

# PyHal

## API Documentation

August 15, 2017

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Package hal</b>	<b>7</b>
1.1 Modules . . . . .	7
1.2 Variables . . . . .	8
<b>2 Package hal.charts</b>	<b>9</b>
2.1 Modules . . . . .	9
2.2 Variables . . . . .	9
<b>3 Module hal.charts.bar</b>	<b>10</b>
3.1 Functions . . . . .	10
<b>4 Module hal.charts.correlation</b>	<b>11</b>
4.1 Functions . . . . .	11
<b>5 Module hal.charts.plotter</b>	<b>12</b>
5.1 Class Plot2d . . . . .	12
5.1.1 Methods . . . . .	12
5.1.2 Properties . . . . .	12
5.2 Class Plot3d . . . . .	13
5.2.1 Methods . . . . .	13
5.2.2 Properties . . . . .	14
5.3 Class Plot4d . . . . .	14
5.3.1 Methods . . . . .	14
5.3.2 Properties . . . . .	15
<b>6 Package hal.files</b>	<b>16</b>
6.1 Modules . . . . .	16
6.2 Variables . . . . .	16
<b>7 Module hal.files.models</b>	<b>17</b>
7.1 Variables . . . . .	17
7.2 Class FileSystem . . . . .	17
7.2.1 Methods . . . . .	17
7.2.2 Properties . . . . .	20
7.3 Class Document . . . . .	21
7.3.1 Methods . . . . .	21

7.3.2	Properties . . . . .	23
7.4	Class Directory . . . . .	23
7.4.1	Methods . . . . .	24
7.4.2	Properties . . . . .	24
7.5	Class MP3Song . . . . .	25
7.5.1	Methods . . . . .	25
7.5.2	Properties . . . . .	26
<b>8</b>	<b>Module hal.files.save_as</b>	<b>27</b>
8.1	Functions . . . . .	27
8.2	Variables . . . . .	27
<b>9</b>	<b>Package hal.internet</b>	<b>28</b>
9.1	Modules . . . . .	28
9.2	Variables . . . . .	28
<b>10</b>	<b>Module hal.internet.engines</b>	<b>29</b>
10.1	Class SearchEngineResult . . . . .	29
10.1.1	Methods . . . . .	29
10.1.2	Properties . . . . .	29
10.2	Class SearchEngine . . . . .	29
10.2.1	Methods . . . . .	30
10.2.2	Properties . . . . .	30
<b>11</b>	<b>Module hal.internet.github</b>	<b>31</b>
11.1	Variables . . . . .	31
11.2	Class GithubRawApi . . . . .	31
11.2.1	Methods . . . . .	31
11.2.2	Properties . . . . .	32
11.3	Class GithubApi . . . . .	32
11.3.1	Methods . . . . .	32
11.3.2	Properties . . . . .	33
11.4	Class GithubUser . . . . .	33
11.4.1	Methods . . . . .	33
11.4.2	Properties . . . . .	34
11.5	Class GithubUserRepository . . . . .	34
11.5.1	Methods . . . . .	34
11.5.2	Properties . . . . .	35
<b>12</b>	<b>Module hal.internet.parser</b>	<b>36</b>
12.1	Functions . . . . .	36
12.2	Variables . . . . .	36
12.3	Class HtmlTable . . . . .	36
12.3.1	Methods . . . . .	37
12.3.2	Properties . . . . .	37
<b>13</b>	<b>Module hal.internet.selenium</b>	<b>38</b>
13.1	Variables . . . . .	38
13.2	Class SeleniumForm . . . . .	38
13.2.1	Methods . . . . .	38
<b>14</b>	<b>Module hal.internet.web</b>	<b>40</b>
14.1	Functions . . . . .	40

14.2 Variables . . . . .	40
14.3 Class Webpage . . . . .	41
14.3.1 Methods . . . . .	41
14.3.2 Properties . . . . .	42
<b>15 Module hal.internet.youtube</b>	<b>43</b>
15.1 Functions . . . . .	43
15.2 Variables . . . . .	44
<b>16 Package hal.maths</b>	<b>45</b>
16.1 Modules . . . . .	45
16.2 Variables . . . . .	45
<b>17 Module hal.maths.crypt</b>	<b>46</b>
17.1 Class MD5 . . . . .	46
17.1.1 Methods . . . . .	46
17.1.2 Properties . . . . .	46
17.2 Class MD6 . . . . .	46
17.2.1 Methods . . . . .	47
17.2.2 Properties . . . . .	47
17.2.3 Class Variables . . . . .	47
17.3 Class SHA . . . . .	48
17.3.1 Methods . . . . .	48
17.3.2 Properties . . . . .	49
17.3.3 Class Variables . . . . .	49
17.4 Class DES . . . . .	49
17.4.1 Methods . . . . .	50
17.4.2 Properties . . . . .	50
17.4.3 Class Variables . . . . .	50
17.5 Class ARC . . . . .	51
17.5.1 Methods . . . . .	51
17.5.2 Properties . . . . .	51
17.5.3 Class Variables . . . . .	51
17.6 Class AES . . . . .	52
17.6.1 Methods . . . . .	52
17.6.2 Properties . . . . .	52
17.7 Class HMAC . . . . .	52
17.7.1 Methods . . . . .	53
17.7.2 Properties . . . . .	53
17.8 Class BLOWFISH . . . . .	53
17.8.1 Methods . . . . .	53
17.8.2 Properties . . . . .	54
17.9 Class IDEA . . . . .	54
17.9.1 Methods . . . . .	54
17.9.2 Properties . . . . .	55
17.10 Class CAST128 . . . . .	55
17.10.1 Methods . . . . .	55
17.10.2 Properties . . . . .	55
17.11 Class Dsa . . . . .	56
17.11.1 Methods . . . . .	56
17.11.2 Properties . . . . .	56

