Perfect Dream11 Team

Generated by Doxygen 1.8.13

Contents

1 Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

2 Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

3 Namespace Documentation

3.1 access Namespace Reference

Classes

- class Read

Functions

- def addH ()

Variables

- list files = [ ]

  List of all the CSV files need to be preprocessed.

3.1.1 Detailed Description

```
@file
```

3.1.2 Function Documentation

3.1.2.1 addH()

```
def access.addH ( )
```

```
It will add header at the first row to the CSVs genrated
Such as: "Player Name" , "Team" , "Run scored" etc..
```

3.2 name Namespace Reference

Classes

- class AllNames
- class FullName

Functions

- def get_initials (s)

  Function that returns the initials of the name passed as argument.
- def get_small (s)

  Function that returns the list of all small characters in a person's name.
- def LCS (s1, s2)

  Function that returns the length of the longest common subsequence (LCS) between two strings.
- def ED (s1, s2)

  Function that returns the edit distance between two strings, i.e.

Variables

- string **allcsv** = "allcsv.txt"
- string **dream11** = "dream11.txt"
- **n** = AllNames(dream11, allcsv)

3.2.1 Detailed Description

```
@file File Documentation
```

3.2.2 Function Documentation

3.2.2.1 ED()

```
def name.ED (
              s1,
              s2 )
```

Function that returns the edit distance between two strings, i.e.

The minimum number of insertions, deletions, and replacements to be done on one string to convert it to the other.

Parameters

s1	First String
s2	Second String

Returns

Edit distance between the two strings.

3.2.2.2 get_initials()

```
def name.get_initials (
              s )
```

Function that returns the initials of the name passed as argument.

Parameters

s	Name passed as string to the function

3.2.2.3 get_small()

```
def name.get_small (
              s )
```

Function that returns the list of all small characters in a person's name.

Parameters

s	Name passed as string to the function

3.2.2.4 LCS()

```
def name.LCS (
              s1,
              s2 )
```

Function that returns the length of the longest common subsequence (LCS) between two strings.

LCS can be used as a heuristic to determine how similar two names are.

Parameters

s1	First string
s2	Second string

Returns

Length of the longest common subsequence.

4 Class Documentation

4.1 name.AllNames Class Reference

Public Member Functions

- def __init__ (self, first, second)

  Constructor that defines the players playing in the match, and the list of all players in the dataset to which these names are to be mapped.
- def __init__ (self, dream11, allcsv)

Constructor that opens files where the list of players playing in a match are stored, and the list of all names contained in the dataset.

- def get_players (self)
- def print_all_players (self)
- def mapped_player (self, player)

  Given a particular player with name given as per Dream11, this function determines what the player's name is most likely to be in the dataset.
- def all_players (self)

Public Attributes

- first

  List of names of players playing in that match.
- second

  List of names of players in the dataset.
- dream11

  File object of the file containing the players playing in this match.
- allcsv

  File object of the file containing the players in the entire dataset.

4.1.1 Detailed Description

```
Class that determines the names of players playing in a match with the name of the corresponding player in the
```

4.1.2 Constructor & Destructor Documentation

4.1.2.1 __init__() [1/2]

```
def name.AllNames.__init__ (
              self,
              first,
              second )
```

Constructor that defines the players playing in the match, and the list of all players in the dataset to which these names are to be mapped.

Parameters

first	List of all names playing in the match.
second	List of all names of players in the dataset.

4.1.2.2 __init__() [2/2]

```
def name.AllNames.__init__ (
              self,
```

```
                dream11,
                allcsv )
```

Constructor that opens files where the list of players playing in a match are stored, and the list of all names contained in the dataset.

Parameters

dream11	Name of file containing names of players playing in the match.
allcsv	Name of file containing names of players contained in the dataset.

4.1.3 Member Function Documentation

4.1.3.1 all_players()

```
def name.AllNames.all_players (
                self )
```

```
Determines the names of all the players playing in the match by mapping it to the dataset names. Uses the func
```

4.1.3.2 get_players()

```
def name.AllNames.get_players (
                self )
```

```
Using the file objects of self, this function populates the list ```self.first``` and ```self.second```.
```

4.1.3.3 mapped_player()

```
def name.AllNames.mapped_player (
                self,
                player )
```

Given a particular player with name given as per Dream11, this function determines what the player's name is most likely to be in the dataset.

Parameters

player	FullName object of a player playing in the match. The name is given as per Dream11 nomenclature.

4.1.3.4 print_all_players()

```
def name.AllNames.print_all_players (
              self )
```

Function simply prints the set of players playing in the match, and the set of players in the dataset. Used in

4.1.4 Member Data Documentation

4.1.4.1 allcsv

```
name.AllNames.allcsv
```

File object of the file containing the players in the entire dataset.

