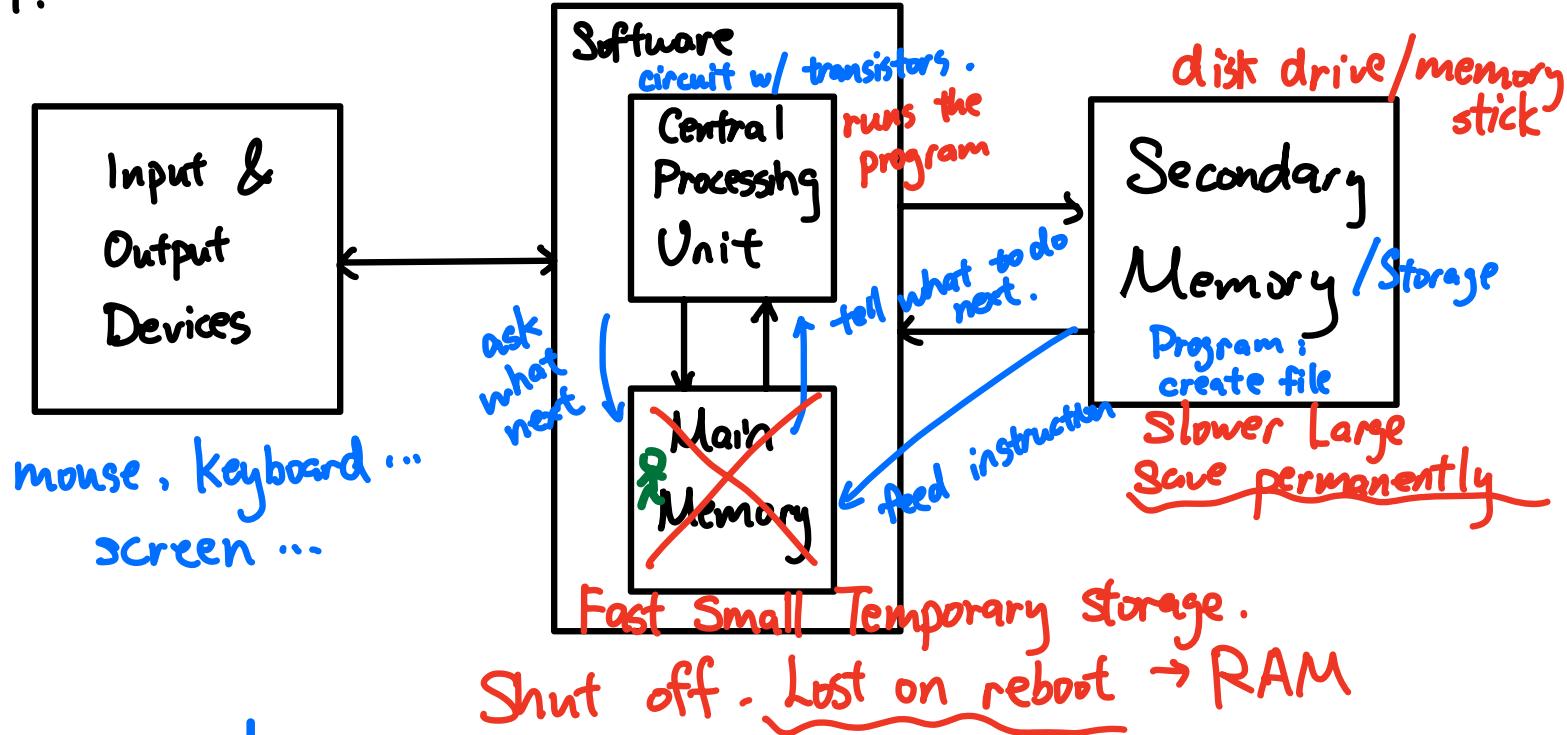


Python For everybody

1.



2. $x = x + 1$

先 evaluate 右边. 再看左边.

3. $\begin{cases} ** & \text{power.} \\ \% & \text{remainder.} \end{cases}$

operator precedence.

$() \rightarrow ** \rightarrow * / \rightarrow + -$
→ left to right

4. `type(var)`. return var's type.

Type Conversing : `float(99)`. `int('99')`

* Division provides floating result. Python 2.
Could have an error if can't convert.

5. User input: `msg = input('...')`

- 6 Try - Except: 有点类似 if let, else
eg. try :
 sth. isstr = int(astr)
except :
 sth. kstr = l. 给予 error detection.
7. quit() 直接退出 script
8. def : define function.
max(), min() : return max/min num/ch
return { stop the func
 } determine residual value
9. break : end the current loop and jumps to
the statement immediately following the loop.
escape the loop.
- continue : quit the current, start the next one
10. Loop : Set initial value.
for thing in data :
 update & check
11. None :
eg. smallest = None
if smallest is None :
12. is.
 type & val. 有点类似 Nil.
 stranger ==
 is not !=

Python Data Structure

I. String.

index Start w/ 0.

python error : index out of range.

`len (str)` return length .

Loop through string :

`for char in str:` definite loop.

| while index < len(str) : index = index + 1
| **indefinite loop.**

Slicing: str[0:3]. ⇒ only show 0, 1, 2

up to but not including.