<b>18 Module hal.maths.maths</b>	<b>57</b>
18.1 Functions . . . . .	57
18.2 Variables . . . . .	57
18.3 Class Integer . . . . .	57
18.3.1 Methods . . . . .	57
18.3.2 Properties . . . . .	58
18.3.3 Class Variables . . . . .	58
18.4 Class EightQueen . . . . .	58
18.4.1 Methods . . . . .	59
18.4.2 Properties . . . . .	59
<b>19 Package hal.ml</b>	<b>60</b>
19.1 Modules . . . . .	60
19.2 Variables . . . . .	60
<b>20 Package hal.ml.analysis</b>	<b>61</b>
20.1 Modules . . . . .	61
20.2 Variables . . . . .	61
<b>21 Module hal.ml.analysis.correlation</b>	<b>62</b>
21.1 Functions . . . . .	62
<b>22 Package hal.ml.data</b>	<b>64</b>
22.1 Modules . . . . .	64
22.2 Variables . . . . .	64
<b>23 Module hal.ml.data.parser</b>	<b>65</b>
23.1 Functions . . . . .	65
23.2 Variables . . . . .	65
23.3 Class Parser . . . . .	65
23.3.1 Methods . . . . .	65
23.3.2 Properties . . . . .	66
23.4 Class CSVParser . . . . .	66
23.4.1 Methods . . . . .	66
23.4.2 Properties . . . . .	66
<b>24 Module hal.ml.features</b>	<b>67</b>
24.1 Functions . . . . .	67
<b>25 Package hal.ml.models</b>	<b>68</b>
25.1 Modules . . . . .	68
25.2 Variables . . . . .	68
<b>26 Module hal.ml.models.classification</b>	<b>69</b>
26.1 Functions . . . . .	69
<b>27 Module hal.ml.models.pipelined</b>	<b>70</b>
27.1 Functions . . . . .	70
<b>28 Module hal.ml.models.regression</b>	<b>71</b>
28.1 Functions . . . . .	71
<b>29 Module hal.ml.models.time_series</b>	<b>72</b>

29.1 Functions . . . . .	72
<b>30 Module hal.ml.predict</b>	<b>73</b>
30.1 Class BasePrediction . . . . .	73
30.1.1 Methods . . . . .	73
30.1.2 Properties . . . . .	73
<b>31 Package hal.ml.utils</b>	<b>74</b>
31.1 Modules . . . . .	74
31.2 Variables . . . . .	74
<b>32 Module hal.ml.utils.matrix</b>	<b>75</b>
32.1 Functions . . . . .	75
<b>33 Module hal.ml.utils.misc</b>	<b>78</b>
33.1 Functions . . . . .	78
<b>34 Package hal.mongodb</b>	<b>80</b>
34.1 Modules . . . . .	80
34.2 Variables . . . . .	80
<b>35 Module hal.mongodb.utils</b>	<b>81</b>
35.1 Functions . . . . .	81
<b>36 Package hal.profile</b>	<b>82</b>
36.1 Modules . . . . .	82
36.2 Variables . . . . .	82
<b>37 Module hal.profile.mem</b>	<b>83</b>
37.1 Functions . . . . .	83
<b>38 Module hal.profile.performance</b>	<b>84</b>
38.1 Class EightQueenTest . . . . .	84
38.1.1 Methods . . . . .	84
38.1.2 Properties . . . . .	85
<b>39 Package hal.streams</b>	<b>86</b>
39.1 Modules . . . . .	86
39.2 Variables . . . . .	86
<b>40 Module hal.streams.pretty_table</b>	<b>87</b>
40.1 Functions . . . . .	87
40.2 Variables . . . . .	88
<b>41 Package hal.strings</b>	<b>89</b>
41.1 Modules . . . . .	89
41.2 Variables . . . . .	89
<b>42 Module hal.strings.utils</b>	<b>90</b>
42.1 Functions . . . . .	90
42.2 Variables . . . . .	90
<b>43 Package hal.time</b>	<b>91</b>
43.1 Modules . . . . .	91

---

43.2 Variables . . . . .	91
<b>44 Module hal.time.profile</b>	<b>92</b>
44.1 Functions . . . . .	92
44.2 Variables . . . . .	92
<b>45 Module hal.time.utils</b>	<b>93</b>
45.1 Functions . . . . .	93
45.2 Variables . . . . .	93
<b>46 Package hal.wrappers</b>	<b>94</b>
46.1 Modules . . . . .	94
46.2 Variables . . . . .	94
<b>47 Module hal.wrappers.methods</b>	<b>95</b>
47.1 Functions . . . . .	95
<b>Index</b>	<b>96</b>

