

# PyHal

## API Documentation

January 10, 2017

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Package hal</b>	<b>5</b>
1.1 Modules . . . . .	5
1.2 Variables . . . . .	6
<b>2 Package hal.files</b>	<b>7</b>
2.1 Modules . . . . .	7
2.2 Variables . . . . .	7
<b>3 Module hal.files.models</b>	<b>8</b>
3.1 Variables . . . . .	8
3.2 Class FileSystem . . . . .	8
3.2.1 Methods . . . . .	8
3.2.2 Properties . . . . .	11
3.3 Class Document . . . . .	11
3.3.1 Methods . . . . .	11
3.3.2 Properties . . . . .	14
3.4 Class Directory . . . . .	14
3.4.1 Methods . . . . .	14
3.4.2 Properties . . . . .	15
3.5 Class MP3Song . . . . .	15
3.5.1 Methods . . . . .	16
3.5.2 Properties . . . . .	16
<b>4 Package hal.internet</b>	<b>17</b>
4.1 Modules . . . . .	17
4.2 Variables . . . . .	17
<b>5 Module hal.internet.engines</b>	<b>18</b>
5.1 Class SearchEngineResult . . . . .	18
5.1.1 Methods . . . . .	18
5.1.2 Properties . . . . .	18
5.2 Class SearchEngine . . . . .	18
5.2.1 Methods . . . . .	19
5.2.2 Properties . . . . .	19
<b>6 Module hal.internet.parser</b>	<b>20</b>

6.1	Functions . . . . .	20
6.2	Variables . . . . .	20
6.3	Class HtmlTable . . . . .	20
6.3.1	Methods . . . . .	21
6.3.2	Properties . . . . .	21
<b>7</b>	<b>Module hal.internet.selenium</b>	<b>22</b>
7.1	Variables . . . . .	22
7.2	Class SeleniumForm . . . . .	22
7.2.1	Methods . . . . .	22
<b>8</b>	<b>Module hal.internet.web</b>	<b>24</b>
8.1	Functions . . . . .	24
8.2	Variables . . . . .	24
8.3	Class Webpage . . . . .	24
8.3.1	Methods . . . . .	25
8.3.2	Properties . . . . .	26
<b>9</b>	<b>Module hal.internet.youtube</b>	<b>27</b>
9.1	Functions . . . . .	27
<b>10</b>	<b>Package hal.maths</b>	<b>28</b>
10.1	Modules . . . . .	28
10.2	Variables . . . . .	28
<b>11</b>	<b>Module hal.maths.crypt</b>	<b>29</b>
11.1	Variables . . . . .	29
11.2	Class MD5 . . . . .	29
11.2.1	Methods . . . . .	29
11.2.2	Properties . . . . .	29
11.3	Class MD6 . . . . .	30
11.3.1	Methods . . . . .	30
11.3.2	Properties . . . . .	30
11.3.3	Class Variables . . . . .	31
11.4	Class SHA . . . . .	31
11.4.1	Methods . . . . .	31
11.4.2	Properties . . . . .	32
11.4.3	Class Variables . . . . .	32
11.5	Class DES . . . . .	33
11.5.1	Methods . . . . .	33
11.5.2	Properties . . . . .	33
11.5.3	Class Variables . . . . .	34
11.6	Class ARC . . . . .	34
11.6.1	Methods . . . . .	34
11.6.2	Properties . . . . .	35
11.6.3	Class Variables . . . . .	35
11.7	Class AES . . . . .	35
11.7.1	Methods . . . . .	35
11.7.2	Properties . . . . .	35
11.8	Class HMAC . . . . .	36
11.8.1	Methods . . . . .	36
11.8.2	Properties . . . . .	36