`str[0 : ∞]` ⇒ okay! show the string,
|| no error.

str [ɔ:], str [ə:].

2. String Modification .

'char' in sth : likes "=". return T/F.

`str >/< /:= str : >/< . Upper/Lower case G > c.`

`str.lower()` , `str.upper()` object method .

`dir(str)` : directions , show all methods on this type .

`str.find (char) :` return position index found.

. find (char, start position) -1 Not found

`str.replace(str1, str2)` : replace str1 w/ str2.

replace all

{ `strip()` : remove whitespaces both end
 `lstrip()` : left
 `rstrip()` : right

String - Unicode in python 3

No need for conversion. Better character set.

3. Files.

`handle = open(fileName, mode)` ↑
 file connection.
open, read, write, close. sequence of str.
`\n` : new line. single character.

4. Process Files :

for line in handle: print(line).

→ line.rstrip()

↓ add newline at the end.

if not line.startswith('sth'): continue. skip.
try: # open file
except: print("Error")
quit()

Compensate for bad file

5. { Algorithm: A set of rules/steps to solve a problem
 Data Structure: A particular way of organizing data in computer.

Collection: ≥ 1 value in 1 variable

{ List: [...]. contain anything. mutable
 String literal immutable

range(list): return a list of numbers

range(4): [0, 1, 2, 3] ⇒ iterator.

eg. for i in range(len(str)): counted loop.

Slicing (like String)

[0 : 3] \Rightarrow 0, 1, 2

up to but NOT including.

{ list() : create list
list.append(sth) : append a new item. / remove()
sth in list : .contains(). Bool T/F.
not in
list.sort() : sort the list
list.insert(sth)
len(), max(), min(), sum()

6. String split into list.

list = str.split() whitespace as delimiter
.split(', ')

定界符.

7. Dictionary. sth = dict(). key-value pair

sth[key] = value

Dict : {key: value, ...}. {}.

if keyVal in dict:
dict[keyVal] += 1
else:
dict[keyVal] += 1.

dict.get(key, defaultVal)

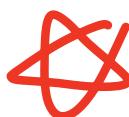
get Value
OR

set default

for key in dict: go through key set.

list(dict) = dict.keys().

dict.values(), dict.items()



list of tuples

for key, value in dict.items():

8. Tuples. = (...). immutable.
list(tuple) => mutable.

(x, y) = (4, 'sth') multiple assignments

tups = dict.items().

eg. (0, 1, 2) < (5, 1, 2) True

(0, 1, 99) < (0, 3, -1) True



ONLY Look NOT-MATCHES

sorted(dict.items()). => sort in KEY orders

* 可以通过 tmp.append((value, key))
sorted(tmp) => sort in VALUE orders.

* sorted((v, k) for k, v in c.items()) ≤
sorted(sth, reverse = TRUE) ≥.

From book

1. import math.

math.log10(sth).

2. import random

random.random() # [0.0, 1.0)

`random.randint(5, 10) # [5, 10].`

`random.choice([1, 2, 3]) # choosing element.`

3. `delimiter.join(str-list).`

4. { .append modify
+ Create a new list

Q. `line.translate(str.maketrans(fromstr, tostr, delestr))`

replace the char at the same idx.
delete.

6. import string.

`string.punctuation # ' ! " # $ % ... '`

eg. `line.translate(line.maketrans(' ', ' ', string.punc))`
delete all punctuation.

7. immutable ≈ hashable.

eg. tuple ✓ hashable.
list ✗ hashable

Access Web Data

1. Regular Expression.

import re

{ re. search() ≈ find()
| re..findall()

re. search (pattern, line)

e.g. startswith() = re.search ('^ start ', line)

Regular Expression Quick Guide

^	Matches the beginning of a line
\$	Matches the end of the line
.	Matches any character
\s	Matches whitespace
\S	Matches any non-whitespace character
*	Repeats a character zero or more times
*?	Repeats a character zero or more times (non-greedy)
+	Repeats a character one or more times
+?	Repeats a character one or more times (non-greedy)
[aeiou]	Matches a single character in the listed set
[^XYZ]	Matches a single character not in the listed set
[a-z0-9]	The set of characters can include a range
(Indicates where string extraction is to start
)	Indicates where string extraction is to end

dot .
matches all
characters.

wild card

* matches
ALL

{ ^ _ not underscore
| ^ not whitespace

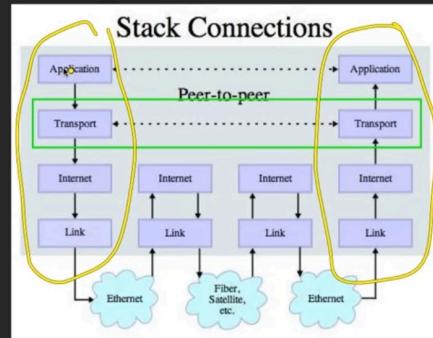
{ \ escape char.
| \\$, \ ...