# 1 Package hal

## 1.1 Modules

- **charts** (Section 2, p. 9)
  - **bar** (Section 3, p. 10)
  - **correlation** (Section 4, p. 11)
  - **plotter**: Show elegant plots in any dimension.  
(Section 5, p. 12)
- **files** (Section 6, p. 16)
  - **models**: Main entities in files, such as documents, folders.  
(Section 7, p. 17)
  - **save\_as** (Section 8, p. 27)
- **internet** (Section 9, p. 28)
  - **engines**: Abstract search engines.  
(Section 10, p. 29)
  - **github**: Common classes and entities in Github  
(Section 11, p. 31)
  - **parser**: Parse anything there is on the Internet.  
(Section 12, p. 36)
  - **selenium**: Some utils methods for a selenium webdriver  
(Section 13, p. 38)
  - **web**: Deal with webpages.  
(Section 14, p. 40)
  - **youtube**: Get rss feed for youtube channel.  
(Section 15, p. 43)
- **maths**: MATHS: important and scalable math functions  
(Section 16, p. 45)
  - **crypt**: Perform fast hash, encryption and calculations related to cryptography.  
(Section 17, p. 46)
  - **maths**: A few elegant and powerful mathematical functions.  
(Section 18, p. 57)
- **ml** (Section 19, p. 60)
  - **analysis** (Section 20, p. 61)
    - \* **correlation** (Section 21, p. 62)
  - **data** (Section 22, p. 64)
    - \* **parser**: Parsers for raw databases.  
(Section 23, p. 65)
  - **features**: Collection of methods to find weights of features and select the best ones.  
(Section 24, p. 67)
  - **models** (Section 25, p. 68)
    - \* **classification**: Prediction methods based on classification algorithms.  
(Section 26, p. 69)
    - \* **pipelined**: Prediction methods based on multiple models mixed up.  
(Section 27, p. 70)
    - \* **regression**: Prediction methods based on regression algorithms.  
(Section 28, p. 71)
    - \* **time\_series**: Multi-purpose prediction methods to be used in time-series.  
(Section 29, p. 72)
  - **predict**: " General model to make prediction about everything.  
(Section 30, p. 73)

- **utils** (Section 31, p. 74)
  - \* **matrix**: Functions to deal with matrices.  
(Section 32, p. 75)
  - \* **misc**: Various tools and utilities to deal with database and machine learning.  
(Section 33, p. 78)
- **mongodb** (Section 34, p. 80)
  - **utils**: Various utilities to deal with MondoDB databases  
(Section 35, p. 81)
- **profile** (Section 36, p. 82)
  - **mem** (Section 37, p. 83)
  - **performance**: Perform benchmarks and tests on your PC.  
(Section 38, p. 84)
- **streams** (Section 39, p. 86)
  - **pretty\_table**: Pretty prints table in SQL style  
(Section 40, p. 87)
- **strings** (Section 41, p. 89)
  - **utils**: Typical operations on strings made easy  
(Section 42, p. 90)
- **time** (Section 43, p. 91)
  - **profile** (Section 44, p. 92)
  - **utils** (Section 45, p. 93)
- **wrappers** (Section 46, p. 94)
  - **methods**: Typical (and useful) function wrappers  
(Section 47, p. 95)

## 1.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None



## 2 Package hal.charts

### 2.1 Modules

- **bar** (*Section 3, p. 10*)
- **correlation** (*Section 4, p. 11*)
- **plotter**: Show elegant plots in any dimension.  
(*Section 5, p. 12*)

### 2.2 Variables

Name	Description
__package__	<b>Value:</b> None

## 3 Module *hal.charts.bar*

### 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_symlog_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  
    Symlog bar chart
```

## 4 Module *hal.charts.correlation*

### 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**(*vectorx*, *vectory*)

:param *vectorx*: vector in x axis  
 :param *vectory*: vector in y axis  
 :return: 2d scatter plot

**param**(*self*, *functionx*, *functiony*, *min*, *max*, *points*)

:param *functionx*: function in x value  
 :param *functiony*: function in y value  
 :param *min*: minimum value  
 :param *max*: maximum value  
 :param *points*: number of points to display  
 :return: 2d parametric graph of given function from min to max

**plot**(*self*, *function*, *min*, *max*, *points*)

:param *function*: function to plot  
 :param *min*: minimum value  
 :param *max*: maximum value  
 :param *points*: number of points  
 :return: plot 2d function

*Inherited from object*

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

#### 5.1.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

## 5.2 Class Plot3d



### 5.2.1 Methods

**scatter**(*vectorx, vectory, vectorz*)

---

```

:param vectorx: vector in x axis
:param vectory: vector in y axis
:param vectorz: vector in z axis
:return: plot 3d scattered points
  
```

**param**(*self, functionx, functiony, functionz, min, max, points*)

---

```

:param functionx: function in x
:param functiony: function in y
:param functionz: function in z
:param min: minimum
:param max: maximum
:param points: number of points
:return: 3d parametric graph of given function from min to max
  
```

```
plot(self, function, minx, maxx, pointsx, miny, maxy, pointsy)
```

```
:param function: function to plot
:param minx: minimum of x-values
:param maxx: maximum of x-values
:param pointsx: points in x axis
:param miny: minimum of y-values
:param maxy: maximum of y-values
:param pointsy: points in y axis
:return: plot 3d function
```

### *Inherited from object*