11.9 Class BLOWFISH . . . . .	36
11.9.1 Methods . . . . .	37
11.9.2 Properties . . . . .	37
11.10 Class IDEA . . . . .	37
11.10.1 Methods . . . . .	37
11.10.2 Properties . . . . .	38
11.11 Class CAST128 . . . . .	38
11.11.1 Methods . . . . .	39
11.11.2 Properties . . . . .	39
11.12 Class Dsa . . . . .	39
11.12.1 Methods . . . . .	39
11.12.2 Properties . . . . .	40
<b>12 Module hal.maths.maths</b>	<b>41</b>
12.1 Functions . . . . .	41
12.2 Variables . . . . .	41
12.3 Class Integer . . . . .	41
12.3.1 Methods . . . . .	41
12.3.2 Properties . . . . .	42
12.3.3 Class Variables . . . . .	42
12.4 Class EightQueen . . . . .	42
12.4.1 Methods . . . . .	43
12.4.2 Properties . . . . .	43
<b>13 Module hal.maths.plotter</b>	<b>44</b>
13.1 Class Plot2d . . . . .	44
13.1.1 Methods . . . . .	44
13.1.2 Properties . . . . .	45
13.2 Class Plot3d . . . . .	45
13.2.1 Methods . . . . .	45
13.2.2 Properties . . . . .	46
13.3 Class Plot4d . . . . .	46
13.3.1 Methods . . . . .	46
13.3.2 Properties . . . . .	47
<b>14 Package hal.ml</b>	<b>48</b>
14.1 Modules . . . . .	48
14.2 Variables . . . . .	48
<b>15 Package hal.ml.data</b>	<b>49</b>
15.1 Modules . . . . .	49
15.2 Variables . . . . .	49
<b>16 Module hal.ml.data.parser</b>	<b>50</b>
16.1 Variables . . . . .	50
16.2 Class Parser . . . . .	50
16.2.1 Methods . . . . .	50
16.2.2 Properties . . . . .	50
16.3 Class CSVParser . . . . .	51
16.3.1 Methods . . . . .	51
16.3.2 Properties . . . . .	51

<b>17 Module hal.ml.features</b>	<b>52</b>
17.1 Functions . . . . .	52
17.2 Variables . . . . .	52
<b>18 Package hal.ml.models</b>	<b>53</b>
18.1 Modules . . . . .	53
18.2 Variables . . . . .	53
<b>19 Module hal.ml.models.classification</b>	<b>54</b>
19.1 Functions . . . . .	54
19.2 Variables . . . . .	54
<b>20 Module hal.ml.models.pipelined</b>	<b>55</b>
20.1 Functions . . . . .	55
20.2 Variables . . . . .	55
<b>21 Module hal.ml.models.regression</b>	<b>56</b>
21.1 Functions . . . . .	56
21.2 Variables . . . . .	56
<b>22 Module hal.ml.models.time_series</b>	<b>57</b>
22.1 Functions . . . . .	57
<b>23 Module hal.ml.predict</b>	<b>58</b>
23.1 Variables . . . . .	58
23.2 Class BasePrediction . . . . .	58
23.2.1 Methods . . . . .	58
23.2.2 Properties . . . . .	58
<b>24 Module hal.ml.utils</b>	<b>59</b>
24.1 Functions . . . . .	59
<b>25 Package hal.profile</b>	<b>61</b>
25.1 Modules . . . . .	61
25.2 Variables . . . . .	61
<b>26 Module hal.profile.performance</b>	<b>62</b>
26.1 Variables . . . . .	62
26.2 Class EightQueenTest . . . . .	62
26.2.1 Methods . . . . .	62
26.2.2 Properties . . . . .	63
<b>27 Package hal.wrappers</b>	<b>64</b>
27.1 Modules . . . . .	64
27.2 Variables . . . . .	64
<b>28 Module hal.wrappers.methods</b>	<b>65</b>
28.1 Functions . . . . .	65
<b>Index</b>	<b>66</b>