4.1.4.2 dream11

```
name.AllNames.dream11
```

File object of the file containing the players playing in this match.

4.1.4.3 first

```
name.AllNames.first
```

List of names of players playing in that match.

Nomenclature is defined by Dream11 website.

4.1.4.4 second

```
name.AllNames.second
```

List of names of players in the dataset.

This nomenclature is independant of Dream11.

The documentation for this class was generated from the following file:

- name.py

4.2 name.FullName Class Reference

Public Member Functions

- def __init__ (self, name)

  Constructor used to initialize a player's full name.

Public Attributes

- caps
  Contains the list of capital letters (initials) in a person's name.
- small
  Contains the list of small letters in a person's name.
- fullname
  Contains the name of the person.directly.

4.2.1 Detailed Description

```
Class that contains information related to an individuals name. This information can be used to determine what
```

4.2.2 Constructor & Destructor Documentation

4.2.2.1 __init__()

```
def name.FullName.__init__ (
              self,
              name )
```

Constructor used to initialize a player's full name.

Parameters

name	Player's name passed as a simple string

The documentation for this class was generated from the following file:

- name.py

4.3 project.Match Class Reference

Public Member Functions

- def __init__ (self, url, tournament, first, second, time, fimg, simg)
  Constructor which will create the match object.
- def print_match (self)

Public Attributes

- url
  Stores the url of the corresponding match on the Dream11 website.
- tournament
  Stores the url of the corresponding match on the Dream11 website.

- first

  Stores the name of the first team of the cricket match.
- second

  Stores the name of the second team of the cricket match.
- time

  Time left till the match commences.
- fimg

  Url of the logo of the first team.
- simg

  Url of the logo of the second team.
- batsmen

  List of batsmen participating in the match.
- bowlers

  List of bowlers participating in the match.
- wk

  List of wicket keepers in the match.
- ar

  List of all rounders in the match.
- **data**

4.3.1 Detailed Description

```
Match class
This is a match class which will store upcoming match data fetched from the Dream11 website.
```

4.3.2 Constructor & Destructor Documentation

4.3.2.1 __init__()

```
def project.Match.__init__ (
              self,
              url,
              tournament,
              first,
              second,
              time,
              fimg,
              simg )
```

Constructor which will create the match object.

Data for some attributes like batsmen will be fetched later.

Parameters

url	url of perticular match.
tournament	list of tournament.
first	Name of first team.
second	Name of second team.
time	Time remaining to select team.
fimg	Flag image of first team.
simg	Flag image of second team.

4.3.3 Member Function Documentation

4.3.3.1 print_match()

```
def project.Match.print_match (
              self )
```

This method will print the match data of upcoming matches. The printing will be done in html format for ease c

4.3.4 Member Data Documentation

4.3.4.1 ar

```
project.Match.ar
```

List of all rounders in the match.

4.3.4.2 batsmen

```
project.Match.batsmen
```

List of batsmen participating in the match.

4.3.4.3 bowlers

```
project.Match.bowlers
```

List of bowlers participating in the match.

4.3.4.4 fimg

```
project.Match.fimg
```

Url of the logo of the first team.

4.3.4.5 first

`project.Match.first`

Stores the name of the first team of the cricket match.

4.3.4.6 second

`project.Match.second`

Stores the name of the second team of the cricket match.

4.3.4.7 simg

`project.Match.simg`

Url of the logo of the second team.

4.3.4.8 time

`project.Match.time`

Time left till the match commences.

4.3.4.9 tournament

`project.Match.tournament`

Stores the url of the corresponding match on the Dream11 website.

4.3.4.10 url

`project.Match.url`

Stores the url of the corresponding match on the Dream11 website.

4.3.4.11 wk

`project.Match.wk`

List of wicket keepers in the match.

The documentation for this class was generated from the following file:

- project.py

4.4 player.Player Class Reference

Public Member Functions

- def __init__ (self, name, credit)

  Constructor which will create the Player object.
- def print_pl (self)
- def open_match (p, match)

  This method will open the perticular match in web browser and fetch the players information and credits from the html returned by the webdriver.

Public Attributes

- name

  Stores player name of some match.
- credit

  Stores credit given to individual player.

4.4.1 Detailed Description

```
Player class
This is a player class which will store details of players playing in the perticular match.
```

4.4.2 Constructor & Destructor Documentation

4.4.2.1 __init__()

```
def player.Player.__init__ (
              self,
              name,
              credit )
```

Constructor which will create the Player object.

Initialising some variables.

Parameters

name	Name of players.
credit	Individual player credit.