```
--delattr--(), --format--(), --getattr__(), --hash--(), --init--(), --new--(), --reduce--(),
--reduce_ex--(), --repr--(), --setattr--(), --sizeof--(), --str--(), --subclasshook--()
```

#### 5.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

### 5.3 Class Plot4d

```
object └─ hal.charts.plotter.Plot4d
```

#### 5.3.1 Methods

```
scatter(vectorx, vectory, vectorz, vectorw)
```

```
:param vectorx: vector in x axis
:param vectory: vector in y axis
:param vectorz: vector in z axis
:param vectorw: vector in w axis
:return: plot 4d scattered points
```

**param**(*self, functionx, functiony, functionz, functionw, min, max, points*)

```
:param functionx: function in x
:param functiony: function in y
:param functionz: function in z
:param functionw: function in w
:param min: minimum
:param max: maximum
:param points: number of points
:return: 4d parametric graph of given function from min to max
```

**plot**(*self, function, minx, maxx, miny, maxy, minz, maxz, precision, kind*)

```
:param function: function to plot
:param minx: minimum of x-values
:param maxx: maximum of x-values
:param miny: minimum of y-values
:param maxy: maximum of y-values
:param minz: minimum of z-values
:param maxz: 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__(), __getattr__(), __hash__(), __init__(), __new__(), __reduce__(),
__reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

### 5.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## 6 Package *hal.files*

### 6.1 Modules

- **models:** Main entities in files, such as documents, folders.  
(Section 7, p. 17)
- **save\_as** (Section 8, p. 27)

### 6.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None



## 7 Module *hal.files.models*

Main entities in files, such as documents, folders.

### 7.1 Variables

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

### 7.2 Class *FileSystem*

object —  
     *hal.files.models.FileSystem*

#### 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 barckets.
```

**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, bad_chars=BAD_CHARS, r=" ")
```

```
:param name: string
    Name to edit
:param bad_chars: []
    List of bad strings to remove
:param r: 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.
```

**ls**(*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

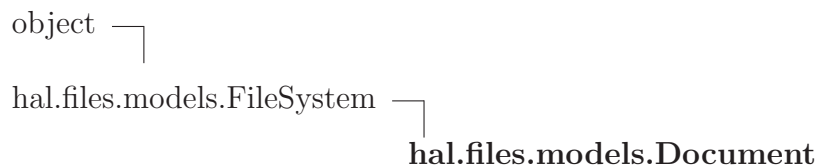
### *Inherited from object*

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

### 7.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

### 7.3 Class Document



#### 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.

<b>is_image</b> ( <i>self</i> )
:return: True iff document is an image.

<b>is_audio</b> ( <i>self</i> )
:return: True iff document is an audio.

<b>is_hidden</b> ( <i>self</i> )
: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(), ls(), ls\_dir(),  
ls\_recurse(), prettify(), remove\_brackets(), remove\_year(), rename(), trash()

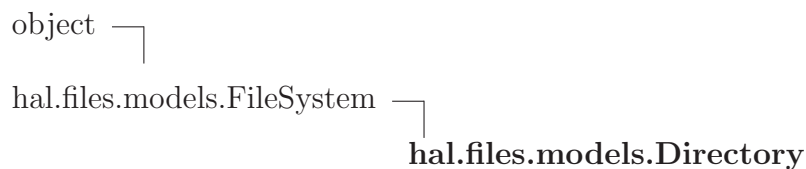
*Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(),  
\_\_repr\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

### 7.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## 7.4 Class Directory



## 7.4.1 Methods

<code>__init__(self, path)</code>
-----------------------------------

: param path: string Path to file
--------------------------------------

Overrides: object.__init__
----------------------------

<code>create_new(path)</code>
-------------------------------

: param path: string Path to directory to create
---

: return: void Creates new directory
---

<code>get_path_name(self)</code>
----------------------------------

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

<code>is_empty(self)</code>
-----------------------------

: 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(), ls(), ls\_dir(),  
ls\_recurse(), prettify(), remove\_brackets(), remove\_year(), rename(), trash()

*Inherited from **object***

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(),  
\_\_repr\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

## 7.4.2 Properties

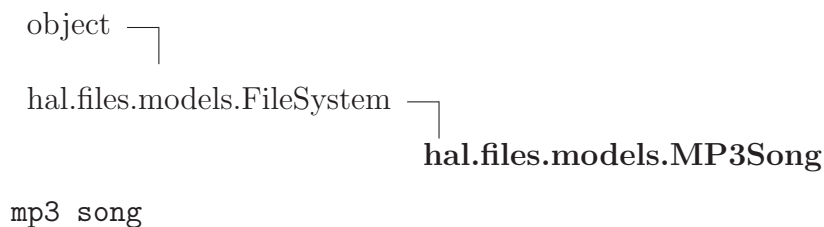
Name	Description
<i>Inherited from object</i>	

*continued on next page*



Name	Description
<code>--class--</code>	

## 7.5 Class MP3Song



### 7.5.1 Methods

```

__init__(self, path)

:param path: string
    Path to file
Overrides: object.__init__ extit(inherited documentation)

