PyHal

API Documentation

August 18, 2017

Contents

1 Package hal

1.1 Modules

- **charts** *(Section 2, p. 9)*
  - **bars**: Create easily bar charts
    (Section 3, p. 10)
  - **correlation**: Everything you need to create correlation charts
    (Section 4, p. 12)
  - **plotter**: Show elegant plots in any dimension.
    (Section 5, p. 13)
- **files** *(Section 6, p. 17)*
  - **models**: Main entities in files, such as documents, folders.
    (Section 7, p. 18)
  - **save_as**: Save various data to file
    (Section 8, p. 29)
- **internet** *(Section 9, p. 30)*
  - **engines**: Abstract search engines.
    (Section 10, p. 31)
  - **github**: Common classes and entities in Github
    (Section 11, p. 33)
  - **parser**: Parse anything there is on the Internet.
    (Section 12, p. 38)
  - **selenium_bots**: Some utils methods for a selenium web-driver
    (Section 13, p. 40)
  - **web**: Deal with web-pages.
    (Section 14, p. 42)
  - **youtube**: Get rss feed for youtube channel.
    (Section 15, p. 45)
- **maths**: MATHS: important and scalable math functions
  (Section 16, p. 47)
  - **crypt**: Perform fast hash, encryption and calculations related to cryptography.
    (Section 17, p. 48)
  - **maths**: A few elegant and powerful mathematical functions.
    (Section 18, p. 58)
- **ml** *(Section 19, p. 61)*
  - **analysis** *(Section 20, p. 62)*
    - **correlation**: Correlate values in arrays producing fancy good-looking matrices
      (Section 21, p. 63)
  - **data** *(Section 22, p. 65)*
    - **parser**: Parsers for raw databases
      (Section 23, p. 66)
  - **features**: Collection of methods to find weights of features and select the best ones.
    (Section 24, p. 69)
  - **models** *(Section 25, p. 70)*
    - **classification**: Prediction methods based on classification algorithms.
      (Section 26, p. 71)
    - **pipelined**: Prediction methods based on multiple models mixed up.
      (Section 27, p. 72)
    - **regression**: Prediction methods based on regression algorithms.
      (Section 28, p. 73)

        - **time_series**: Multi-purpose prediction methods to be used in time-series. *(Section 29, p. 74)*
    - **predict**: " General model to make prediction about everything. *(Section 30, p. 76)*
    - **utils** *(Section 31, p. 78)*
        - **matrix**: Functions to deal with matrices. *(Section 32, p. 79)*
        - **misc**: Various tools and utilities to deal with database and machine learning. *(Section 33, p. 82)*
- **mongodb** *(Section 34, p. 83)*
    - **utils**: Various utilities to deal with MondoDB databases *(Section 35, p. 84)*
- **profile** *(Section 36, p. 85)*
    - **mem**: Profile OS memory *(Section 37, p. 86)*
    - **performance**: Perform benchmarks and tests on your PC. *(Section 38, p. 87)*
- **streams** *(Section 39, p. 89)*
    - **pretty_table**: Pretty prints table in SQL style *(Section 40, p. 90)*
- **strings** *(Section 41, p. 92)*
    - **utils**: Typical operations on strings made easy *(Section 42, p. 93)*
- **tests** *(Section 43, p. 94)*
    - **utils**: Tools to ease testing process *(Section 44, p. 95)*
- **time** *(Section 45, p. 96)*
    - **profile**: Tired of formatting ETA times? This is just for you *(Section 46, p. 97)*
    - **utils**: Parse, convert time formats *(Section 47, p. 98)*
- **wrappers** *(Section 48, p. 99)*
    - **methods**: Typical (and useful) function wrappers *(Section 49, p. 100)*

1.2 Variables

Name	Description
__package__	**Value:** `None`

2 Package hal.charts

2.1 Modules

- **bars**: Create easily bar charts
  (Section 3, p. 10)
- **correlation**: Everything you need to create correlation charts
  (Section 4, p. 12)
- **plotter**: Show elegant plots in any dimension.
  (Section 5, p. 13)

2.2 Variables

Name	Description
__package__	**Value:** `None`

3 Module hal.charts.bars

Create easily bar charts

3.1 Functions

create_bar_chart(*title*, *x_labels*, *y_values*, *y_label*)

```
:param title: str
    Title of chart
:param x_labels: [] of str
    Names for each variable
:param y_values: [] of float
    Values of x labels
:param y_label: str
    Label of y axis
:return: Subplot
    Bar chart
```

create_multiple_bar_chart(*title*, *x_labels*, *mul_y_values*, *mul_y_labels*, *normalize*=False)

```
:param title: str
    Title of chart
:param x_labels: [] of str
    Names for each variable
:param mul_y_values: [] of [] of float
    List of values of x labels
:param mul_y_labels: [] of str
    List of labels for each y value
:param normalize: bool
    True iff you want to normalize each y series
:return: Subplot
    Bar chart
```

create_sym_log_bar_chart(*title*, *x_labels*, *y_values*, *y_label*)

```
:param title: str
    Title of chart
:param x_labels: [] of str
    Names for each variable
:param y_values: [] of float
    Values of x labels
:param y_label: str
    Label of y axis
:return: return
    Sym-log bar chart
```

4 Module hal.charts.correlation

```
Everything you need to create correlation charts
```

4.1 Functions

create_correlation_matrix_plot(*correlation_matrix*, *title*, *feature_list*)

```
:param correlation_matrix: [] of []
    Correlation matrix of features
:param title: str
    Title of plot
:param feature_list: [] of str
    List of names of features
:return: void
    shows the given correlation matrix as image
```

5 Module hal.charts.plotter

```
Show elegant plots in any dimension.
```

5.1 Class Plot2d

object

hal.charts.plotter.Plot2d

```
2d plot
```

5.1.1 Methods

scatter(*vector_x*, *vector_y*)

```
:param vector_x: vector in x axis
:param vector_y: vector in y axis
:return: 2d scatter plot
```

param(*self*, *function_x*, *function_y*, *min_val*, *max_val*, *points*)

```
:param function_x: function in x value
:param function_y: function in y value
:param min_val: minimum value
:param max_val: maximum value
:param points: number of points to display
:return: 2d parametric graph of given function from min to max
```

plot(*self*, *function*, *min_val*, *max_val*, *points*)

```
:param function: function to plot
:param min_val: minimum value
:param max_val: maximum value
:param points: number of points
:return: plot 2d function
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

5.1.2 Properties

Name	Description
Inherited from object	
__class__	

5.2 Class Plot3d

object ─┐
hal.charts.plotter.Plot3d

```
3D plot
```

5.2.1 Methods

scatter(*vector_x*, *vector_y*, *vector_z*)

```
:param vector_x: vector in x axis
:param vector_y: vector in y axis
:param vector_z: vector in z axis
:return: plot 3d scattered points
```

param(*self*, *function_x*, *function_y*, *function_z*, *min_val*, *max_val*, *points*)