# 1 Package hal

## 1.1 Modules

- **files** (*Section 2, p. 7*)
  - **models**: Main entities in files, such as documents, folders.  
(*Section 3, p. 8*)
- **internet** (*Section 4, p. 17*)
  - **engines**: Abstract search engines.  
(*Section 5, p. 18*)
  - **parser**: Parse anything there is on the Internet.  
(*Section 6, p. 20*)
  - **selenium**: Some utils methods for a selenium webdriver  
(*Section 7, p. 22*)
  - **web**: Deal with webpages.  
(*Section 8, p. 24*)
  - **youtube**: Get rss feed for youtube channel.  
(*Section 9, p. 27*)
- **maths**: MATHS: important and scalable math functions  
(*Section 10, p. 28*)
  - **crypt**: Perform fast hash, encryption and calculations related to cryptography.  
(*Section 11, p. 29*)
  - **maths**: A few elegant and powerful mathematical functions.  
(*Section 12, p. 41*)
  - **plotter**: Show elegant plots in any dimension.  
(*Section 13, p. 44*)
- **ml** (*Section 14, p. 48*)
  - **data** (*Section 15, p. 49*)
    - \* **parser**: Parsers for raw databases.  
(*Section 16, p. 50*)
  - **features**: Collection of methods to find weights of features and select the best ones.  
(*Section 17, p. 52*)
  - **models** (*Section 18, p. 53*)
    - \* **classification**: Prediction methods based on classification algorithms.  
(*Section 19, p. 54*)
    - \* **pipelined**: Prediction methods based on multiple models mixed up.  
(*Section 20, p. 55*)
    - \* **regression**: Prediction methods based on regression algorithms.  
(*Section 21, p. 56*)
    - \* **time\_series**: Multi-purpose prediction methods to be used in time-series.  
(*Section 22, p. 57*)
  - **predict**: " General model to make prediction about everything.  
(*Section 23, p. 58*)
  - **utils**: Various tools and utilities to deal with database and machine learning.  
(*Section 24, p. 59*)
- **profile** (*Section 25, p. 61*)
  - **performance**: Perform benchmarks and tests on your PC.  
(*Section 26, p. 62*)
- **wrappers** (*Section 27, p. 64*)
  - **methods**: Typical (and useful) function wrappers  
(*Section 28, p. 65*)

## 1.2 Variables

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

## 2 Package hal.files

### 2.1 Modules

- **models:** Main entities in files, such as documents, folders.  
(Section 3, p. 8)

### 2.2 Variables

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

### 3 Module *hal.files.models*

Main entities in files, such as documents, folders.

#### 3.1 Variables

Name	Description
BAD_CHARS	<b>Value:</b> ['.', ':', '"', '\xe2\x80\x99', '&', '720p', '1080p', 'yi...]
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> '/'
__package__	<b>Value:</b> 'hal.files'

#### 3.2 Class *FileSystem*

object —  
     *hal.files.models.FileSystem*

**Known Subclasses:** *hal.files.models.Directory*, *hal.files.models.Document*, *hal.files.models.MP3Song*

##### 3.2.1 Methods

<b>__init__</b> ( <i>self</i> , <i>path</i> )
<pre>:param path: string     Path to file Overrides: object.__init__</pre>



---

**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*=['.', ':', '"', '\xe2\x80\x99', '&', '720p', '1080p', 'yi...', *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
```

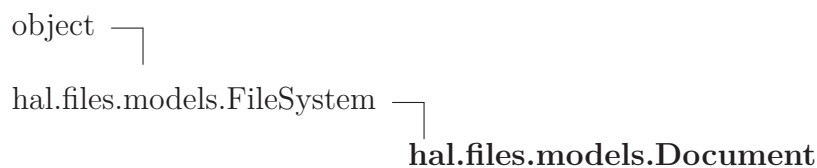
### *Inherited from object*

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

### 3.2.2 Properties

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

## 3.3 Class Document



### 3.3.1 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 3.2)*

*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***

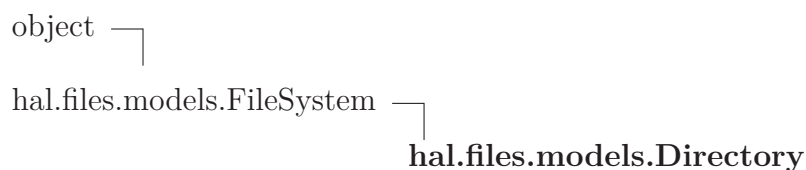
```

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

```

**3.3.2 Properties**

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

**3.4 Class Directory****3.4.1 Methods**

<b><code>__init__(self, path)</code></b>
<pre> :param path: string     Path to file Overrides: object.__init__ </pre>

<b><code>create_new(path)</code></b>
<pre> :param path: string     Path to directory to create :return: void     Creates new directory </pre>