```

```
set_name(self, name)
```

```
set_artist(self, artist)
```

```
set_album(self, album)
```

```
set_nr_track(self, nr_track)
```

```
set_year(self, year)
```

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

```

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

```

*Inherited from object*

```

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

```

### 7.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

## 8 Module *hal.files.save\_as*

### 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
<code>__package__</code>	Value: <code>'hal.files'</code>

## 9 Package *hal.internet*

### 9.1 Modules

- **engines**: Abstract search engines.  
(Section 10, p. 29)
- **github**: Common classes and entities in Github  
(Section 11, p. 31)
- **parser**: Parse anything there is on the Internet.  
(Section 12, p. 36)
- **selenium**: Some utils methods for a selenium webdriver  
(Section 13, p. 38)
- **web**: Deal with webpages.  
(Section 14, p. 40)
- **youtube**: Get rss feed for youtube channel.  
(Section 15, p. 43)

### 9.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None

## 10 Module *hal.internet.engines*

Abstract search engines.

### 10.1 Class *SearchEngineResult*



#### 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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __subclasshook__()
```

#### 10.1.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

### 10.2 Class *SearchEngine*



**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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),  
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

**10.2.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 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
```

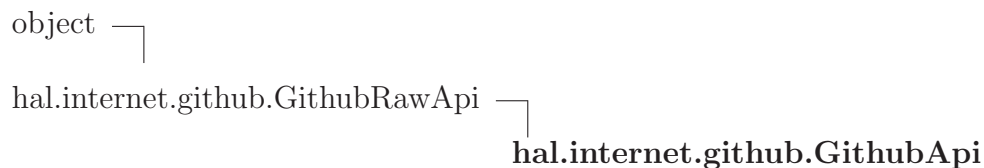
*Inherited from object*

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

### 11.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 11.3 Class GithubApi



Wrapper for generic Github API

### 11.3.1 Methods

<b><code>__init__(self, api_type)</code></b> <hr/> <p><code>:param api_type: str</code>          Type of API to build          Overrides: <code>object.__init__</code></p>
<b><code>get_trending_daily()</code></b> <hr/> <p><code>:return: []</code>          List of GithubUserRepository</p>

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

`__getitem__()`

*Inherited from object*

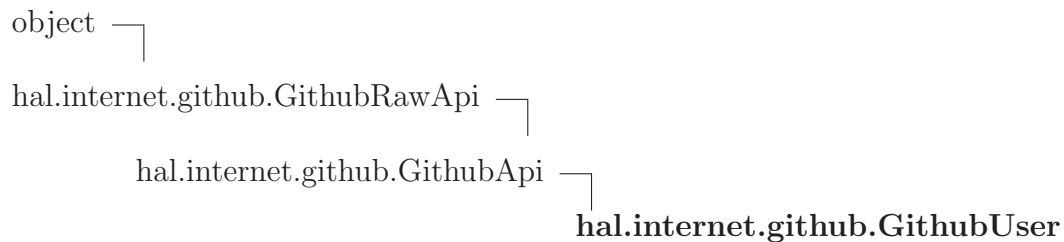
`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`



### 11.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 11.4 Class 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: []
    List of GithubUserRepository
```

```
get_starred_repos(self)
```

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

```
get_trending_daily_not_starred(self)
```

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

get\_trending\_daily()

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

\_\_getitem\_\_()

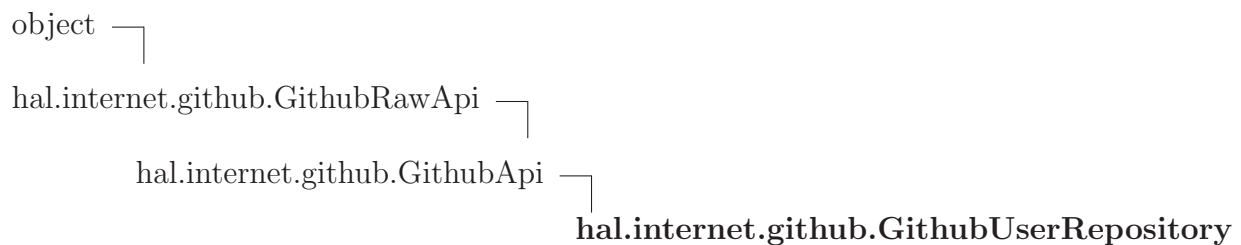
*Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(),  
\_\_repr\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

#### 11.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

### 11.5 Class GithubUserRepository



Model of a generic Github user repository

#### 11.5.1 Methods

<b>__init__</b> (self, username, repository_name)
<pre> :param username: str     Username of user :param repository_name: str     Name of repository Overrides: object.__init__ </pre>

<code>--eq--(self, other)</code>
----------------------------------

*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--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,  
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

### 11.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

## 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
<code>--package--</code>	<b>Value:</b> <code>'hal.internet'</code>

### 12.3 Class *HtmlTable*



**12.3.1 Methods**

<code>__init__(self, html_source)</code>
--

<code>:param html_source: string</code> Html source of table
---

Overrides: <code>object.__init__</code>
---

<code>parse(self)</code>
--------------------------

<code>:return: list of list</code> List of list of values in table
---

***Inherited from str***

`__add__()`, `__contains__()`, `__eq__()`, `__format__()`, `__ge__()`, `__getattr__()`, `__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
<i>Inherited from object</i>	
<code>__class__</code>	

## 13 Module hal.internet.selenium

Some utils methods for a selenium webdriver

### 13.1 Variables

Name	Description
<code>--package--</code>	Value: None

### 13.2 Class SeleniumForm

Great and simple static methods to deal with selenium webdrivers.

#### 13.2.1 Methods

**fill\_form\_field**(*browser, field\_name, field\_value*)