```
:param function_x: function in x
:param function_y: function in y
:param function_z: function in z
:param min_val: minimum
:param max_val: maximum
:param points: number of points
:return: 3d parametric graph of given function from min to max
```

plot(*self*, *function*, *min_x*, *max_x*, *points_x*, *min_y*, *max_y*, *points_y*)

```
:param function: function to plot
:param min_x: minimum of x-values
:param max_x: maximum of x-values
:param points_x: points in x axis
:param min_y: minimum of y-values
:param max_y: maximum of y-values
:param points_y: points in y axis
:return: plot 3d function
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

5.2.2 Properties

Name	Description
Inherited from object	
__class__	

5.3 Class Plot4d

object ─┐
hal.charts.plotter.Plot4d

```
4D plot generator with slider
```

5.3.1 Methods

scatter(*vector_x*, *vector_y*, *vector_z*, *vector_w*)

```
:param vector_x: vector in x axis
:param vector_y: vector in y axis
:param vector_z: vector in z axis
:param vector_w: vector in w axis
:return: plot 4d scattered points
```

plot(*self*, *function*, *min_x*, *max_x*, *min_y*, *max_y*, *min_z*, *max_z*, *precision*=`0.5`, *kind*=`"contour"`)

```
:param function: function to plot
:param min_x: minimum of x-values
:param max_x: maximum of x-values
:param min_y: minimum of y-values
:param max_y: maximum of y-values
:param min_z: minimum of z-values
:param max_z: maximum of z-values
:param precision: precision
:param kind: slice: x cont -> 3d plot with y, z variables in plane
    and w as "z"-axis contour: x cont -> 3d plot with y,z variables in
    plane and w colored
:return: plot 4d function
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

5.3.2 Properties

Name	Description
Inherited from object __class__	

6 Package hal.files

6.1 Modules

- **models**: Main entities in files, such as documents, folders.
  (Section 7, p. 18)
- **save_as**: Save various data to file
  (Section 8, p. 29)

6.2 Variables

Name	Description
__package__	**Value:** `None`

7 Module hal.files.models

Main entities in files, such as documents, folders.

7.1 Variables

Name	Description
BAD_CHARS	**Value:** `[".", ":", "\"", "\xe2\x80\x99", "&", "720p", "1080p", "y...`
RUSSIAN_CHARS	**Value:** `["\xd1\x88", "\xd0\xb0", "\xd0\xb1", "\xd0\xbb", "\xd0\xb...`
VIDEO_FORMAT	**Value:** `[".", ".3g2", ".3gp", ".amv", ".asf", ".avi", ".drc", ".f...`
ARCHIVE_FORMAT	**Value:** `[".7z", ".??_", ".?Q?", ".?Z?", ".a", ".ace", ".afa", ".a...`
SUBTITLE_FORMAT	**Value:** `[".srt", ".sub", ".sbv"]`
TEXT_FORMAT	**Value:** `[".cnf", ".conf", ".cfg", ".chm", ".epub", ".log", ".asc"...`
IMAGE_FORMAT	**Value:** `[".ani", ".bmp", ".cal", ".fax", ".gif", ".img", ".jbg", ...`
AUDIO_FORMAT	**Value:** `[".3gp", ".aa", ".aac", ".aax", ".act", ".aiff", ".amr", ...`
PATH_SEPARATOR	**Value:** `"/" if "posix" in os.name else "\\"`

7.2 Class FileSystem

object

hal.files.models.FileSystem

Models a folder/file in a OS

7.2.1 Methods

__init__(*self*, *path*)

```
:param path: string
    Path to file
```

Overrides: object.__init__

fix_raw_path(*path*)

```
:param path: string
    Path to fix
:return: string
    Right path
```

remove_year(*name*)

```
:param name: string
    Name to edit
:return: string
    Given string bu with no years.
```

remove_brackets(*name*)

```
:param name: string
    Name to edit
:return: string
    Given string bu with no brackets
```

extract_name_max_chars(*name*, *max_chars*=64, *blank*=" ")

```
:param name: string
    Name to edit
:param max_chars: int
    Maximum chars of new name
:param blank: string
    Char that represents the blank between words.
:return: string
    Name edited to contain at most max_chars (truncate to nearest word)
```

prettify(*name*, *blank*=" ")

```
:param name: string
    Name to edit
:param blank: string
    Default blanks in name.
:return: string
    Prettier name from given one: replace bad chars with good ones.
```

ls_dir(*path*, *include_hidden*=False)

```
:param path: string
    Path to directory to get list of files and folders
:param include_hidden: bool
    Whether to include hidden files in list.
:return: list
    List of paths in given directory.
```

ls_recurse(*path*, *include_hidden*=False)

```
:param path: string
    Path to directory to get list of files and folders
:param include_hidden: bool
    Whether to include hidden files in list.
:return: list
    List of paths in given directory recursively.
```

list_content(*path*, *recurse*, *include_hidden*=False)

```
:param path: string
    Path to directory to get list of files and folders
:param recurse: bool
    Whether to recurse into subdirectories or not.
:param include_hidden: bool
    Whether to include hidden files in list.
:return: list
    List of paths in given directory recursively.
```

is_archive_mac(*self*)

```
:return: True iff document is an MACOSX archive.
```

is_russian(*self*)

```
:return: True iff document has a russian name.
```

trash(*self*)

```
:return: void
    Trash given file/folder
```

rename(*self*, *new_path*)

```
:param new_path: string
    New path to use
:return: void
    Rename to new path
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.2.2 Properties

Name	Description
Inherited from object	
__class__	

7.3 Class Document

object

hal.files.models.FileSystem

hal.files.models.Document

```
File with content in a OS
```

7.3.1 Methods

__init__(*self*, *path*)

```
:param path: string
    Path to file
```

Overrides: object.__init__

move_file_to_directory(*file_path*, *directory_path*)

```
:param file_path: string
    Path to file to move
:param directory_path: string
    Path to target directory where to move file
:return: void
    Move file to given directory
```

move_file_to_file(*old_path*, *new_path*)

```
:param old_path: string
    Old path of file to move
:param new_path: string
    New path (location) of file
:return: void
    Move file from old location to new one
```

write_data_to_file(*data*, *out_file*)

```
:param data: string
    Data to write to file.
:param out_file: string
    Path to output file.
:return: void
    Writes given data to given path file.
```

extract_name_extension(*file_name*)

```
:param file_name: string
    Name of file
:return: tuple string, string
    Name of file, extension of file
```

get_path_name(*self*)

```
:return: tuple string, string
    Name of path, name of file (or folder)
```

is_video(*self*)

```
:return: True iff document is a video.
```

is_subtitle(*self*)

```
:return: True iff document is a subtitle.
```

is_text(*self*)

```
:return: True iff document is a text file.
```

is_image(*self*)

```
:return: True iff document is an image.
```

is_audio(*self*)

```
:return: True iff document is an audio.
```

is_hidden(*self*)

```
:return: bool
    True iff path is hidden
```

Inherited from hal.files.models.FileSystem(Section 7.2)

extract_name_max_chars(), fix_raw_path(), is_archive_mac(), is_russian(), list_content(), ls_dir(), ls_recurse(), prettify(), remove_brackets(), remove_year(), rename(), trash()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.3.2 Properties