⚠ string list ⇒ int list .
map(int , str_list)

2. Networked Technology.

Transport Control Protocol (TCP)

- Built on top of IP (Internet Protocol)
- Assumes IP might lose some data - stores and retransmits data if it seems to be lost
- Handles “flow control” using a transmit window
- Provides a nice reliable **pipe** and there's application, talks to a transport layer, Internet Protocol Suite



TCP Connections / Sockets

“In computer networking, an Internet **socket** or network **socket** is an endpoint of a bidirectional **inter-process** communication flow across an **Internet** Protocol-based computer network, such as the **Internet**.”



TCP Port Numbers

- A port is an **application-specific** or process-specific software communications endpoint
- It allows multiple networked applications to coexist on the same server.
- There is a list of well-known TCP port numbers

HTTP : 80.

 import socket

Python has built-in support for TCP Sockets

```
import socket
mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect( ('data.pr4e.org', 80) )
```

Host → Port

TCP/IP

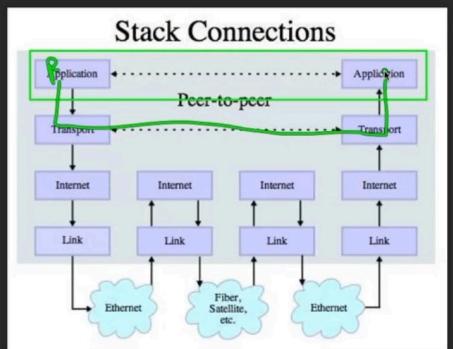
Application Protocol

- Since TCP (and Python) gives us a reliable **socket**, what do we want to do with the **socket**? What problem do we want to solve?

- Application Protocols

- Mail

- World Wide Web
 - we made a connection with a socket and then connect it



allows browsers to retrieve web docs. from internet servers.

HTTP - Hypertext Transfer Protocol

- The dominant Application Layer Protocol on the Internet
- Invented for the Web - to Retrieve HTML, Images, Documents, etc
- Extended to be data in addition to documents - RSS, Web Services, etc..Basic Concept - Make a Connection - Request a document - Retrieve the Document - Close the Connection

https://www.baidu.com/sth.html
protocol host document

An HTTP Request in Python

```
import socket

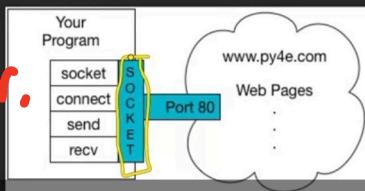
mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect(('data.pr4e.org', 80))
cmd = 'GET http://data.pr4e.org/romeo.txt HTTP/1.0\r\n\r\n'.encode()
mysock.send(cmd)

while True:
    data = mysock.recv(512)
    if (len(data) < 1):
        break
    print(data.decode())
mysock.close()
```

receive up to 512 char.

That's like a matrix thing too,

↓
string



status code :
302 : post. provide location header, redirect immedi
200 : give a doc.
404 : Not found.

In the metadata in the header, it tells content-type. knows how to interpret .

The Request / Response Cycle .

Which application talks first? Client or Server?

3. Retrieve Web Page.

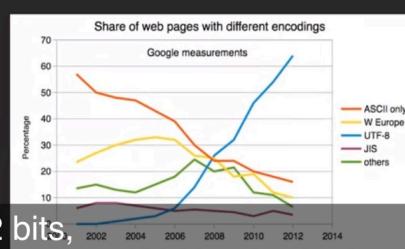
ASCII: `ord(char)` # provide ASCII
Unicode for diff. lang.

Multi-Byte Characters

- To represent the wide range of characters computers must handle we represent characters with more than one byte

- UTF-16 – Fixed length - Two bytes
- UTF-32 – Fixed Length - Four Bytes
- UTF-8 – 1-4 bytes
 - Upwards compatible with ASCII
 - Automatic detection between ASCII and UTF-8
 - UTF-8 is recommended practice for encoding data to be exchanged over the web.

