* Course Overview
* Module Overview
* Prerequisites and Course Outline
* HTTP Requests and Responses
  + URL: resource locator
    - Allows you to uniquely identify where you want to get your content from
  + Browser makes http request to server to where resource is located
  + Server return http response
  + HTTP: hypertext transfer protocol
    - Simple ubiquitous text-based protocol used by browsers and apps to access web content
  + HTTP Clients
    - Client server protocol
    - Web browsers
    - Mobile apps
    - Programs
  + HTTP Servers
    - Host web pages and web content
    - Static or dynamic(javascript add interactivity to web page)
  + HTTP Requests
    - Clients make http requests
    - GET: fetch resources
    - POST: create/update resources
    - PUT: idempotently create/update
    - HEAD: to get only http header
    - DELETE: to delete resources
  + HTTP Responses
    - Servers are standing by to field requests
    - Send back http responses
      * Status line with code such as 404, 200, etc
      * Response header with metadata
      * Response body
* Web Scraping
  + Automated extraction of data from websites
  + Website content is first fetched(usually using HTTP) and then parsed to extract specific information
  + Web Pages
    - Websites are collections of web pages
    - Web pages consist of markup
    - This markup is understood and rendered by browsers
  + Fetching and Parsing
    - The same HTML markup can be accessed(fetched) via HTTP
    - Possesses an in-built hierarchical structure
    - Parsers can exploit this structure to extract information
* HTTP Client Libraries
  + Web scraping
    - Fetching content through http request through client library
    - Parsing content(HTML parsing, DOM Parsing, Computer Vision)
  + Fetching content
    - Urllib, Urlib2, Requests, Httplib, Httplib2
  + Requests
    - High level api
    - Easy to use
  + Httplib2
    - Fine-grained control of http request
  + Urllib
    - Part of python standard library
    - No need to download
  + Web servers make content available on HTTP endpoints
  + Browsers make HTTP requests under-the-hood to get web pages
  + Web scraping usually involves making such requests programmatically
  + Command-line HTTP requests
    - Curl
  + Python libraries for programmatic access
    - Requests
    - Httplib2
    - Urllib