Name	Description
Inherited from object __class__	

7.4 Class Directory

object

hal.files.models.FileSystem

hal.files.models.Directory

```
Folder of a OS
```

7.4.1 Methods

__init__(*self*, *path*)

```
:param path: string
    Path to file
```

Overrides: object.__init__

create_new(*path*)

```
:param path: string
    Path to directory to create
:return: void
    Creates new directory
```

get_path_name(*self*)

```
:return: tuple string, string
    Name of path, name of file (or folder)
```

is_empty(*self*)

```
:return: Bool
    True iff empty
```

Inherited from hal.files.models.FileSystem(Section 7.2)

extract_name_max_chars(), fix_raw_path(), is_archive_mac(), is_russian(), list_content(), ls_dir(), ls_recurse(), prettify(), remove_brackets(), remove_year(), rename(), trash()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.4.2 Properties

Name	Description
Inherited from object	
__class__	

7.5 Class MP3Song

object
hal.files.models.FileSystem
hal.files.models.MP3Song

```
mp3 song
```

7.5.1 Methods

__init__(*self*, *path*)

```
:param path: str
    Location of .mp3 file
```

Overrides: object.__init__

set_name(*self*, *name*)

```
:param name: str
    Song’s title
:return: void
    Sets song’s title
```

set_artist(*self*, *artist*)

```
:param artist: str
    Song’s artist
:return: void
    Sets song’s artist
```

set_album(*self*, *album*)

```
:param album: str
    Song’s album
:return: void
    Sets song’s albu
```

set_nr_track(*self*, *nr_track*)

```
:param nr_track: int
    Number of track
:return: void
    Sets song’s track number
```

set_year(*self*, *year*)

```
:param year: int
    Year of song
:return: void
    Sets song’s year
```

Inherited from hal.files.models.FileSystem(Section 7.2)

extract_name_max_chars(), fix_raw_path(), is_archive_mac(), is_russian(), list_content(), ls_dir(), ls_recurse(), prettify(), remove_brackets(), remove_year(), rename(), trash()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.5.2 Properties

Name	Description
Inherited from object __class__	

8 Module hal.files.save_as

```
Save various data to file
```

8.1 Functions

save_dicts_to_csv(*dicts*, *output_file*)

```
:param dicts: [] of {}
    Dictionaries with same values
:param output_file: str
    Path to output file to write data
:return: void
    Saves .csv file with posts data
```

save_matrix_to_csv(*headers*, *data*, *output_file*)

```
:param headers: [] of str
    Column names
:param data: matrix ([] of [])
    Data
:param output_file: str
    Path to output file to write data
:return: void
    Saves .csv file with data
```

8.2 Variables

Name	Description
__package__	**Value:** `'hal.files'`

9 Package hal.internet

9.1 Modules

- **engines**: Abstract search engines.
  (Section 10, p. 31)
- **github**: Common classes and entities in Github
  (Section 11, p. 33)
- **parser**: Parse anything there is on the Internet.
  (Section 12, p. 38)
- **selenium_bots**: Some utils methods for a selenium web-driver
  (Section 13, p. 40)
- **web**: Deal with web-pages.
  (Section 14, p. 42)
- **youtube**: Get rss feed for youtube channel.
  (Section 15, p. 45)

9.2 Variables

Name	Description
__package__	**Value:** `None`

10 Module hal.internet.engines

Abstract search engines.

10.1 Class SearchEngineResult

object ─┐

hal.internet.engines.SearchEngineResult

Result of general search engine

10.1.1 Methods

__init__(*self*, *title*, *link*, *description*="")

`x.__init__(...) initializes x; see help(type(x)) for signature`

Overrides: object.__init__ extit(inherited documentation)

__str__(*self*)

`str(x)`

Overrides: object.__str__ extit(inherited documentation)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __subclasshook__()

10.1.2 Properties

Name	Description
Inherited from object __class__	

10.2 Class SearchEngine

object ─┐

hal.internet.engines.SearchEngine

```
Internet general search engine
```

10.2.1 Methods

__init__(*self*, *url*, *blank_replace*="+")

```
:param url: string
    Url of search engine used in all query.
:param blank_replace:
    Every search engine has to replace blanks in query
```

Overrides: object.__init__

parse_query(*self*, *query*)

```
:param query: string
    Query to search engine.
:return: string
    Parse given query in order to meet search criteria of search engine
```

get_search_page(*self*, *query*, *using_tor*=False)

```
:param query: string
    Query to search engine.
:param using_tor: bool
    Whether use tor or not to fetch web pages
:return: string
    Get HTML source of search page of given query.
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.2.2 Properties

Name	Description
Inherited from object	
__class__	

11 Module hal.internet.github

```
Common classes and entities in Github
```

11.1 Variables

Name	Description
GITHUB_URL_BASE	**Value:** `"https://github.com"`
API_TOKEN_FILE	**Value:** `"api_token"`
API_TOKEN	**Value:** `open(API_TOKEN_FILE).read().strip()`

11.2 Class GithubRawApi

object ─┐
hal.internet.github.GithubRawApi

```
Wrapper for generic Github API
```

11.2.1 Methods

__init__(*self*, *url*=`_API_URL_BASE`, *get_api_content_now*=`False`)

```
:param url: str
    Url of API content to get
:param get_api_content_now: bool
    True iff you want to get API content response when building object
```

Overrides: object.__init__

__getitem__(*self*, *key*)

```
:param key: str
    Dictionary key to find specific user field
:return: str
    Dictionary value of given key
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

11.2.2 Properties

Name	Description
Inherited from object __class__	

11.3 Class GithubApi

object
hal.internet.github.GithubRawApi
hal.internet.github.GithubApi

```
Wrapper for generic Github API
```

11.3.1 Methods

__init__(*self*, *api_type*)

```
:param api_type: str
    Type of API to build
```

Overrides: object.__init__

get_trending_daily()

```
:return: []
    List of GithubUserRepository
```

Inherited from hal.internet.github.GithubRawApi(Section 11.2)

__getitem__()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

11.3.2 Properties

Name	Description
Inherited from object __class__	

11.4 Class GithubUser

```
object ─┐
        │
hal.internet.github.GithubRawApi ─┐
                                  │
        hal.internet.github.GithubApi ─┐
                                       │
                                       hal.internet.github.GithubUser
```

Model of a generic Github user profile

11.4.1 Methods

__init__(*self*, *username*)

```
:param username: str
    Username of user
```

Overrides: object.__init__

get_repos(*self*)

```
:return: [] of GithubUserRepository
    List of user repositories
```

get_starred_repos(*self*)

```
:return: [] of GithubUserRepository
    List of starred repositories
```

get_trending_daily_not_starred(*self*)

```
:return: []
    List of daily-trending repositories which are not starred by user
```