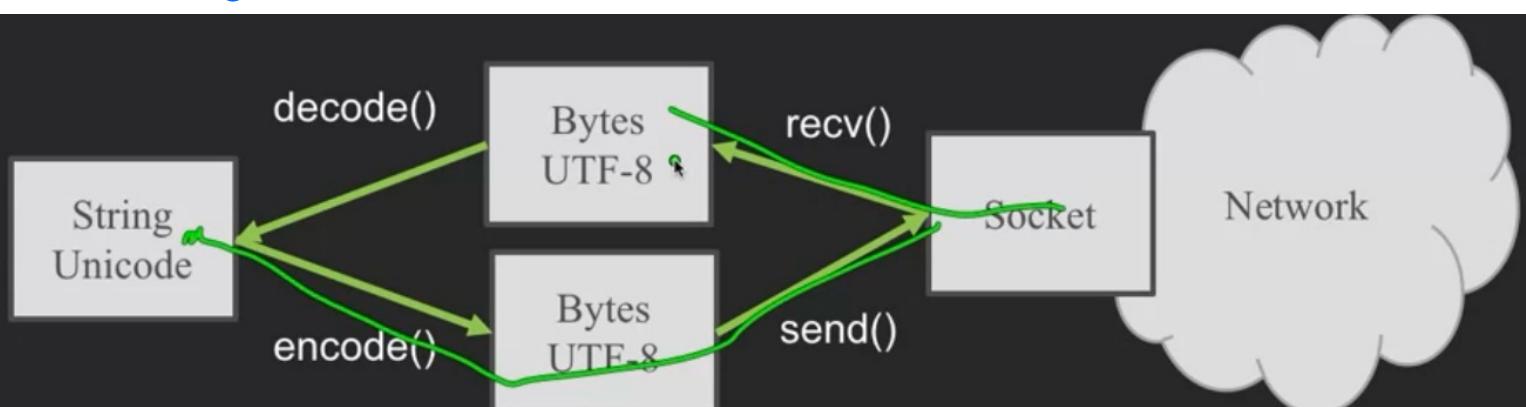
<https://en.wikipedia.org/wiki/UTF-8>



Python 3:
All str \Rightarrow Unicode
type = String.

* { Python 2 : byte str = str.
Python 3 : str = unicode.

* `str.encode()` \Rightarrow byte array
`data.decode()` \Rightarrow string
 ↓
 bytes



4. handle web request

Using `urllib` in Python

Since HTTP is so common, we have a library that does all the socket work for us and makes web pages look like a file

```
import urllib.request, urllib.parse, urllib.error  
fhand = urllib.request.urlopen('http://data.pr4e.org/romeo.txt')  
for line in fhand:  
    print(line.decode().strip()) => body data  
        some library bits and then we say,  
        okay, urllib.request.urlopen.  
                                            urllib1.py
```

* **Beautiful Soup - Search String.**
`from bs4 import BeautifulSoup`

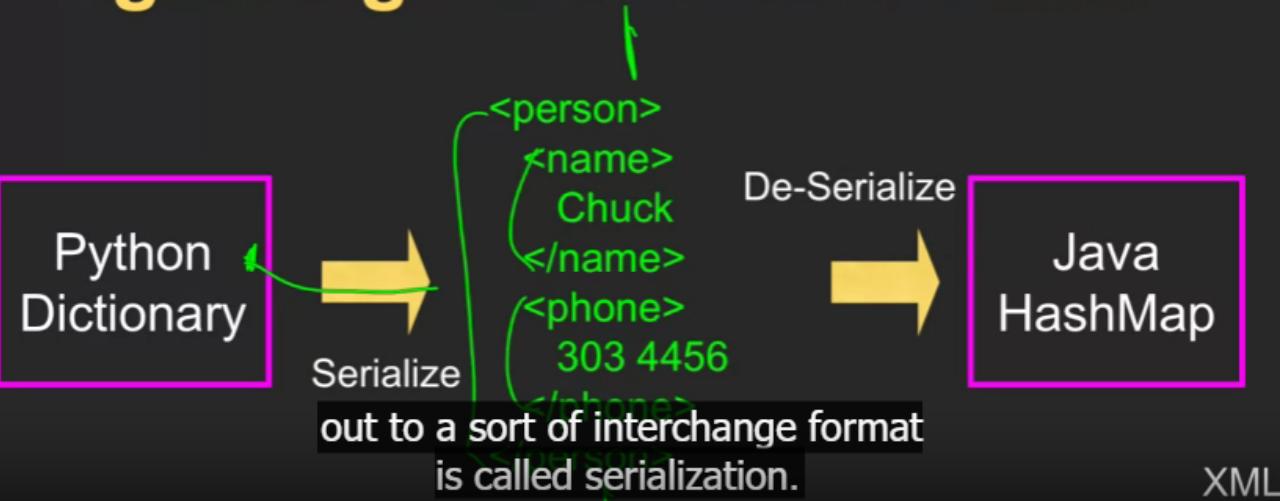
```
import urllib.request, urllib.parse, urllib.error  
from bs4 import BeautifulSoup  
  
url = input('Enter - ')  
html = urllib.request.urlopen(url).read()  
soup = BeautifulSoup(html, 'html.parser')  
  
# Retrieve all of the anchor tags  
tags = soup('a')  
for tag in tags:  
    print(tag.get('href', None))
```



python urllinks.py
Enter - http://www.dr-chuck.com/page1.htm
http://www.dr-chuck.com/page2.htm
That's what this parsing is doing.

* The TCP/IP provides pipes / sockets between applications.

Agreeing on a “Wire Format”



5. eXtensible Markup Language (XML)

可扩展 标记 语言。

{ HTML : 显示网页数据 ⇒ 预定义标签.
| XML : 交换、传输数据. ⇒ 自定义标签.

↳ { Simple Element
Complex Element : has child node
Share Structured data .

