

# Final Project Write Up

Saturday, October 17, 2015 11:16 AM

## Problem/Idea Definition:

My son plays in a hockey league. The league has a site which displays division group standings and then the record of each team in that division grouping. However, the site layout is cluttered and some of the team information (e.g., schedule) is not well laid out. I want to use the requests library to get the relevant pages. I will then use the beautiful soup library to parse out the interesting content. This content will be stored in a Hockey Team Class object. Using the retrieved information in the object, a nicely formatted web page will be created. This page will have all the relevant information for the division team in question (my son's team). Finally, webbrowser will be used to open the page.

## Solution Design:

### High Level Steps:

- Create a Hockey Team Class to store all team information
  - Team name
  - Team ID (PointStreak identification number)
  - League (LCAHL)
  - League Type (Travel/House)
  - Season (Fall/Winter or Spring and Year)
  - Season ID (PointStreak identification number)
  - Age Group (e.g. Pee Wee A)
    - Note: This varies by location, in Michigan (USA) travel hockey each age is in a specific group whereas in other states multiple ages can be in the same group. e.g., Michigan Pee Wee A = 11 year olds, Michigan Pee Wee AA = 12 year olds, Illinois Pee Wee AA = Best 11/12 year olds, Pee Wee A = Next best 11/12 year olds
  - League Division (e.g. Howe 2)
  - League Division Group (e.g., Blue)
  - League Division Group ID (PointStreak identification number)
  - League Web Site
  - Crossover Division (True/False)
    - League Division Crossover Group (e.g., Yellow - if play crossover games with another group or None if don't)
    - League Division Crossover Group ID
  - Team standing info
    - Games Played
    - Wins
    - Losses
    - Ties
    - Points (Win = 2, Tie = 1, Loss = 0)
    - Goals Fielded (Goals scored against other teams)
    - Goals Allowed (Goals scored against this team)
    - Penalty Minutes
    - Last 5
    - Streak
  - Games Played
    - Game number
    - Home/Away
    - Team Played
    - Date
    - Time
    - Score

- Win/Tie/Loss
- Games Scheduled
  - Game number
  - Home/Away
  - Team Played
  - Date
  - Time
  - Location (Arena and possibly specific rink)
- Retrieve relevant web pages using requests
  - Start with team home page
    - Parse out:
      - Team Home URL (start from here - future enhancement to search for)
      - Season ID
      - Division ID
      - Team Schedule URL
      - Division Standings URL
      - Division Schedule URL
  - Retrieve division group standings page
  - Retrieve team's schedule page
- Enter information or parse each page with beautiful soup to glean it
  - Division group standings page
    - League Name (LCAHL)
    - Travel/House
    - Season/Year
    - Age Group
    - Division
    - Division Group
    - Team Names and IDs
    - For each team
      - Games Played
      - Wins
      - Losses
      - Ties
      - Points
      - Goals Fielded
      - Goal Allowed
      - Penalty Minutes
      - Last 5
      - Streak
  - For each team, look at schedule page
    - Game number
    - Home Team
    - Home Score (if present)
    - Away Team
    - Away Score (if present)
    - Date
    - Time
    - Rink (none if "final")
- Using the parsed information, instantiate a class object for team and populate it with all the information
- Once finished, create a new web page with all the information nicely formatted
  - Main page will show division group standings - similar to current site
  - The bottom of the page will show a list of all played games and the final score of each (for

- each team)
    - A list of all future scheduled games will be also shown for the team
- Once the page is complete it will be launched with webbrowser

Class Design:

```
class HockeyTeam(object):
```

```

-name
-----
-uid
-league
-leaguetype
-season
-sid
-agegroup
-ldivision
-ldivisiongroup
-gid
-team site
-memberName
-team schedpath
-div standpath
-div schedpath
-xover
-div team records
def newseason()
def xoverseason()
def updateteamstanding()
def updateteamschedule()
```

References/Resources Used:

- [Automate the Boring Stuff with Python, Practical Programming for Total Beginners - Chapter 11 – Web Scraping](#)
- [Web Scraping with Python](#), Chapters 1 & 2
- [W3 Schools](#) for HTML and CSS reference
- Requests Library [Documentation](#)
- Beautiful Soup 4 Library [Documentation](#)
- Got some help from [StackOverflow](#)