Inherited from hal.internet.github.GithubApi(Section 11.3)

get_trending_daily()

Inherited from hal.internet.github.GithubRawApi(Section 11.2)

__getitem__()

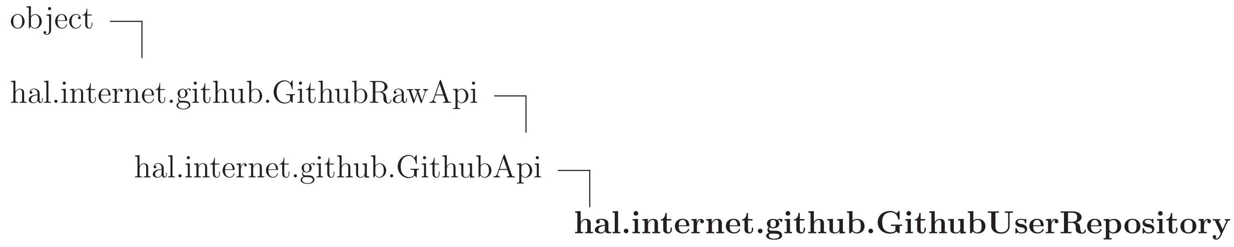
Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

11.4.2 Properties

Name	Description
Inherited from object __class__	

11.5 Class GithubUserRepository


object
hal.internet.github.GithubRawApi
hal.internet.github.GithubApi
hal.internet.github.GithubUserRepository

```
Model of a generic Github user repository
```

11.5.1 Methods

__init__(*self*, *username*, *repository_name*)

```
:param username: str
    Username of user
:param repository_name: str
    Name of repository
```

Overrides: object.__init__

__eq__(*self*, *other*)

Inherited from hal.internet.github.GithubApi(Section 11.3)

get_trending_daily()

Inherited from hal.internet.github.GithubRawApi(Section 11.2)

__getitem__()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

11.5.2 Properties

Name	Description
Inherited from object __class__	

12 Module hal.internet.parser

Parse anything there is on the Internet.

12.1 Functions

is_string_well_formatted(*string*)

```
:param string: string
    String to parse
:return: bool
    True iff string is good formatted
```

html_stripper(*string*)

```
:param string: string
    String to parse
:return: string
    Given string with raw HTML elements removed
```

12.2 Variables

Name	Description
__package__	**Value:** 'hal.internet'

12.3 Class HtmlTable

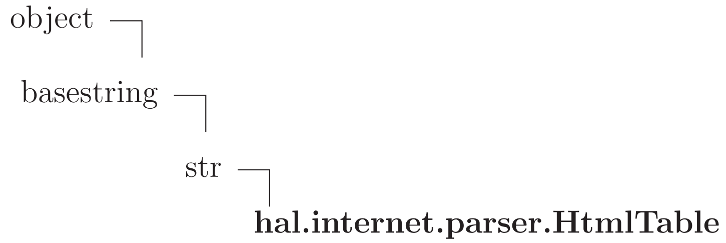


Table written in HTML language

12.3.1 Methods

__init__(*self*, *html_source*)

```
:param html_source: string
    Html source of table
```

Overrides: object.__init__

parse(*self*)

```
:return: list of list
    List of list of values in table
```

Inherited from str

__add__(), __contains__(), __eq__(), __format__(), __ge__(), __getattribute__(), __getitem__(), __getnewargs__(), __getslice__(), __gt__(), __hash__(), __le__(), __len__(), __lt__(), __mod__(), __mul__(), __ne__(), __new__(), __repr__(), __rmod__(), __rmul__(), __sizeof__(), __str__(), capitalize(), center(), count(), decode(), encode(), endswith(), expandtabs(), find(), format(), index(), isalnum(), isalpha(), isdigit(), islower(), isspace(), istitle(), isupper(), join(), ljust(), lower(), lstrip(), partition(), replace(), rfind(), rindex(), rjust(), rpartition(), rsplit(), rstrip(), split(), splitlines(), startswith(), strip(), swapcase(), title(), translate(), upper(), zfill()

Inherited from object

__delattr__(), __reduce__(), __reduce_ex__(), __setattr__(), __subclasshook__()

12.3.2 Properties

Name	Description
Inherited from object	
__class__	

13 Module hal.internet.selenium_bots

Some utils methods for a selenium web-driver

13.1 Variables

Name	Description
__package__	**Value:** None

13.2 Class SeleniumForm

Great and simple static methods to deal with selenium web-drivers.

13.2.1 Methods

fill_form_field(*browser*, *field_name*, *field_value*)

```
:param browser: web-driver
    Browser to use to submit form.
:param field_name :string
    Name of field to fill
:param field_value: string
    Value with which to fill field.
:return: void
    Fill given field with given value.
```

fill_login_form(*browser*, *username*, *username_field*, *user_password*, *user_password_field*)

```
:param browser: web-driver
    Browser to use to submit form
:param username: string
    Username of user to login
:param username_field: string
    Name of field to fill with username
:param user_password: string
    Password of user to login
:param user_password_field: string
    Name of field to fill with user password
:return: void
    Form filled with given information
```

submit_form(*browser*, *button_name*)

```
:param browser: web-driver
    Browser to use to submit form.
:param button_name: string
    Name of button to press to submit form
:return: void
    Submit form.
```

14 Module hal.internet.web

Deal with web-pages.

14.1 Functions

is_url(*candidate_url*)

```
:param candidate_url: str
    Possible url to check for url
:return: bool
    True iff candidate is a valid url
```

download_url(*url*, *local_file*)

```
:param url: string
    Url to download
:param local_file: string
    Save url as this path
:return: void
    Download link to local file
```

download_pdf_to_file(*url*, *local_file*, *chunk_size*=1024)

```
:param url: string
    PDF url to download
:param local_file: string
    Save url as this path
:param chunk_size: int
    Download file in this specific chunk size
:return: void
    Download link to local file
```

14.2 Variables

Name	Description
CHROME_USER_AGENT	**Value:** ["Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWe...

continued on next page

Name	Description
URL_VALID_REGEX	**Value:** `re.compile(r"^(?:http|ftp)s?://" r"(?:"+ r"(?:[A-Z0-9](?:...`

14.3 Class Webpage

object ─┐
hal.internet.web.Webpage

```
representation of URL (web page)
```

14.3.1 Methods

__init__(*self*, *url*, *using_tor*=False)

```
:param url: string
    Url of webpage
:param using_tor: bool
    Whether using tor or not to fetch source page
```

Overrides: object.__init__

parse_url(*raw_url*)

```
:param raw_url: url to parse
:return: parses correctly url
```

get_scheme(*self*)

```
:return: get scheme (HTTP, HTTPS, FTP ..) from given url
```

get_hostname(*self*)

```
:return: extract hostname from given url
```

get_domain(*self*)

```
:return: get domain from given url
```

get_html_source(*self*, *tor*=False)

```
:return: str
    HTML source of webpage
```

get_links(*self*, *recall*, *timeout*)

```
:param recall: max time to attempt to fetch url
:param timeout: max time (s) to wait for web_page response
:return: array of out_links
```

open_in_browser(*self*, *times*)

```
:param times: int
    Times to open webpage in browser
:return: void
    Open a web-driver and go to webpage
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

14.3.2 Properties

Name	Description
Inherited from object	
__class__	

15 Module hal.internet.youtube