```
:param browser: webdriver
    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 wiht given value.
```

**fill\_login\_form**(*browser, username, username\_field, userpassword, userpassword\_field*)

---

```
:param browser: webdriver
    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 userpassword: string
    Password of user to login.
:param userpassword_field: string
    Name of field to fill with userpassword.
:return: void
    Form filled with given information.
```

**submit\_form**(*browser, button\_name*)

---

```
:param browser: webdriver
    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 webpages.

### 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	<b>Value:</b> ["Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWe..."]

*continued on next page*



Name	Description
URL_VALID_REGEX	<b>Value:</b> <code>re.compile(r"^(?:http ftp)s?://" + r"(?: (? : [A-Z0-9] (? : [A-Z0...</code>

### 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 wendrive and go to webpage
```

### *Inherited from object*

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

#### 14.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 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 t
:param youtube_channel_url: string
    base url of youtube channels.
@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 t
: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"
: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 t
: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	<b>Value:</b> "https://www.youtube.com/user/"
YOUTUBE_FEED_BASE-URL	<b>Value:</b> "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. 46)
- **maths**: A few elegant and powerful mathematical functions.  
(Section 18, p. 57)

### 16.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> 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\_\_ exitit(inherited documentation)

hash( <i>self</i> )
:return: hash plaintext

*Inherited from object*

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

### 17.1.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

## 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 signatureOverrides: `object.__init__` `exitit` (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__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,  
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`**17.2.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

**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, salt=None)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature  
 Overrides: `object.__init__` `exitit`(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



<b>hash_sha384(<i>self</i>)</b>
:return: sha384 hash

<b>hash_sha512(<i>self</i>)</b>
:return: sha512 hash

<b>hash_shasalted(<i>self</i>)</b>
:return: sha512 hash

***Inherited from object***

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,  
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

**17.3.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

**17.3.3 Class Variables**

Name	Description
ALLOWED_SIZE	<b>Value:</b> [1, 224, 256, 384, 512]

**17.4 Class DES**

```

object └─
          hal.maths.crypt.DES
DES hash
```

## 17.4.1 Methods

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

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

Overrides: `object.__init__` `exitit`(inherited documentation)

**`hash(self)`**

:return: hash of given size

**`hash_des(self)`**

:return: des hash

**`hash_des3(self)`**

:return: des3 hash

*Inherited from object*

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

## 17.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 17.4.3 Class Variables

Name	Description
<code>ALLOWED_SIZE</code>	<b>Value:</b> [1, 3]

## 17.5 Class ARC

```

object └─
        hal.maths.crypt.ARC

```

ARC hash

### 17.5.1 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__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,  
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

### 17.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

### 17.5.3 Class Variables



hmac 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: hash plaintext

#### *Inherited from object*

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

### 17.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 17.8 Class BLOWFISH

```

object └─
          hal.maths.crypt.BLOWFISH

```

blowfish 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)

```
hash(self)
```

```
:return: hash plaintext
```

### *Inherited from object*

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

### 17.8.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 17.9 Class IDEA

```
object └─ hal.maths.crypt.IDEA
```

IDEA hash

### 17.9.1 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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

### 17.9.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 17.10 Class CAST128

```
object └─
          hal.maths.crypt.CAST128
```

CAST 128 hash

### 17.10.1 Methods

```
__init__(self, string, key)
```

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

```
Overrides: object.__init__ extit(inherited documentation)
```

```
encrypt(self)
```

```
decrypt(self)
```

### *Inherited from object*

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

### 17.10.2 Properties





## 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

**blumblumshub**(*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
<code>--package--</code>	<b>Value:</b> <code>'hal.maths'</code>

### 18.3 Class Integer

object └─ **hal.maths.maths.Integer**

#### 18.3.1 Methods

**--init--**(*self, string*)

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature  
Overrides: `object.__init__` extit(inherited documentation)

```
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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

#### 18.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

#### 18.3.3 Class Variables

Name	Description
LOW_PRIMES	<b>Value:</b> [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, ...]

