

# Mediator.py

Links "functions.py" with "website.py".

## Functions:

### getInfoGenes

**function getInfoGenes (): --> dictionary**  
Return a dictionary containing information on geneTable.

### getInfoDiseases

**function getInfoDiseases (): --> dictionary**  
Return a dictionary containing information on diseaseTable.

### getInfo

**function getInfo (): --> tuple containing two dictionaries**  
Returns two dictionaries containing information on the two datasets.

### getDiseaseTableList

**function getDiseaseTableList ( start=0, end=None, step=1, ): --> list**  
Return a list containing the rows of Disease Table from start index to end index. It works like as slicing.

**Parameters:**  
**start : int**  
start index  
  
**end : int**  
end index  
  
**step : int**  
interval index

### getGeneTableList

**function getGeneTableList ( start=0, end=None, step=1, ): --> list**  
Return a list containing the rows of Gene Table from start index to end index. It works like as slicing.

**Parameters:**  
**start : int**  
start index  
  
**end : int**  
end index  
  
**step : int**  
interval index

### getDistinctGenes

**function getDistinctGenes (): --> dict**  
Return a dict with the unique genes present in the gene dataset.

### getDistinctDiseases

**function getDistinctDiseases (): --> dict**  
Return a dict with the unique disease present in the disease dataset.

### getGeneEvidences

**function getGeneEvidences ( gene, ): --> dict**  
Receives as input a geneid or a gene\_symbol and returns a dictionary with the sentences that relates the COVID-19 with the gene.

**Parameters:**  
**gene : str**  
the geneid or gene\_symbol input

### getDiseaseEvidences

**function getDiseaseEvidences ( disease, ): --> dict**  
Receives as input a diseaseID or a disease name and returns a dictionary with the sentences that relates the COVID-19 with the disease.

**Parameters:**  
**disease : str**  
The diseaseID or disease name input

### getCorrelation

**function getCorrelation ( num\_rows, min\_occurrences, ): --> dictionary**  
Returns a dict with the correlations between genes and diseases sorted by the highest number of occurrences. It allows to customize the number of correlations to be shown and the minimum occurrence. When called it calls the method [correlation\\_gene\\_disease](#) of the class [Testing](#) defined in "functions.py"

**Parameters:**  
**num\_rows : int**  
shows only this number of correlations  
  
**min\_occurrences : int**  
show only the correlations with this minimum occurrences

### getDiseasesRelatedToGene

**function getDiseasesRelatedToGene ( gene, ): --> dict**  
Receives as input a geneID or a gene symbol and returns a dict with the diseases related to the gene.

**Parameters:**  
**gene : str**  
geneID or gene\_symbol

### getGenesRelatedToDisease

**function getGenesRelatedToDisease ( disease, ): --> dict**  
Receives as input a diseaseID or a disease name and then returns a dict with the genes related to the disease.

**Parameters:**  
**disease : str**  
diseaseID or disease name

### getDocumentation

**function getDocumentation ( path, name\_file, ): --> dictionary**  
Reads the documentation from .json files and return a dict with the documentation. You can either input the whole path, or the folder and the name of the file. To know more about the format of the documentation read the [section](#) dedicated in the Project Overview

**Parameters:**  
**path : str**  
The path to the file or the path to the folder  
  
**name\_file : str**  
The name of the file. It can be either with extension or without. It's unnecessary if you input the path to the file in 'path'.