```
Get rss feed for youtube channel.
```

15.1 Functions

get_channel_page_from_name(*channel_name*)

```
:param channel_name: string
    name of channel (e.g in  https://www.youtube.com/user/caseyneistat"
    you should take "caseyneistat")
:return string
    source page of youtube channel.
```

get_channel_id_from_name(*channel_name*)

```
:param channel_name: string name of channel (e.g in
    "https://www.youtube.com/user/caseyneistat" you should take
    "caseyneistat")
:return string
    id of youtube channel
```

get_channel_feed_url_from_id(*channel_id*)

```
:param channel_id: string
    id of channel (e.g in
    "https://www.youtube.com/channel/UC2zjki3bJIaXmgV_LBQ2jTg" you should
    take "UC2zjki3bJIaXmgV_LBQ2jTg")
:return string
    rss url feed of youtube channel
```

get_channel_feed_url_from_name(*channel_name*)

```
:param channel_name: string
    name of channel (e.g in "https://www.youtube.com/user/caseyneistat"
    you should take "caseyneistat")
:return string
    rss url feed of youtube channel
```

get_channel_feed_url_from_video(*video_url*)

```
:param video_url: string
    Url of video (e.g in https://www.youtube.com/watch?v=KB_iTbDrkxE)
:return string
    rss url feed of youtube channel.
```

15.2 Variables

Name	Description
YOUTUBE_USER_BASE_URL	**Value:** `"https://www.youtube.com/user/"`
YOUTUBE_FEED_BASE_URL	**Value:** `"https://www.youtube.com/feeds/videos.xml?channel_id="`

16 Package hal.maths

```
MATHS: important and scalable math functions
```

16.1 Modules

- **crypt**: Perform fast hash, encryption and calculations related to cryptography.
  (Section 17, p. 48)
- **maths**: A few elegant and powerful mathematical functions.
  (Section 18, p. 58)

16.2 Variables

Name	Description
__package__	**Value:** `None`

17 Module hal.maths.crypt

Perform fast hash, encryption and calculations related to cryptography.

17.1 Class MD5

object ─┐
hal.maths.crypt.MD5

md5 hash

17.1.1 Methods

__init__(*self*, *string*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

:return: hash plaintext

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.1.2 Properties

Name	Description
Inherited from object __class__	

17.2 Class MD6

object ─┐
hal.maths.crypt.MD6

`md6 hash`

17.2.1 Methods

__init__(*self*, *string*, *size*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

```
:return: return md6 hash
```

hex(*self*, *data*, *size*)

```
:param data: plaintext
:param size: bytes
:return: hex representation
```

raw(*self*, *data*, *size*)

```
:param data: plaintext
:param size: bytes
:return: raw representation
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.2.2 Properties

Name	Description
Inherited from object __class__	

17.2.3 Class Variables

Name	Description
ALLOWED_SIZE	**Value:** `[64, 128, 224, 256, 384, 512]`

17.3 Class SHA

object ─┐
hal.maths.crypt.SHA

`general SHA hash`

17.3.1 Methods

__init__(*self*, *string*, *size*=`ALLOWED_SIZE [0]`, *salt*=`None`)

`x.__init__(...) initializes x; see help(type(x)) for signature`

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

`:return: hash of given size`

hash_sha1(*self*)

`:return: sha1 hash`

hash_sha224(*self*)

`:return: sha224 hash`

hash_sha256(*self*)

`:return: sha256 hash`

hash_sha384(*self*)

```
:return: sha384 hash
```

hash_sha512(*self*)

```
:return: sha512 hash
```

hash_sha_salted(*self*)

```
:return: sha512 hash
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.3.2 Properties

Name	Description
Inherited from object __class__	

17.3.3 Class Variables

Name	Description
ALLOWED_SIZE	**Value:** `[1, 224, 256, 384, 512]`

17.4 Class ARC

object
hal.maths.crypt.ARC

```
ARC hash
```

17.4.1 Methods

__init__(*self*, *string*, *key*, *size*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

```
:return: hash of given size
```

hash_ar2(*self*)

```
:return: des hash
```

hash_arc4(*self*)

```
:return: des3 hash
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.4.2 Properties

Name	Description
Inherited from object __class__	

17.4.3 Class Variables

Name	Description
ALLOWED_SIZE	**Value:** `[2, 4]`

17.5 Class HMAC

object ─┐
hal.maths.crypt.HMAC

```
hmac hash
```

17.5.1 Methods

__init__(*self*, *string*, *key*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

```
:return: hash plaintext
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.5.2 Properties

Name	Description
Inherited from object __class__	

17.6 Class BLOWFISH

object ─┐
hal.maths.crypt.BLOWFISH

```
blowfish hash
```

17.6.1 Methods

__init__(*self*, *string*, *key*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

```
:return: hash plaintext
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.6.2 Properties

Name	Description
Inherited from object	
__class__	

17.7 Class IDEA

object

hal.maths.crypt.IDEA

```
IDEA hash
```

17.7.1 Methods

__init__(*self*, *string*, *key*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

```
:return: IDEA hash
```

change_key(*self*, *key*)

```
:param key: new key
:return: change key
```

encrypt(*self*)

```
:return: encrypt with key
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.7.2 Properties

Name	Description
Inherited from object	
__class__	

17.8 Class CAST128

object

hal.maths.crypt.CAST128

```
CAST 128 hash
```

17.8.1 Methods

__init__(*self*, *string*, *key*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

encrypt(*self*)

```
:return: str
    Encrypt
```

decrypt(*self*)

```
:return: str
    Decrypt
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.8.2 Properties

Name	Description
Inherited from object __class__	

17.9 Class Dsa

object ─┐

hal.maths.crypt.Dsa

```
dsa hash
```

17.9.1 Methods

__init__(*self*, *string*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

hash(*self*)

```
:return: hash plaintext
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.9.2 Properties

Name	Description
Inherited from object	
__class__	

18 Module hal.maths.maths

A few elegant and powerful mathematical functions.

18.1 Functions

get_prime(*bits*)

```
:param bits: size of number to generate (bits)
:return: prime number of given size
```

blum_blum_shub(*seed*, *amount*, *prime0*, *prime1*)

```
:param seed: seeder
:param amount: amount of number to generate
:param prime0: one prime number
:param prime1: the second prime number
:return: pseudo-number generator
```

18.2 Variables

Name	Description
LOW_PRIMES	**Value:** `[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, ...`
__package__	**Value:** `'hal.maths'`

18.3 Class Integer

object ─┐
hal.maths.maths.Integer

Big int std python won't recognize

18.3.1 Methods

__init__(*self*, *string*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

is_naive_prime(*self*)