XML Basics

- Start Tag
- End Tag
- Text Content
- Attribute
- Self Closing Tag

```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```

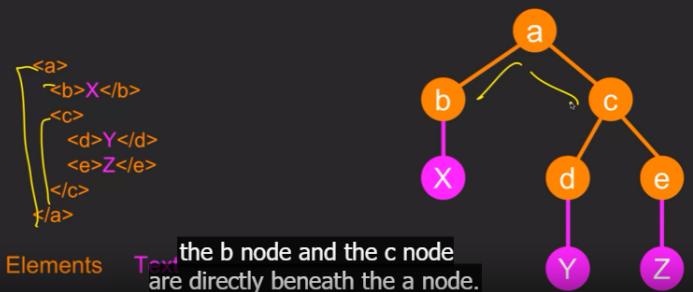
XML Terminology

- Tags indicate the beginning and ending of elements
- Attributes - Keyword/value pairs on the opening tag of XML
- Serialize / De-Serialize - Convert data in one program into a common format that can be stored and/or transmitted between systems in a programming language-independent manner

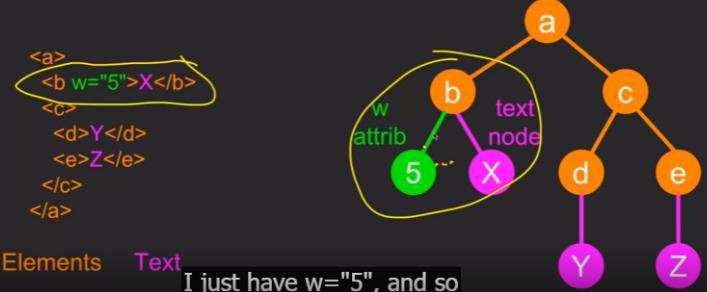
Serialization and deserialization is the act of taking from an internal structure

{ 1a/1b X
1a/1c/1d Y
1a/1c/1e Z .

XML as a Tree



XML Text and Attributes



* **XML Schema**: a way to establish outside of program, and the separately check on send/receive ends. **resolve disagreements.**

XML Schema

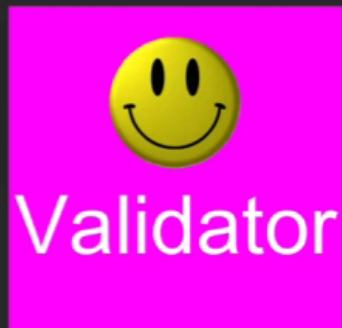
- Description of the **legal format** of an XML document
- Expressed in terms of constraints on the structure and content of documents
- Often used to specify a “**contract**” between systems - “My system will only accept XML that conforms to this particular Schema.”
- If a particular piece of XML meets the specification of the Schema - it is said to “**validate**” And validation is not the act of transferring the data or ^{http://} _{data}

XML Document

```

<person>
  <lastname>Severance</lastname>
  <age>17</age>
  <dateborn>2001-04-17</dateborn>
</person>
  
```

XML Validation



XML Schema Contract

```

<xs:complexType name="person">
  <xs:sequence>
    <xs:element name="lastname" type="xs:string"/>
    <xs:element name="age" type="xs:integer"/>
    <xs:element name="datebo<xs:type name="date"/>>
  </xs:sequence>
</xs:complexType>
  
```

All those kinds of things

are the questions that are being asked by

DTD → SGML → XML
Doc. Type. Def → Stand. General. ML → XML

W3C (World Wide Web Consortium) = xsd

ISO 8601 Date/Time Format

2002-05-30T09:30:10Z

↑
Year-month-day ↑
Time of day ↑
Timezone - typically
specified in UTC / GMT
rather than local time
zone.

And like I said we tend to do
everything in absolute time.
http://en.wikipedia.org/wiki/ISO_8601

6. import xml.

```
import xml.etree.ElementTree as ET
data = '''<person>
    <name>Chuck</name>
    <phone type="intl">
        +1 734 303 4456
    </phone>
    <email hide="yes"/>
</person>'''

tree = ET.fromstring(data)
print('Name:', tree.find('name').text)
print('Attr:', tree.find('email').get('hide'))
```

content attr

But just to make these simple
on one screen I've kept it simple.

topLevel = ET.fromstring(data)

```
yangling@yangling-OptiPlex-5090:~/Documents$ python yangling@yangling-OptiPlex-5090:~/Documents$ python
import xml.etree.ElementTree as ET      yang.li@yangli-OptiPlex-5090:~/Documents$ python

input = '''
<stuff>
    <users>
        <user x="2">
            <id>001</id>
            <name>Chuck</name>
        </user>
        <user x="7">
            <id>009</id>
            <name>Brent</name>
        </user>
    </users>
</stuff>'''

stuff = ET.fromstring(input)
lst = stuff.findall('users/user')
print('User count:', len(lst))

