My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Namespace Documentation

3.1 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)

3.1.1 Detailed Description

```
@file File Documentation
```

3.1.2 Function Documentation

3.1.2.1 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.

3.1.2.2 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

3.1.2.3 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

3.1.2.4 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.

Chapter 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 ir
```

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

Index