```
:return: bool
    Checks if prime in very naive way
```

is_probably_prime(*self*)

```
:return: test with miller-rabin
```

test_miller_rabin(*self*, *precision*)

```
:param precision: number of rounds to perform (higher -> better
precision)
:return: True iff probably prime
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

18.3.2 Properties

Name	Description
Inherited from object	
__class__	

18.4 Class EightQueen

object ─┐

hal.maths.maths.EightQueen

```
8 queen problem solver
```

18.4.1 Methods

__init__(*self*, *board_size*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

under_attack(*col*, *queens*)

```
:param col: int
    Column number
:param queens: []
    List of queens
:return: bool
    True iff queen is under attack
```

solve(*self*, *table_size*)

```
:param table_size: int
    Size of table
:return: []
    List of possible solutions
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

18.4.2 Properties

Name	Description
Inherited from object __class__	

19 Package hal.ml

19.1 Modules

- **analysis** *(Section 20, p. 62)*
  - **correlation**: Correlate values in arrays producing fancy good-looking matrices *(Section 21, p. 63)*
- **data** *(Section 22, p. 65)*
  - **parser**: Parsers for raw databases *(Section 23, p. 66)*
- **features**: Collection of methods to find weights of features and select the best ones. *(Section 24, p. 69)*
- **models** *(Section 25, p. 70)*
  - **classification**: Prediction methods based on classification algorithms. *(Section 26, p. 71)*
  - **pipelined**: Prediction methods based on multiple models mixed up. *(Section 27, p. 72)*
  - **regression**: Prediction methods based on regression algorithms. *(Section 28, p. 73)*
  - **time_series**: Multi-purpose prediction methods to be used in time-series. *(Section 29, p. 74)*
- **predict**: " General model to make prediction about everything. *(Section 30, p. 76)*
- **utils** *(Section 31, p. 78)*
  - **matrix**: Functions to deal with matrices. *(Section 32, p. 79)*
  - **misc**: Various tools and utilities to deal with database and machine learning. *(Section 33, p. 82)*

19.2 Variables

Name	Description
__package__	**Value:** `None`

20 Package hal.ml.analysis

20.1 Modules

- **correlation**: Correlate values in arrays producing fancy good-looking matrices *(Section 21, p. 63)*

20.2 Variables

Name	Description
__package__	**Value:** `None`

21 Module hal.ml.analysis.correlation

```
Correlate values in arrays producing fancy good-looking matrices
```

21.1 Functions

get_correlation_matrix(*matrix*)

```
:param matrix: [] of []
    List of features to get correlation matrix
:return: [] of []
    correlation matrix
```

show_correlation_matrix(*correlation_matrix*, *title*, *feature_list*)

```
:param correlation_matrix: [] of []
    Correlation matrix of features
:param title: str
    Title of plot
:param feature_list: [] of str
    List of names of features
:return: void
    shows the given correlation matrix as image
```

get_correlation_matrix_of_columns(*headers_to_test*, *headers*, *data*)

```
:param headers_to_test: [] of str
    List of columns to get correlation matrix of
:param headers: [] of str
    List of all headers in matrix
:param data: [] of []
    Matrix of float values
:return: [] of []
    Correlation matrix of selected columns
```

show_correlation_matrix_of_columns(*title*, *headers_to_test*, *headers*, *data*)

```
:param title: str
    Title to show
:param headers_to_test: [] of str
    List of columns to get correlation matrix of
:param headers: [] of str
    List of all headers in matrix
:param data: [] of []
    Matrix of float values
:return: void
    Shows on screen correlation matrix of selected headers
```

save_correlation_matrix_of_columns(*title*, *headers_to_test*, *headers*, *data*, *out_file*)

```
:param title: str
    Title to show
:param headers_to_test: [] of str
    List of columns to get correlation matrix of
:param headers: [] of str
    List of all headers in matrix
:param data: [] of []
    Matrix of float values
:param out_file: str
    Output file
:return: void
    Saves correlation matrix of selected headers
```

save_correlation_matrix_of_data_files_in_folder(*folder_path*)

```
:param folder_path: str
    Folder containing logs data
:return: void
    Saves each file's correlation matrix of common headers
```

22 Package hal.ml.data

22.1 Modules

- **parser**: Parsers for raw databases
  (Section 23, p. 66)

22.2 Variables

Name	Description
__package__	**Value:** `None`

23 Module hal.ml.data.parser

`Parsers for raw databases`

23.1 Functions

parse_csv_file(*file_path*)

```
:param file_path: str
   Path to file to parse
:return: tuple [], [] of []
   headers of csv file and data
```

23.2 Variables

Name	Description
__package__	**Value:** `'hal.ml.data'`

23.3 Class Parser

object

hal.ml.data.parser.Parser

Known Subclasses: hal.ml.data.parser.CSVParser

`Mother of all data-files parsers`

23.3.1 Methods

__init__(*self*, *database_file*)

```
:param database_file: a raw .csv file that contains any data
about anything
```

Overrides: object.__init__

get_lines(*self*)

```
:return: [] of str
    Lines in file
```

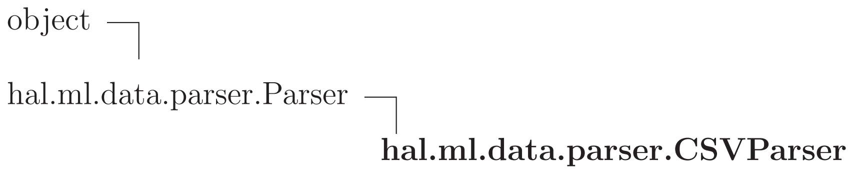
Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

23.3.2 Properties

Name	Description
Inherited from object	
__class__	

23.4 Class CSVParser



object

hal.ml.data.parser.Parser

hal.ml.data.parser.CSVParser

```
Parses CSV data files
```

23.4.1 Methods

__init__(*self*, *database_file*)

```
:param database_file: a raw .csv file that contains any data
about anything
```

Overrides: object.__init__

parse_data(*self*)

```
store values in array, store lines in array; the result is a 2D
matrix
```

Inherited from hal.ml.data.parser.Parser(Section 23.3)

get_lines()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

23.4.2 Properties

Name	Description
Inherited from object __class__	

24 Module hal.ml.features

Collection of methods to find weights of features and select the best ones.

24.1 Functions

select_k_best(*x_data*, *y_data*, *num_features*)

select k best features in dataset

get_best_features(*x_data*, *y_data*)

finds the optimal number of features

get_features(*x_data*, *y_data*, *num_features*)

finds the optimal features

25 Package hal.ml.models

25.1 Modules

- **classification**: Prediction methods based on classification algorithms.
  (Section 26, p. 71)
- **pipelined**: Prediction methods based on multiple models mixed up.
  (Section 27, p. 72)
- **regression**: Prediction methods based on regression algorithms.
  (Section 28, p. 73)
- **time_series**: Multi-purpose prediction methods to be used in time-series.
  (Section 29, p. 74)

25.2 Variables

Name	Description
__package__	**Value:** `None`

26 Module hal.ml.models.classification

