**Python for Everyone: Notes**

**Chapter 13: Using Web Services**

* Data on the web
  + With the http request/response well understood and well supported, there was a natural move toward exchanging data between programs using these protocols
  + We needed to come up with an agreed way to represent data going between applications and across networks
  + There are two commonly used formats: XML and JSON
* Sending data across the “net”
  + “wire protocol” what we send to the wire
* Agreeing on a wire format
  + Python -> serialize -> deserialize -> java hashmap
* XML
  + Marking up data to send across the network
* XML elements or nodes
  + Simple elements
  + Complex element
* eXtensible Markup Language
  + primary purpose is to help information system share structured data
  + it started as a simplified subset of the standard generalized markup language (SGML)
  + and is designed to be relatively human legible
* XML basics
  + Start tag
  + End tag
  + Text content
  + Attribute
  + Self closing tag
* White space
  + Line ends do not matter
  + White space is generally discarded on text elements.
  + We indent only to be legible
* XML terminology
  + Tags indicate the beginning and ending of elements
  + Attributes – keyword/value pairs on the opening tag of xml
  + Serialize/deserialize – convert data in one program into a common format that can be stored and/or transmitted between systems in a programming language independent manner
* XML as a tree
  + Elements
  + Text
* XML schema
  + Describing a contract as to what is acceptable xml
* 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
* Many xml schema languages
  + Document type definition (DTD)
  + Standard generalized markup language (SGML)
  + XML schema from w3c (XSD)
* XSD XML schema (w3c spec)
  + We will focus on the world wide web consortium version
  + It is often called “w3c schema” because “schema” is considered generic
  + More commonly it is called xsd because the file names end in .xsd
* Javascript object notation
  + Object literal notation in javascript
  + JSON represents data as nested lists and dictionaries
* Service oriented approach
  + Most non trivial web application use services
  + They use services from other applications
    - Credit card charge
    - Hotel reservation systems
  + Services publish the rules applications must follow to make use of the service (api)
* Multiple systems
  + Initially – two systems cooperate and split the problem
  + As the data / service becomes useful – multiple accplications want to use the information / application
* Web services
* Application program interface
  + The api itself is largely abstract in that it specifies an interface and controls the behavior of the object specified in that interface. The software that provides the functionality described by an api is said to be an implementation of the api. An api is typically defined in terms of the programming language used to build an application
* Api security and rate limiting
  + The compute resources to run these api are not free
  + The data provided by these api is usually valuable
  + The data providers might limit the number of request per dat, demand an api key, or might charge for usage
  + They might change rules as time progress