```
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 3.2)*

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

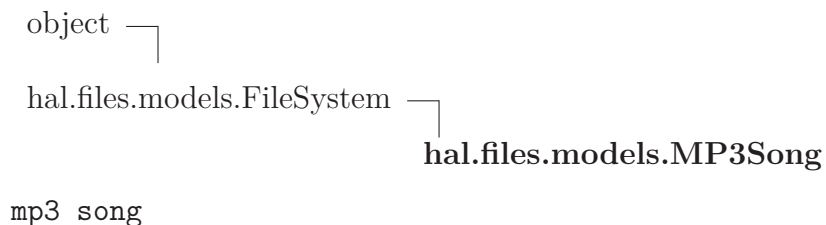
*Inherited from object*

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

### 3.4.2 Properties

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

## 3.5 Class MP3Song



### 3.5.1 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 3.2)*

```
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__()
```

### 3.5.2 Properties

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



## 4 Package *hal.internet*

### 4.1 Modules

- **engines:** Abstract search engines.  
(Section 5, p. 18)
- **parser:** Parse anything there is on the Internet.  
(Section 6, p. 20)
- **selenium:** Some utils methods for a selenium webdriver  
(Section 7, p. 22)
- **web:** Deal with webpages.  
(Section 8, p. 24)
- **youtube:** Get rss feed for youtube channel.  
(Section 9, p. 27)

### 4.2 Variables

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

## 5 Module *hal.internet.engines*

Abstract search engines.

### 5.1 Class *SearchEngineResult*



#### 5.1.1 Methods

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

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

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

```
__str__(self)
```

`str(x)`

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

#### *Inherited from object*

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

#### 5.1.2 Properties

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

### 5.2 Class *SearchEngine*



### 5.2.1 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*)

:param query: string  
 Query to search engine.  
 :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\_\_()

### 5.2.2 Properties

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

## 6 Module *hal.internet.parser*

Parse anything there is on the Internet.

### 6.1 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
```

### 6.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'hal.internet'</code>

### 6.3 Class *HtmlTable*



### 6.3.1 Methods

<b><code>__init__(self, html_source)</code></b>
<p><code>:param html_source: string</code>          Html source of table</p> <p>Overrides: <code>object.__init__</code></p>
<b><code>parse(self)</code></b>
<p><code>:return: list of list</code>          List of list of values in table</p>

#### *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__()`

### 6.3.2 Properties

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

## 7 Module hal.internet.selenium

Some utils methods for a selenium webdriver

### 7.1 Variables

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

### 7.2 Class SeleniumForm

Great and simple static methods to deal with selenium webdrivers.

#### 7.2.1 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.
```

## 8 Module *hal.internet.web*

Deal with webpages.

### 8.1 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
```

### 8.2 Variables

Name	Description
CHROME_USER_AGENT	<b>Value:</b> ["Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWe...
URL_VALID_REGEX	<b>Value:</b> re.compile(r"^(?:http ftp)s?://"r"(?:([A-Z0-9])(?:[A-Z0...

### 8.3 Class Webpage

object  **hal.internet.web.Webpage**  
representation of URL (web page)



### 8.3.1 Methods

<b><code>__init__(self, url, using_tor=False)</code></b>
<hr/>
<code>:param url: string</code> Url of webpage
<code>:param using_tor: bool</code> Whether using tor or not to fetch source page
Overrides: object. <code>__init__</code>

<b><code>parse_url(raw_url)</code></b>
<hr/>
<code>:param raw_url: url to parse</code>
<code>:return: parses correctly url</code>

<b><code>get_scheme(self)</code></b>
<hr/>
<code>:return: get scheme (HTTP, HTTPS, FTP ..) from given url</code>

<b><code>get_hostname(self)</code></b>
<hr/>
<code>:return: extract hostname from given url</code>

<b><code>get_domain(self)</code></b>
<hr/>
<code>:return: get domain from given url</code>

<b><code>get_html_source(self, tor=False)</code></b>
<hr/>
<code>:return: BeautifulSoup to parse</code>

```
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__()
```

### 8.3.2 Properties

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

## 9 Module *hal.internet.youtube*

Get rss feed for youtube channel.

### 9.1 Functions

```
get_channel_page(channel_name,  
youtube_channel_url="https://www.youtube.com/user/")
```