```
Prediction methods based on classification algorithms.
```

26.1 Functions

extra_trees_classifier()

```
:return: sklearn ExtraTreesClassifier
    Classical extra tree classifier
```

random_forest()

```
:return: sklearn RandomForestClassifier
    Classical random forest classifier
```

knn()

```
:return: sklearn KNN
    Classical knn
```

ada_boost()

```
:return: sklearn AdaBoostClassifier
    Classical Ada boost
```

bayes_gauss()

```
:return: sklearn GaussianNB
    Slower than svr but equally accurate
```

bayes_bernoulli()

```
:return: sklearn BernoulliNB
    Bayes-Bernoulli model
```

27 Module hal.ml.models.pipelined

Prediction methods based on multiple models mixed up.

27.1 Functions

logistic_rbm()

```
:return: rbm -> logistic
```

anova_svm()

```
:return: anova -> svc
```

28 Module hal.ml.models.regression

Prediction methods based on regression algorithms.

28.1 Functions

support_vector_machine()

```
:return: sklearn svm.SVR
    Classical polynomial SVM
```

logistic_regression()

```
:return: sklearn LogisticRegression
    Logistic regression model
```

29 Module hal.ml.models.time_series

```
Multi-purpose prediction methods to be used in time-series.
```

29.1 Functions

test_stationary(*time_series*)

```
:param time_series: []
:return: void
    Shows plot and checks for stationary series
```

get_str_end(*dates*, *end*)

```
:param dates: []
    List of str date
:param end: float
    End of prediction
:return: str
    End of prediction
```

arma(*dates*, *values*, *start*=`None`, *end*=`None`, *plot*=`False`)

```
Predict days values using ARMA algorithm.
:param dates: list of str date
:param values: list of float values
:param start: start predicting in this day
:param end: end of prediction
:param plot: whether to plot or not values in graph
```

arima(*dates*, *values*, *start*=`None`, *end*=`None`)

```
Predict days values using ARIMA algorithm.
:param dates: list of str date
:param values: list of float values
:param start: start predicting in this day
:param end: end of prediction
```

var(*dates*, *values*)

```
Predict days values using ARIMA algorithm.
:param dates: list of str date
:param values: list of float values
```

dynamic_var(*dates*, *values*)

```
Predict days values using ARIMA algorithm.
:param dates: list of str date
:param values: list of float values
```

30 Module hal.ml.predict

```
" General model to make prediction about everything.
```

30.1 Class BasePrediction

object

hal.ml.predict.BasePrediction

```
The mother of all predictions
```

30.1.1 Methods

__init__(*self*, *model*, *rounds*)

```
:param model: sklearn.model
    Model chosen for prediction
:param rounds: int
    Number of rounds to repeat prediction (and get better results)
```

Overrides: object.__init__

train(*self*, *x_data*, *y_data*)

```
:param x_data: data
    Input x
:param y_data: data
    Input y
:return: void
    Train model on inputs
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.1.2 Properties

continued on next page

Name	**Description**

Name	**Description**
Inherited from object __class__	

31 Package hal.ml.utils

31.1 Modules

- **matrix**: Functions to deal with matrices.
  (Section 32, p. 79)
- **misc**: Various tools and utilities to deal with database and machine learning.
  (Section 33, p. 82)

31.2 Variables

Name	Description
__package__	**Value:** `None`

32 Module hal.ml.utils.matrix

Functions to deal with matrices.

32.1 Functions

precision(*matrix*)

```
:param matrix: 2 x 2 matrix
    True positives are in [0,0], true negatives are in [1,1],
    false negatives are in [0,1] and false positives are in [1,0]
:return: float
    Calculates accuracy on database
```

recall(*matrix*)

```
:param matrix: 2 x 2 matrix
    True positives are in [0,0], true negatives are in [1,1],
    false negatives are in [0,1] and false positives are in [1,0]
:return: float
    Calculates recall on database
```

true_neg_rate(*matrix*)

```
:param matrix: 2 x 2 matrix
    True positives are in [0,0], true negatives are in [1,1],
    false negatives are in [0,1] and false positives are in [1,0]
:return: float
    Calculates true negative rate on database
```

accuracy(*matrix*)

```
:param matrix: 2 x 2 matrix
    True positives are in [0,0], true negatives are in [1,1],
    false negatives are in [0,1] and false positives are in [1,0]
:return: float
    Calculates accuracy on database
```

f1_score(*matrix*)

```
:param matrix: 2 x 2 matrix
    True positives are in [0,0], true negatives are in [1,1],
    false negatives are in [0,1] and false positives are in [1,0]
:return: float
    Calculates F1 score on database
```

get_column_of_matrix(*column_index*, *matrix*)

```
:param column_index: int >= 0
    Column index to take
:param matrix: [] of []
    Matrix
:return: []
    Column of array at position given
```

get_subset_of_matrix(*headers_to_sample*, *all_headers*, *data*)

```
:param headers_to_sample: [] of str
    List of columns to get
:param all_headers: [] of str
    List of all headers in matrix
:param data: [] of []
    Matrix of float values
:return: [] of []
    Correlation matrix of selected columns
```

remove_column_from_matrix(*headers*, *header_to_remove*, *data*)

```
:param headers: [] of str
    Column names
:param header_to_remove: str
    Name of column to remove
:param data: matrix ([] of [])
    Data
:return: headers, data
    Headers without header removed and data without column removed
```

add_columns_to_matrix(*headers*, *data*, *new_headers*, *new_columns*)

```
:param headers: headers: [] of str
    Column names
:param data: matrix ([] of [])
    Data
:param new_headers: [] of str
    Names of new columns
:param new_columns: ([] of [])
    New columns to add
:return: headers, data
    New headers (with new headers) and data with new columns
```

33 Module hal.ml.utils.misc

```
Various tools and utilities to deal with database and machine learning.
```

33.1 Functions

pearson(*lst1*, *lst2*)

```
Pearson coefficient of arrays
```

normalize_array(*arr*)

```


:param arr: [] of float
    Array of floats
:return: [] of float
    Normalized (in [0, 1]) input array
```

34 Package hal.mongodb

34.1 Modules

- **utils**: Various utilities to deal with MondoDB databases
(Section 35, p. 84)

34.2 Variables

Name	Description
__package__	**Value:** `None`

35 Module hal.mongodb.utils

```
Various utilities to deal with MondoDB databases
```

35.1 Functions

get_documents_count(*db_name*)

```
:param db_name: str
    Name of db
:return: int
    Number of documents in db
```

get_documents_in_collection(*db_name*, *collection_name*, *with_id*=`True`)

```
:param db_name: str
    Name of db
:param collection_name: str
    Name of collection
:param with_id: bool
    True iff each document should also come with its id
:return: [] of {}
    List of documents in collection in database
```

get_documents_in_database(*db_name*, *with_id*=`True`)

```
:param db_name: str
    Name of db
:param with_id: bool
    True iff each document should also come with its id
:return: [] of {}
    List of documents in collection in database
```