## 18.4 Class EightQueen

```
object └─ hal.maths.maths.EightQueen
```

8 queen problem solver

**18.4.1 Methods**

<code>__init__(self, board_size)</code>
---

<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature
--

Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)
--

<code>under_attack(col, queens)</code>
--

<code>solve(self, n)</code>
-----------------------------

***Inherited from object***

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

**18.4.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 19 Package hal.ml

### 19.1 Modules

- **analysis** (*Section 20, p. 61*)
  - **correlation** (*Section 21, p. 62*)
- **data** (*Section 22, p. 64*)
  - **parser**: Parsers for raw databases.  
(*Section 23, p. 65*)
- **features**: Collection of methods to find weights of features and select the best ones.  
(*Section 24, p. 67*)
- **models** (*Section 25, p. 68*)
  - **classification**: Prediction methods based on classification algorithms.  
(*Section 26, p. 69*)
  - **pipelined**: Prediction methods based on multiple models mixed up.  
(*Section 27, p. 70*)
  - **regression**: Prediction methods based on regression algorithms.  
(*Section 28, p. 71*)
  - **time\_series**: Multi-purpose prediction methods to be used in time-series.  
(*Section 29, p. 72*)
- **predict**: " General model to make prediction about everything.  
(*Section 30, p. 73*)
- **utils** (*Section 31, p. 74*)
  - **matrix**: Functions to deal with matrices.  
(*Section 32, p. 75*)
  - **misc**: Various tools and utilities to deal with database and machine learning.  
(*Section 33, p. 78*)

### 19.2 Variables

Name	Description
--package--	<b>Value:</b> None

## 20 Package *hal.ml.analysis*

### 20.1 Modules

- **correlation** (*Section 21, p. 62*)

### 20.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None

## 21 Module *hal.ml.analysis.correlation*

### 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. 65)

### 22.2 Variables

Name	Description
--package--	<b>Value:</b> 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
<code>--package--</code>	<b>Value:</b> <code>'hal.ml.data'</code>

### 23.3 Class Parser

object —  
    *hal.ml.data.parser.Parser*

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

#### 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*)

*Inherited from object*

```
--delattr--(), --format--(), --getattrattribute--(), --hash--(), --new--(), --reduce--(), --reduce_ex--(),
```

`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

### 23.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 23.4 Class CSVParser



### 23.4.1 Methods

<code>__init__(self, database_file)</code>
:param database_file: a raw .csv file that contains any data about anything
Overrides: <code>object.__init__</code>

<code>parse_data(self)</code>
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__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

### 23.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 24 Module *hal.ml.features*

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

### 24.1 Functions

<code>select_k_best(<math>x, y, k</math>)</code>
select k best features in dataset

<code>get_best_features(<math>x, y</math>)</code>
finds the optimal number of features

<code>get_features(<math>x, y, n\_features\_to\_select</math>)</code>
finds the optimal features

## 25 Package *hal.ml.models*

### 25.1 Modules

- **classification**: Prediction methods based on classification algorithms.  
(Section 26, p. 69)
- **pipelined**: Prediction methods based on multiple models mixed up.  
(Section 27, p. 70)
- **regression**: Prediction methods based on regression algorithms.  
(Section 28, p. 71)
- **time\_series**: Multi-purpose prediction methods to be used in time-series.  
(Section 29, p. 72)

### 25.2 Variables

Name	Description
--package--	Value: None

## 26 Module `hal.ml.models.classification`

Prediction methods based on classification algorithms.

### 26.1 Functions

<code>extra_trees_classifier()</code>
---------------------------------------

<code>random_forest()</code>
------------------------------

<code>knn()</code>
--------------------

very fast and slightly more accurate than AdaBoost
--

<code>ada_boost()</code>
--------------------------

fast, accurate but too uncertainty
------------------------------------

<code>bayes_gauss()</code>
----------------------------

slower than svr but equally accurate
--------------------------------------

<code>bayes_bernoulli()</code>
--------------------------------

## 27 Module `hal.ml.models.pipelined`

Prediction methods based on multiple models mixed up.

### 27.1 Functions

<code>logistic_rbm()</code>
-----------------------------

<code>anova_svm()</code>
--------------------------

## 28 Module `hal.ml.models.regression`

Prediction methods based on regression algorithms.

### 28.1 Functions

<code>support_vector_machine()</code>
---------------------------------------

<code>super fast and precise</code>
-------------------------------------

<code>logistic_regression()</code>
------------------------------------

## 29 Module `hal.ml.models.time_series`

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

### 29.1 Functions

**`test_stationarity`**(*timeseries*)

**`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, 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

**`dynamic_var`**(*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



## 30 Module *hal.ml.predict*

" General model to make prediction about everything.

### 30.1 Class *BasePrediction*

object └─  
          **hal.ml.predict.BasePrediction**

#### 30.1.1 Methods

**\_\_init\_\_**(*self*, *model*, *rounds*)

*x*.**\_\_init\_\_**(...) initializes *x*; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

**train**(*self*, *x*, *y*)

#### *Inherited from object*

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

#### 30.1.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 31 Package `hal.ml.utils`

### 31.1 Modules

- **matrix**: Functions to deal with matrices.  
(Section 32, p. 75)
- **misc**: Various tools and utilities to deal with database and machine learning.  
(Section 33, p. 78)

### 31.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None

## 32 Module *hal.ml.utils.matrix*

Functions to deal with matrices.