```
@param channel_name: string  
    name of channel.  
@param youtube_channel_url: string  
    base url of youtube channels.  
@return string  
    source page of youtube channel.
```

```
get_channel_id(channel_name,  
channel_id_field="data-channel-external-id")
```

```
@param channel_name: string  
    channel_name name of channel.  
@param channel_id_field: string  
    default field to get channel id.  
@return string  
    id of youtube channel.
```

```
get_channel_feed_url(channel_name,  
base_feed_url="https://www.youtube.com/feeds/videos.xml?channel_id=")
```

```
@param channel_name: string  
    channel_name name of channel.  
@param base_feed_url: string  
    default base url for rss feed of youtube channels.  
@return string  
    rss url feed of youtube channel.
```

## 10 Package *hal.maths*

MATHS: important and scalable math functions

### 10.1 Modules

- **crypt**: Perform fast hash, encryption and calculations related to cryptography.  
(Section 11, p. 29)
- **maths**: A few elegant and powerful mathematical functions.  
(Section 12, p. 41)
- **plotter**: Show elegant plots in any dimension.  
(Section 13, p. 44)

### 10.2 Variables

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

## 11 Module *hal.maths.crypt*

Perform fast hash, encryption and calculations related to cryptography.

### 11.1 Variables

Name	Description
<code>__package__</code>	Value: <code>'hal.maths'</code>

### 11.2 Class MD5

```

object └─
          hal.maths.crypt.MD5
md5 hash
```

#### 11.2.1 Methods

<code>__init__(self, string)</code> <code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> extit(inherited documentation)
<code>hash(self)</code> <hr/> : return: hash plaintext

*Inherited from object*

```

__delattr__(), __format__(), __getattr__(), __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 MD6

```

object └─
          hal.maths.crypt.MD6
md6 hash

```

#### 11.3.1 Methods

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

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

```

#### 11.3.2 Properties

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

### 11.3.3 Class Variables

Name	Description
ALLOWED_SIZE	<b>Value:</b> [64, 128, 224, 256, 384, 512]

## 11.4 Class SHA

object —  
     **hal.maths.crypt.SHA**  
 general SHA hash

### 11.4.1 Methods

**\_\_init\_\_**(*self*, *string*, *size*, *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

<code>hash_sha256(<i>self</i>)</code>
<code>:return: sha256 hash</code>

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

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

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

### *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>	
<code>__class__</code>	

#### 11.4.3 Class Variables

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



## 11.5 Class DES

```

object └─
          hal.maths.crypt.DES

```

DES hash

### 11.5.1 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_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__()

```

### 11.5.2 Properties

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

### 11.5.3 Class Variables

Name	Description
ALLOWED_SIZE	Value: [1, 3]

## 11.6 Class ARC

object └─  
          **hal.maths.crypt.ARC**

ARC hash

### 11.6.1 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_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__()
```

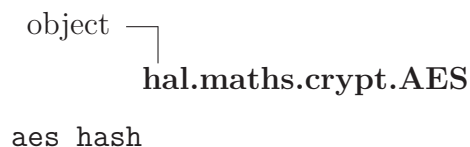
### 11.6.2 Properties

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

### 11.6.3 Class Variables

Name	Description
ALLOWED_SIZE	Value: [2, 4]

## 11.7 Class AES



### 11.7.1 Methods

<b>__init__</b> ( <i>self</i> , <i>string</i> , <i>key</i> )
<code>x.__init__(...)</code> initializes x; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)
<b>hash</b> ( <i>self</i> )
:return: hash plaintext

### *Inherited from object*

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

### 11.7.2 Properties

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

*continued on next page*

Name	Description
<code>__class__</code>	

## 11.8 Class HMAC

object —  
     **hal.maths.crypt.HMAC**  
 hmac hash

### 11.8.1 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__()`

### 11.8.2 Properties

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

## 11.9 Class BLOWFISH

object —  
     **hal.maths.crypt.BLOWFISH**

blowfish hash

### 11.9.1 Methods

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

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

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