4.4.3 Member Function Documentation

4.4.3.1 open_match()

```
def player.Player.open_match (
              p,
              match )
```

This method will open the perticular match in web browser and fetch the players information and credits from the html returned by the webdriver.

Parameters

p	Object of class Project
match	Object of class project

4.4.3.2 print_pl()

```
def player.Player.print_pl (
              self )
```

```
This method will print the player data of perticular matche. The printing will be done in html format for ease
```

4.4.4 Member Data Documentation

4.4.4.1 credit

```
player.Player.credit
```

Stores credit given to individual player.

4.4.4.2 name

```
player.Player.name
```

Stores player name of some match.

The documentation for this class was generated from the following file:

- player.py

4.5 project.Project Class Reference

Public Member Functions

- def __init__ (self, url)

  Constructor which will create the Project object.
- def make_request (self)
- def parse (self)

Public Attributes

- url

  Stores url that will open in webbrowser.
- matches

  Stores the details of matches that are live now.
- driver

  This will provide connectivity with the browser.

4.5.1 Detailed Description

```
Project Class
This will open the required website in a browser and will parse the webpage to get required data.
```

4.5.2 Constructor & Destructor Documentation

4.5.2.1 __init__()

```
def project.Project.__init__ (
              self,
              url )
```

Constructor which will create the Project object.

Initialising required variable.

Parameters

url	url to open in webbrowser.

4.5.3 Member Function Documentation

4.5.3.1 make_request()

```
def project.Project.make_request (
              self )
```

```
Provide connectivity with the browser using webdriver.
```

4.5.3.2 parse()

```
def project.Project.parse (
              self )
```

This parses the opened webpage and generates the data in a useable form.

4.5.4 Member Data Documentation

4.5.4.1 driver

`project.Project.driver`

This will provide connectivity with the browser.

4.5.4.2 matches

`project.Project.matches`

Stores the details of matches that are live now.

4.5.4.3 url

`project.Project.url`

Stores url that will open in webbrowser.

The documentation for this class was generated from the following file:

- project.py

4.6 access.Read Class Reference

Public Member Functions

- def __init__ (self, filename)

  Constructor to initialize the filename of the CSV file associated with the object Filename of the CSV file of the match.
- def read_file (self)
- def print_file (self)
- def gen_csv (self)
- def calculateScore (self)

Public Attributes

- **filename**
- player_details

  Matrix containing per ball data of the match.
- match_details

  Matrix containing match details of the match.
- data

  Contains data of the match in YAML format, to get extra information of the match wrt the players who have caught/run out batsmen.
- batsmen

  Set of batsmen playing in the match, initialized to empty.
- bowlers

  Set of bowlers playing in the match, initialized to empty.
- fielder

  Set of the fielders in the match that have taken a catch or did run out, initialized to empty.
- teamplayers

  Set of all the players playing in a match, initialized to empty.
- year_list

  List of years from when the T20 match is being played.
- year

  Year of the ongoing match.
- bowler_dict

  Dictionary that contains bowler's detail of a particular match.
- batsmen_dict

  Dictionary that contains batsman's detail of a particular match.
- fielder_dict

  Dictionary that contains fielder's detail of a particular match.

4.6.1 Detailed Description

```
Read Class
Responsible for opening and preprocessing a single file
This class is will open an instance of a single CSV file of a match, preprocess it and create/append to CSV fi
```

4.6.2 Constructor & Destructor Documentation

4.6.2.1 __init__()

```
def access.Read.__init__ (
              self,
              filename )
```

Constructor to initialize the filename of the CSV file associated with the object Filename of the CSV file of the match.

'

Parameters

filename	list off al the files in current directory

4.6.3 Member Function Documentation

4.6.3.1 calculateScore()

```
def access.Read.calculateScore (
              self )
```

Member function to calculate the score of each player according to the Dream 11's Fantasy Cricket Point System

4.6.3.2 gen_csv()

```
def access.Read.gen_csv (
              self )
```

Member Function associated with preprocessing the tables and generating a CSV for each player playing in the m

4.6.3.3 print_file()

```
def access.Read.print_file (
              self )
```

Member function to print details of each match ball by ball.

4.6.3.4 read_file()

```
def access.Read.read_file (
              self )
```

Member function to initialize a few variables related to the match, like team names, venue of the match, etc.
Function also reads the CSV/YAML file and stores it into two tables in the memory, player_details, and match_d

4.6.4 Member Data Documentation

4.6.4.1 data

`access.Read.data`

Contains data of the match in YAML format, to get extra information of the match wrt the players who have caught/run out batsmen.

4.6.4.2 match_details

`access.Read.match_details`

Matrix containing match details of the match.

4.6.4.3 player_details

`access.Read.player_details`

Matrix containing per ball data of the match.

The documentation for this class was generated from the following file:

- access.py

Index