36 Package hal.profile

36.1 Modules

- **mem**: Profile OS memory
  (Section 37, p. 86)
- **performance**: Perform benchmarks and tests on your PC.
  (Section 38, p. 87)

36.2 Variables

Name	Description
__package__	**Value:** `None`

37 Module hal.profile.mem

```
Profile OS memory
```

37.1 Functions

get_memory_usage()

```
:return: float
    MB of memory used by this process
```

force_garbage_collect()

```
:return: void
    Releases memory used
```

38 Module hal.profile.performance

```
Perform benchmarks and tests on your PC.
```

38.1 Class EightQueenTest

```
object ─┐
        │
    hal.profile.performance.EightQueenTest
```

```
Test CPU by solving eight-queen problem
```

38.1.1 Methods

__init__(*self*, *size*)

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

Overrides: object.__init__ extit(inherited documentation)

welcome()

```
:return: string
    Welcomes user to this test sessions
```

introduction()

```
:return: string
Welcomes user to this test sessions
```

run_test_with_size(*size*)

```
:param size: int
    Number of rows in grid
:return: int
    Time to solve problem with given size
```

update_std_out_and_log(*self*, *string*)

```
:param string: string
    Stuff to print
:return: void
    Prints to stdout and updates log
```

start(*self*)

```
:return: void
    Starts profiling
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

38.1.2 Properties

Name	Description
Inherited from object	
__class__	

39 Package hal.streams

39.1 Modules

- **pretty_table**: Pretty prints table in SQL style
  (Section 40, p. 90)

39.2 Variables

Name	Description
__package__	**Value:** `None`

40 Module hal.streams.pretty_table

```
Pretty prints table in SQL style
```

40.1 Functions

get_optimal_column_widths(*labels*, *data*)

```
:param labels: [] of str
    List of labels of data
:param data: ([] of []) of anything
    Matrix of any type
:return: [] of int
    Length of longest data in each column (labels and data)
```

get_pretty_row(*data*, *widths*, *filler*, *splitter*)

```
:param data: [] of anything
    List of data
:param widths: [] of int
    Length of longest data in each column
:param filler: char
    Fill empty columns with this char
:param splitter: char
    Separate columns with this char
:return: str
    Pretty formatted row
```

get_blank_row(*widths*, *filler*=`'-'`, *splitter*=`'+'`)

```
:param widths: [] of int
    Length of longest data in each column
:param filler: char
    Fill empty columns with this char
:param splitter: char
    Separate columns with this char
:return: str
    Pretty formatted blank row (with no meaningful data in it)
```

pretty_format_row(*data*, *widths*, *filler*=' ', *splitter*='|')

```
:param data: [] of anything
    List of data
:param widths: [] of int
    Length of longest data in each column
:param filler: char
    Fill empty columns with this char
:param splitter: char
    Separate columns with this char
:return: str
    Pretty formatted row
```

pretty_format_table(*labels*, *data*, *line_separator*='\n')

```
:param labels: [] of str
    List of labels of data
:param data: ([] of []) of anything
    Matrix of any type
:param line_separator: str
    Separate each new line with this
:return: str
    Pretty formatted table (first row is labels, then actual data)
```

40.2 Variables

Name	Description
__package__	**Value:** `None`

41 Package hal.strings

41.1 Modules

- **utils**: Typical operations on strings made easy
  (Section 42, p. 93)

41.2 Variables

Name	Description
__package__	**Value:** `None`

42 Module hal.strings.utils

```
Typical operations on strings made easy
```

42.1 Functions

how_similar_are(*str1*, *str2*)

```
:param str1: str
    First string
:param str2: str
    Second string
:return: float in [0, 1]
    Similarity of a VS b
```

get_average_length_of_word(*words*)

```
:param words: [] of str
    Words
:return: float
    Average length of word on list
```

42.2 Variables

Name	Description
__package__	**Value:** `'hal.strings'`

43 Package hal.tests

43.1 Modules

- **utils**: Tools to ease testing process
  (Section 44, p. 95)

43.2 Variables

Name	Description
__package__	**Value:** `None`

44 Module hal.tests.utils

Tools to ease testing process

44.1 Functions

battery_test(*assert_type*, *tests*, *function*, *args*=None)

```
:param assert_type: function
    Type of assert
:param tests: dict
    key= params in function, value= what should be the result
:param function: function
    Function to apply
:param args: *
    Extra args for function to call
:return: bool
    True iff all tests pass
```

random_name()

```
:return: str
    Pseudo-random name
```

44.2 Variables

Name	Description
__package__	**Value:** 'hal.tests'

45 Package hal.time

45.1 Modules

- **profile**: Tired of formatting ETA times? This is just for you
  (Section 46, p. 97)
- **utils**: Parse, convert time formats
  (Section 47, p. 98)

45.2 Variables

Name	Description
__package__	**Value:** `None`

46 Module hal.time.profile

Tired of formatting ETA times? This is just for you

46.1 Functions

get_time_eta(*total_done*, *total*, *start_time*)

```
:param total_done: int
    Item processed
:param total: int
    Total number of items to process
:param start_time: time
    Time of start processing items
:return: time
    Time to go
```

print_item_info(*details*)

```
:param details: {}
    Details of AthletePerformance
:return: void
    Prints debug info to screen
```

print_time_eta(*time_to_go*, *note=''*)

```
:param time_to_go: {}
    Result of a call get_time_eta(...)
:param note: str
    Notes to append to stdout
:return: void
    Prints debug info to screen
```

46.2 Variables

Name	Description
__package__	**Value:** `'hal.time'`

47 Module hal.time.utils

```
Parse, convert time formats
```

47.1 Functions

parse_hh_mm_ss(*string*)

```
:param string: str
    Hours, minutes and seconds in the form hh:mm:ss to parse
:return: datetime.time
    Time parsed
```

get_seconds(*string*)

```
:param string: str
    Datetime in the form %H:%M:%S
:return: int
    Seconds in time
```

parse_hh_mm(*string*)

```
:param string: str
    Hours and minutes in the form hh:mm to parse
:return: datetime.time
    Time parsed
```

47.2 Variables

Name	Description
MONTHS_NAMES	**Value:** `[datetime.strftime(datetime(year= 1, month= m, day= 1), "...`
MONTHS	**Value:** `{i+ 1: MONTHS_NAMES [i] for i in range(len(MONTHS_NAMES))}`

48 Package hal.wrappers

48.1 Modules

- **methods**: Typical (and useful) function wrappers
  (Section 49, p. 100)

48.2 Variables

Name	Description
__package__	**Value:** `None`

49 Module hal.wrappers.methods

```
Typical (and useful) function wrappers
```

49.1 Functions

handle_exceptions(*function*)

```
:param function: callback function
    function to wrap
:return: callback function return type
    wraps callback function
```

Index