for item in lst:
    print('Name', item.find('name').text)
    print('Id', item.find('id').text)
    print('Attribute', item.get('x'))
```

list of tags = tree.findall('class / item')

7. JSON (JavaScript Object Notation)

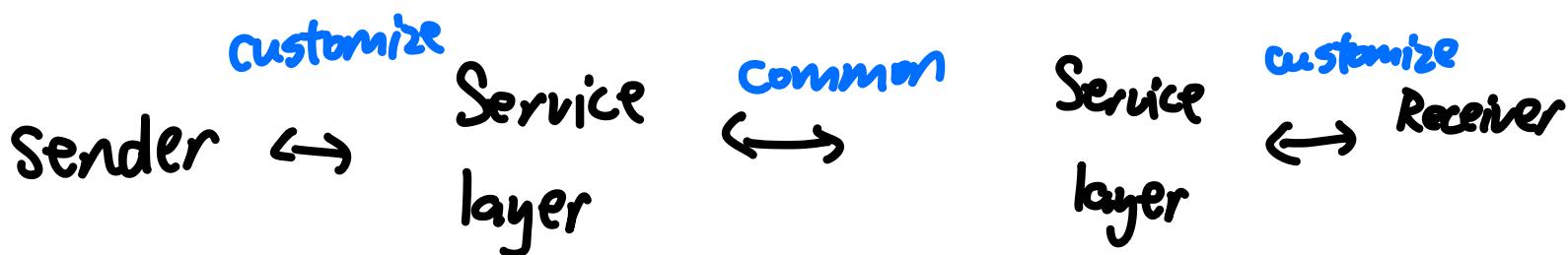
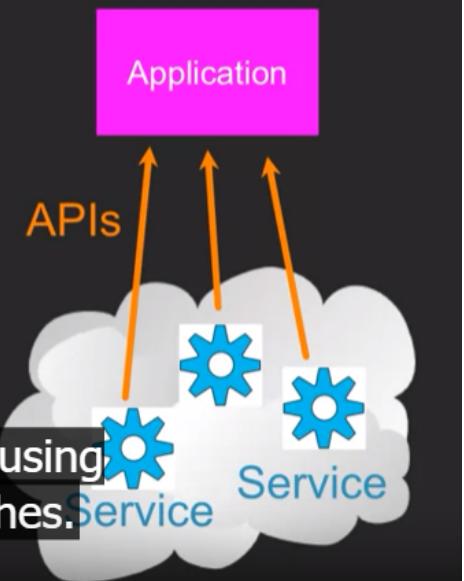
import json

json.loads(data) ⇒ info["key"]
python di

8. Service-oriented architecture.

Service Oriented Approach

- Most non-trivial web applications use services
- They use services from other applications
 - Credit Card Charge
 - Hotel Reservation systems
- Services publish protocols to access data on systems, using well-defined and structured approaches.



9. Application Programming Interface (API)

```
import urllib.request, urllib.parse, urllib.error
import json

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'

while True:
    address = input('Enter location: ')
    if len(address) < 1: break

    url = serviceurl + urllib.parse.urlencode({'address': address})

    print('Retrieving', url)
    uh = urllib.request.urlopen(url)
    data = uh.read().decode()
    print('Retrieved', len(data), 'characters')

    try:
        js = json.loads(data)
    except:
        js = None

    if not js or 'status' not in js or js['status'] != 'OK':
        print('==== Failure To Retrieve ====')
        print(data)
        continue

    lat = js["results"][0]["geometry"]["location"]["lat"]
    lng = js["results"][0]["geometry"]["location"]["lng"]
    print(lat, lng)
    print(location = js["results"][0]["formatted_address"])
    print(location)
```

geojson.py

that goes into the string address.

```

import urllib.request, urllib.parse, urllib.error
import twurl
import json

TWITTER_URL = 'https://api.twitter.com/1.1/friends/list.json'

while True:
    print('')
    acct = input('Enter Twitter Account: ')
    if (len(acct) < 1): break
    url = twurl.augment(TWITTER_URL,
                         {'screen_name': acct, 'count': '5'})
    print('Retrieving', url)
    connection = urllib.request.urlopen(url)
    data = connection.read().decode()
    headers = dict(connection.getheaders())
    print('Remaining', headers['x-rate-limit-remaining'])
    js = json.loads(data)
    print(json.dumps(js, indent=4))

for u in js['users']:
    print(u['screen_name'])
    s = u['status']['text']
    print(' ', s[:50])

import urllib
import oauth
import hidden
# 45M ✓
def augment(url, parameters):
    secrets = hidden.oauth()
    consumer = oauth.OAuthConsumer(secrets['consumer_key'], secrets['consumer_secret'])
    token = oauth.OAuthToken(secrets['token_key'], secrets['token_secret'])
    oauth_request = oauth.OAuthRequest.from_consumer_and_token(consumer,
                                                               token=token, http_method='GET', http_url=url, parameters=parameters)
    oauth_request.sign_request(oauth.SignatureMethod_HMAC_SHA1(), consumer, token)
    return oauth_request.to_url()