```
hash(self)
```

:return: hash plaintext

#### *Inherited from object*

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

### 11.9.2 Properties

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

## 11.10 Class IDEA

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

IDEA hash

### 11.10.1 Methods

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

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

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

<b>hash</b> ( <i>self</i> )
:return: IDEA hash

<b>change_key</b> ( <i>self</i> , <i>key</i> )
:param key: new key
:return: change key

<b>encrypt</b> ( <i>self</i> )
:return: encrypt with key

***Inherited from object***

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

**11.10.2 Properties**

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

**11.11 Class CAST128**

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

CAST 128 hash

**11.11.1 Methods**

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

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

```
encrypt(self)
```

```
decrypt(self)
```

***Inherited from object***

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),  

__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),  

__str__(), __subclasshook__()
```

**11.11.2 Properties**

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

**11.12 Class Dsa**

```

object ┌
      │ hal.maths.crypt.Dsa

```

dsa hash

**11.12.1 Methods**

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

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

```
hash(self)
```

```
:return: hash plaintext
```

### *Inherited from object*

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

#### 11.12.2 Properties

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



## 12 Module *hal.maths.maths*

A few elegant and powerful mathematical functions.

### 12.1 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
```

### 12.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'hal.maths'</code>

### 12.3 Class Integer

```
object └─
          hal.maths.maths.Integer
```

#### 12.3.1 Methods

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

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

#### 12.3.2 Properties

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

#### 12.3.3 Class Variables

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

## 12.4 Class EightQueen

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

8 queen problem solver

## 12.4.1 Methods

```
__init__(self, board_size)
```

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

```
under_attack(self, col, queens)
```

```
solve(self, n)
```

*Inherited from object*

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

## 12.4.2 Properties

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

## 13 Module *hal.maths.plotter*

Show elegant plots in any dimension.

### 13.1 Class *Plot2d*

object —  
     *hal.maths.plotter.Plot2d*

2d plot

#### 13.1.1 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*

```

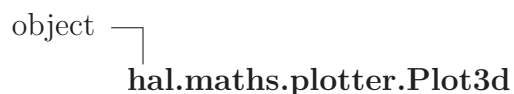
__delattr__(), __format__(), __getattr__(), __hash__(), __init__(),
__new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(),
__sizeof__(), __str__(), __subclasshook__()

```

### 13.1.2 Properties

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

## 13.2 Class Plot3d



### 13.2.1 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__()
```