### 32.1 Functions

**precision**(*matrix*)

Calculates accuracy on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

False Positive - True Negative

**recall**(*matrix*)

Calculates recall on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

False Positive - True Negative

**tn\_rate**(*matrix*)

Calculates true negative rate on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

False Positive - True Negative

**accuracy**(*matrix*)

Calculates recall on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

False Positive - True Negative

**f1\_score**(*matrix*)

Calculates f1 score on database

:param matrix: 2x2 matrix that looks like  
 True Positive - False Negative  
                   |                  |  
                   -                  -  
 False Positive - True Negative

**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

**precision**(*matrix*)

Calculates accuracy on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

| - |

False Positive - True Negative

**recall**(*matrix*)

Calculates recall on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

| - |

False Positive - True Negative

**tn\_rate**(*matrix*)

Calculates true negative rate on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

| - |

False Positive - True Negative

**accuracy**(*matrix*)

Calculates recall on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

| - |

False Positive - True Negative

**f1\_score**(*matrix*)

Calculates f1 score on database

:param matrix: 2x2 matrix that looks like

True Positive - False Negative

	-	
False Positive	-	True Negative

**pearson**(*x, y*)

Pearson coefficient of arrays

**normalize\_array**(*a*)

:param a: [] 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. 81*)

### 34.2 Variables

Name	Description
--package--	<b>Value:</b> 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** (*Section 37, p. 83*)
- **performance**: Perform benchmarks and tests on your PC.  
(*Section 38, p. 84*)

### 36.2 Variables

Name	Description
__package__	<b>Value:</b> None

## 37 Module *hal.profile.mem*

### 37.1 Functions

<code>get_memory_usage()</code>
<hr/>
<code>:return: float</code> MB of memory used by this process

<code>force_garbage_collect()</code>
<hr/>
<code>:return: void</code> 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__ exitit(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
```

<b>update_std_out_and_log</b> ( <i>self</i> , <i>string</i> )
---

<pre>:param string: string     Stuff to print :return: void     Prints to stdout and updates log</pre>
--

<b>start</b> ( <i>self</i> )
------------------------------

***Inherited from object***

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

**38.1.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 39 Package hal.streams

### 39.1 Modules

- **pretty\_table**: Pretty prints table in SQL style  
(Section 40, p. 87)

### 39.2 Variables

Name	Description
--package--	<b>Value:</b> 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
<code>__package__</code>	<b>Value:</b> None



## 41 Package *hal.strings*

### 41.1 Modules

- **utils**: Typical operations on strings made easy  
(*Section 42, p. 90*)

### 41.2 Variables

Name	Description
--package--	<b>Value:</b> None

## 42 Module *hal.strings.utils*

Typical operations on strings made easy

### 42.1 Functions

**how\_similar\_are**(*a*, *b*)

```
:param a: str
    First string
:param b: 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
<code>--package--</code>	<b>Value:</b> <code>'hal.strings'</code>

## 43 Package hal.time

### 43.1 Modules

- **profile** (*Section 44, p. 92*)
- **utils** (*Section 45, p. 93*)

### 43.2 Variables

Name	Description
--package--	<b>Value:</b> None

## 44 Module *hal.time.profile*

### 44.1 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
```

### 44.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> <code>'hal.time'</code>

## 45 Module *hal.time.utils*

### 45.1 Functions

**parse\_hh\_mm\_ss(*h*)**

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

**get\_seconds(*s*)**

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

**parse\_hh\_mm(*h*)**

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

### 45.2 Variables

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

## 46 Package *hal.wrappers*

### 46.1 Modules

- **methods:** Typical (and useful) function wrappers  
(*Section 47, p. 95*)

### 46.2 Variables

Name	Description
--package--	<b>Value:</b> None

## 47 Module *hal.wrappers.methods*

Typical (and useful) function wrappers

### 47.1 Functions

<code>handle_exceptions(<i>function</i>)</code>
---

<code>:param function: callback function                   function to wrap :return: callback function return type           wraps callback function</code>
---

## Index

- hal (*package*), 7–8
  - hal.charts (*package*), 9
    - hal.charts.bar (*module*), 10
    - hal.charts.correlation (*module*), 11
    - hal.charts.plotter (*module*), 12–15
  - hal.files (*package*), 16
    - hal.files.models (*module*), 17–26
    - hal.files.save\_as (*module*), 27
  - hal.internet (*package*), 28
    - hal.internet.engines (*module*), 29–30
    - hal.internet.github (*module*), 31–35
    - hal.internet.parser (*module*), 36–37
    - hal.internet.selenium (*module*), 38–39
    - hal.internet.web (*module*), 40–42
    - hal.internet.youtube (*module*), 43–44
  - hal.maths (*package*), 45
    - hal.maths.crypt (*module*), 46–56
    - hal.maths.maths (*module*), 57–59
  - hal.ml (*package*), 60
    - hal.ml.analysis (*package*), 61
    - hal.ml.data (*package*), 64
    - hal.ml.features (*module*), 67
    - hal.ml.models (*package*), 68
    - hal.ml.predict (*module*), 73
    - hal.ml.utils (*package*), 74
  - hal.mongodb (*package*), 80
    - hal.mongodb.utils (*module*), 81
  - hal.profile (*package*), 82
    - hal.profile.mem (*module*), 83
    - hal.profile.performance (*module*), 84–85
  - hal.streams (*package*), 86
    - hal.streams.pretty\_table (*module*), 87–88
  - hal.strings (*package*), 89
    - hal.strings.utils (*module*), 90
  - hal.time (*package*), 91
    - hal.time.profile (*module*), 92
    - hal.time.utils (*module*), 93
  - hal.wrappers (*package*), 94
    - hal.wrappers.methods (*module*), 95