https://api.twitter.com/1.1/statuses/user_timeline.json?count=2
&oauth_version=1.0&oauth_token=101...SGI&screen_name=drchuck&oa
uth_nonce=09239679&oauth_signature=301...5644&oauth_signature_method=H
MAC-SHA1

```

security stuff.

→ headers.
limit

Of course we need JSON, and so

JSON = json.dump(str)
more succinct.
less self-describing

Databases

1. SQL (Structured Query Language)
2. Object - Oriented
3. Inheritance — subclasses.

e.g. class SubClass (ParentClass):
 extends Parent Class

Definitions



- **Class** - a template
- **Attribute** – A variable within a class
- **Method** - A function within a class
- **Object** - A particular instance of a class
- **Constructor** – Code that runs when an object is created
- **Inheritance** - The ability to extend a class to make a new class.

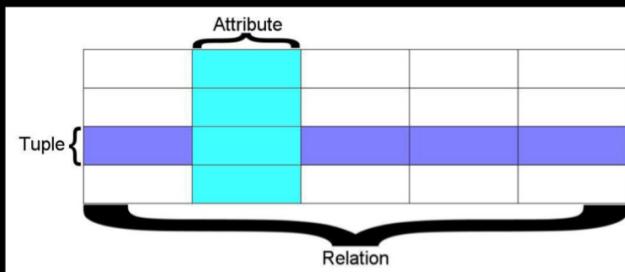
Summary

- Object Oriented programming is a very structured approach to code reuse
- We can group data and functionality together and create many independent instances of a class

4. **Relational Database**: store rows & cols. in tables.

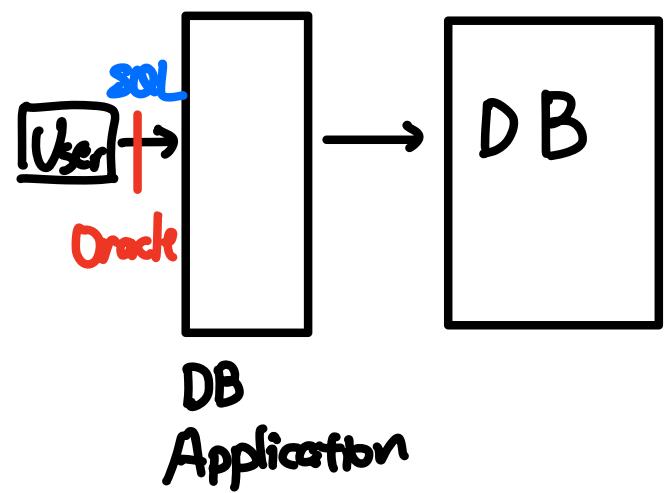
Terminology

- **Database** - contains many tables
- **Relation (or table)** - contains tuples and attributes
- **Tuple (or row)** - a set of fields that generally represents an “object” like a person or a music track
- **Attribute (also column or field)** - one of possibly many elements of data corresponding to the object represented by the row



A relation is defined as a set of tuples that have the same attributes. A tuple usually represents an object and information about that object. Objects are typically physical objects or concepts. A relation is usually described as a table, which is organized into rows and columns. All the data referenced by an attribute are in the same domain and conform to the same constraints.
(Wikipedia)

* First row: Schema metadata



CRUD

Database Model

A **database model** or **database schema** is the **structure or format of a database**, described in a formal language supported by the database management system. In other words, a "database model" is the application of a data model when used in conjunction with a database management system.

http://en.wikipedia.org/wiki/Database_model

5. SQLite - embedded db

* Schema = contract . agreement



C: `INSERT INTO Table (Attr) VALUES (data, data)`
R: `SELECT * FROM Table WHERE cond.`
U: `UPDATE Table SET NewAttr. WHERE cond.`
D: `DELETE FROM Table WHERE cond.`

`SELECT * FROM Table ORDER BY attr. ^ DESC.`

SQL Summary

```
INSERT INTO Users (name, email) VALUES ('Kristin', 'kf@umich.edu')

DELETE FROM Users WHERE email='ted@umich.edu'

UPDATE Users SET name="Charles" WHERE email='csev@umich.edu'

SELECT * FROM Users

SELECT * FROM Users WHERE email='csev@umich.edu'

SELECT * FROM Users ORDER BY email
```

6. Python SQLite .

* Create :
`CREATE TABLE name(`
attr cond
eg. `VARCHAR(128)`
)