#### 13.2.2 Properties

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

## 13.3 Class Plot4d

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

#### 13.3.1 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__()
```

### 13.3.2 Properties

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

## 14 Package hal.ml

### 14.1 Modules

- **data** (*Section 15, p. 49*)
  - **parser**: Parsers for raw databases.  
(*Section 16, p. 50*)
- **features**: Collection of methods to find weights of features and select the best ones.  
(*Section 17, p. 52*)
- **models** (*Section 18, p. 53*)
  - **classification**: Prediction methods based on classification algorithms.  
(*Section 19, p. 54*)
  - **pipelined**: Prediction methods based on multiple models mixed up.  
(*Section 20, p. 55*)
  - **regression**: Prediction methods based on regression algorithms.  
(*Section 21, p. 56*)
  - **time\_series**: Multi-purpose prediction methods to be used in time-series.  
(*Section 22, p. 57*)
- **predict**: " General model to make prediction about everything.  
(*Section 23, p. 58*)
- **utils**: Various tools and utilities to deal with database and machine learning.  
(*Section 24, p. 59*)

### 14.2 Variables

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



## 15 Package *hal.ml.data*

### 15.1 Modules

- **parser**: Parsers for raw databases.  
(Section 16, p. 50)

### 15.2 Variables

Name	Description
__package__	Value: None

## 16 Module *hal.ml.data.parser*

Parsers for raw databases.

### 16.1 Variables

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

### 16.2 Class Parser

object └─ **hal.ml.data.parser.Parser**

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

#### 16.2.1 Methods

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

*Inherited from object*

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

#### 16.2.2 Properties

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

## 16.3 Class CSVParser



### 16.3.1 Methods

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

<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 16.2)*

`get_lines()`

*Inherited from object*

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

### 16.3.2 Properties

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

## 17 Module *hal.ml.features*

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

### 17.1 Functions

<code>select_k_best(<i>x</i>, <i>y</i>, <i>k</i>)</code>
--

select k best features in dataset
-----------------------------------

<code>get_best_features(<i>x</i>, <i>y</i>)</code>
--

finds the optimal number of features
--------------------------------------

<code>get_features(<i>x</i>, <i>y</i>, <i>n_features_to_select</i>)</code>
--

finds the optimal features
----------------------------

### 17.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'hal.ml'</code>

## 18 Package *hal.ml.models*

### 18.1 Modules

- **classification**: Prediction methods based on classification algorithms.  
(Section 19, p. 54)
- **pipelined**: Prediction methods based on multiple models mixed up.  
(Section 20, p. 55)
- **regression**: Prediction methods based on regression algorithms.  
(Section 21, p. 56)
- **time\_series**: Multi-purpose prediction methods to be used in time-series.  
(Section 22, p. 57)

### 18.2 Variables

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

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

Prediction methods based on classification algorithms.

### 19.1 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>
--------------------------------

### 19.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> <code>'hal.ml.models'</code>

## 20 Module *hal.ml.models.pipelined*

Prediction methods based on multiple models mixed up.

### 20.1 Functions

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

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

### 20.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> <code>'hal.ml.models'</code>

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

Prediction methods based on regression algorithms.

### 21.1 Functions

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

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

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

### 21.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> <code>'hal.ml.models'</code>



## 22 Module *hal.ml.models.time\_series*

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

### 22.1 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

## 23 Module *hal.ml.predict*

" General model to make prediction about everything.

### 23.1 Variables

Name	Description
<code>__package__</code>	Value: <code>'hal.ml'</code>

### 23.2 Class *BasePrediction*

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

#### 23.2.1 Methods

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

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

Overrides: object.**`__init__`** `exit`(inherited documentation)

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

#### *Inherited from object*

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

#### 23.2.2 Properties

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

## 24 Module *hal.ml.utils*

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

### 24.1 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

**show\_correlation\_matrix**(*feature\_list, correlation\_matrix*)

Show the given correlation matrix as image

## 25 Package hal.profile

### 25.1 Modules

- **performance:** Perform benchmarks and tests on your PC.  
(Section 26, p. 62)

### 25.2 Variables

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

## 26 Module *hal.profile.performance*

Perform benchmarks and tests on your PC.

### 26.1 Variables

Name	Description
<code>__package__</code>	Value: <code>'hal.profile'</code>

### 26.2 Class *EightQueenTest*

object  ***hal.profile.performance.EightQueenTest***

test CPU by solving eight-queen problem

#### 26.2.1 Methods

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

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

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

**`welcome()`**

:return: introduce script

**`introduction()`**

:return: introduce 8 queen problem

**`run(self)`**

*Inherited from object*

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

**26.2.2 Properties**

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

## 27 Package *hal.wrappers*

### 27.1 Modules

- **methods:** Typical (and useful) function wrappers  
(*Section 28, p. 65*)

### 27.2 Variables

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



## 28 Module *hal.wrappers.methods*

Typical (and useful) function wrappers

### 28.1 Functions

<code>handle_exceptions(<i>function</i>)</code>
<code>:param function: callback function</code> function to wrap
<code>:return: callback function return type</code> wraps callback function

## Index

- hal (*package*), 5–6
  - hal.files (*package*), 7
    - hal.files.models (*module*), 8–16
  - hal.internet (*package*), 17
    - hal.internet.engines (*module*), 18–19
    - hal.internet.parser (*module*), 20–21
    - hal.internet.selenium (*module*), 22–23
    - hal.internet.web (*module*), 24–26
    - hal.internet.youtube (*module*), 27
  - hal.maths (*package*), 28
    - hal.maths.crypt (*module*), 29–40
    - hal.maths.maths (*module*), 41–43
    - hal.maths.plotter (*module*), 44–47
  - hal.ml (*package*), 48
    - hal.ml.data (*package*), 49
    - hal.ml.features (*module*), 52
    - hal.ml.models (*package*), 53
    - hal.ml.predict (*module*), 58
    - hal.ml.utils (*module*), 59–60
  - hal.profile (*package*), 61
    - hal.profile.performance (*module*), 62–63
  - hal.wrappers (*package*), 64
    - hal.wrappers.methods (*module*), 65