('SELECT * FROM movies WHERE year=?', (year,))
def insert(title, director, year):
    cur.execute('INSERT INTO movies VALUES (?,?,?)', (title, director, year))

def get_all_by_year(year):
    return cur.execute('SELECT * FROM movies WHERE year=?',(year,))
Parameterized Commands

def insert(title, director, year):
    cur.execute('INSERT INTO movies VALUES (?,?,?)', (title, director, year))

def get_all_by_year(year):
    return cur.execute('SELECT * FROM movies WHERE year=?', (year,))

We pass a tuple with the specific values
Parameterized Commands

def insert(title, director, year):
    cur.execute('INSERT INTO movies VALUES (?,?,?)', (title, director, year))

def get_all_by_year(year):
    return cur.execute('SELECT * FROM movies WHERE year=?', (year,))

Tuples in Python:
(x,) ← Tuples of size one (notice the comma)
(x,y) ← Tuples of size two
(x,y,z) ← Tuples of size three (or more...as many values as we want)
Big Example - Music Rating App

MusicRater1.0
- Python server and JS/HTML client for rating songs
- Songs and ratings stored in CSV files

MusicRater2.0
- Python server and JS/HTML client for rating songs
- Songs and ratings stored in SQLite Database