```

1 import sqlite3      yang.li@duke.edu, 21 hours ago • Python: Finished Week 4
2
3 conn = sqlite3.connect('emaildb.sqlite')
4 cur = conn.cursor() → handle
5
6 cur.execute('DROP TABLE IF EXISTS Counts')
7
8 cur.execute('''
9     CREATE TABLE Counts (email TEXT, count INTEGER)''')
10
11 fname = input('Enter file name: ')
12 if len(fname) < 1: fname = 'mbox-short.txt'
13 fh = open(fname)
14 for line in fh:
15     if not line.startswith('From: '): continue
16     pieces = line.split()
17     email = pieces[1]
18     cur.execute('SELECT count FROM Counts WHERE email = ? ', (email,))
19     row = cur.fetchone() → fetch one row.
20     if row is None:
21         cur.execute('''INSERT INTO Counts (email, count)
22                         VALUES (?, 1)''', (email,))
23     else:
24         cur.execute('UPDATE Counts SET count = count + 1 WHERE email = ?',
25                     (email,))
26     conn.commit() commit the change. slowest.
27
28 # https://www.sqlite.org/lang_select.html
29 sqlstr = 'SELECT email, count FROM Counts ORDER BY count DESC LIMIT 10'
30
31 for row in cur.execute(sqlstr):
32     print(str(row[0]), row[1])
33
34 cur.close()

```

? placeholder

7. DB Normalization (3NF)

Don't replicate data. Add keys.

Three Kinds of Keys

- Primary key - generally an integer auto-increment field
- Logical key - What the outside world uses for lookup
- Foreign key - generally an integer key pointing to a row in another table

Album
id
title
artist_id
...

Key Rules

Best practices

- Never use your logical key as the primary key
- Logical keys can and do change, albeit slowly
- Relationships that are based on matching string fields are less efficient than integers

User
id
login
password
name
email
created_at
modified_at
login_at

* Logic key - Lookup
Primary key :
 eg. Integer NOT NULL
PRIMARY KEY
AUTOINCREMENT
UNIQUE.

8. JOIN link across multiple tables.

Album		
id	artist_id	title
1	2	Who Made Who
2	1	IV

Artist	
id	name
1	Led Zepplin
2	AC/DC

title	name
1 Who Made Who	AC/DC
2 IV	Led Zepplin

select Album.title, Artist.name from Album join Artist on Album.artist_id = Artist.id

What we want
to see

The tables that
hold the data

How the tables
are linked

	title	genre_id	id	name
1	Black Dog	1	1	Rock
2	Black Dog	1	2	Metal
3	Stairway	1	1	Rock
4	Stairway	1	2	Metal
5	About to Rock	2	1	Rock
6	About to Rock	2	2	Metal
7	Who Made Who	2	1	Rock
8	Who Made Who	2	2	Metal

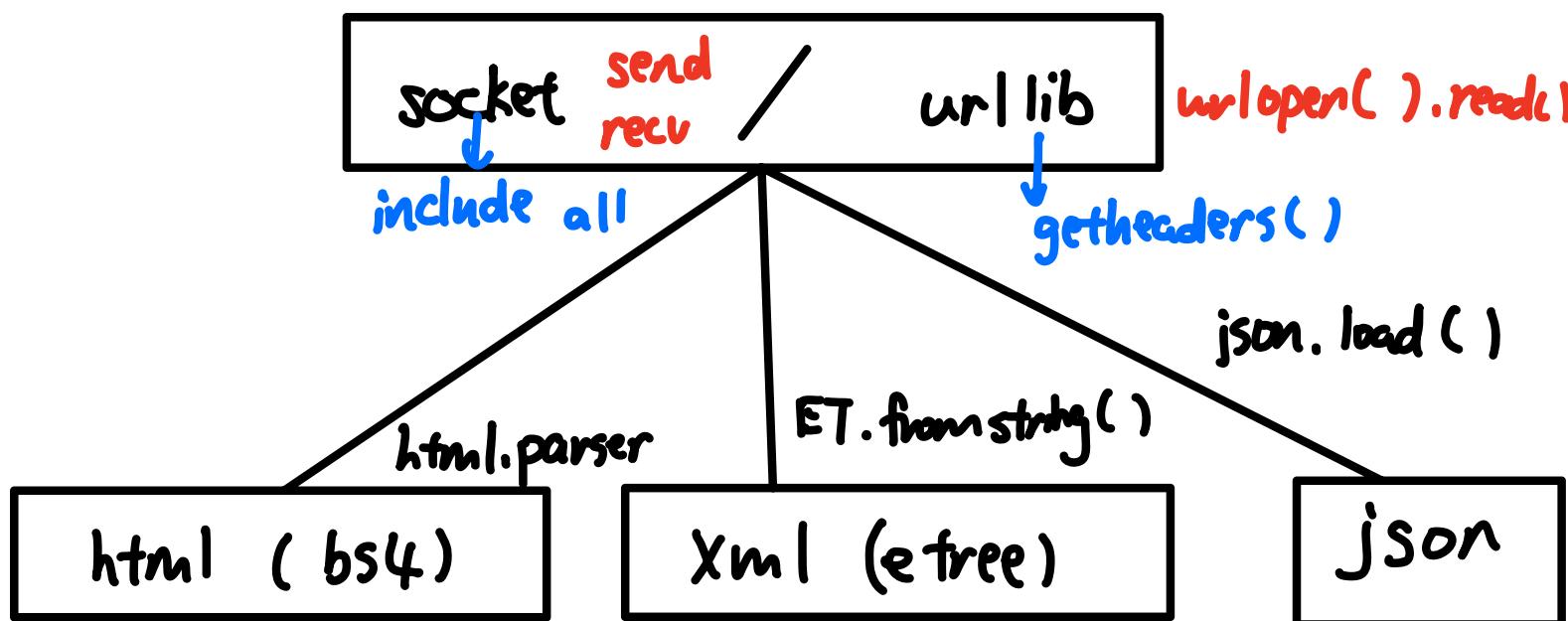
SELECT Track.title,
Track.genre_id,
Genre.id, Genre.name
FROM Track JOIN Genre
排列组合.

Joining two tables without an
ON clause gives all possible
combinations of rows.

※ INSERT OR IGNORE
REPLACE

※ s.execute_script (...) ⇒ multiple SQL stats.

1. execute (one) \Rightarrow ONE Stmt.



DB :

- Con . connect()
- Cur . cursor()
- Cur . execute /executescript()
- Con . commit()
- Con . close()

* import time.
time.sleep(5)

Cur . fetchone(), .fetchall()
.fetchmany()