

---

# **pydons Documentation**

***Release 0.2.3***

**Jakub Urban**

June 30, 2015



<b>1</b>	<b>MatStruct class</b>	<b>3</b>
<b>2</b>	<b>LazyDataset class</b>	<b>5</b>
<b>3</b>	<b>FileBrowser class</b>	<b>7</b>
<b>4</b>	<b>Examples</b>	<b>9</b>
<b>5</b>	<b>Reference</b>	<b>11</b>
<b>6</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



Pydons is a collection of manipulation add-ons for hierarchichal numerical data.



---

## **MatStruct class**

---

MatStruct is an ordered dict with string-only keys, which are accessible also as properties. This makes the notation easier (`obj.group.subgroup.variable` instead of `obj['group']['subgroup']['variable']`) and enables IPython's auto complete.

MatStruct can be serialized to HDF5 or Matlab files using the excellent [hdf5storage](#) package.





---

## LazyDataset class

---

A lazy evaluate proxy class for data sets in HDF5 or netCDF4 files.



---

## FileBrowser class

---

FileBrowser employs MatStruct and LazyDataset to enable easy and fast browsing of netCDF4 or HDF5 files.



---

## Examples

---

Items can be added using either ['keys'] or .properties:

```
import pydons
import numpy as np
struct = pydons.MatStruct()
struct['string'] = 'A string'
struct.list = ['list', 0, [1, 2]]
struct.numpy = np.random.rand(3,3)
```

IPython representation is customized:

```
In [12]: struct
Out[12]:
string: A string
  list: ['list', 0, [1, 2]]
  numpy: [[ 0.71539338  0.69970494  0.19328026]
 [ 0.28645949  0.15262059  0.23362895]
 [ 0.14518748  0.79911631  0.22522526]]
```

MatStruct can be serialized to HDF5 or Matlab files using saveh5 and savemat methods:

```
In [15]: struct.saveh5('struct.h5')
In [16]: pydons.MatStruct.loadh5('struct.h5')
Out[16]:
  list: ['list', 0, [1, 2]]
  numpy: [[ 0.71539338  0.69970494  0.19328026]
 [ 0.28645949  0.15262059  0.23362895]
 [ 0.14518748  0.79911631  0.22522526]]
string: A string
```

(the field order is not maintained—to be fixed soon).

This software is distributed under the MIT license (see the LICENSE file).



---

## Reference

---

Pydons is a collection of manipulation add-ons for hierarchical numerical data.

```
class pydons.FileBrowser (file_name, file_type=None, squeeze=False, transpose=None,  

lazy_min_size=10, lazy_max_size=10000000, any_keys=False)
```

Read hierarchical data file into a MatStruct tree with data in LazyDataset

### Parameters

- **file\_name** – file name
- **file\_type** – file type, default (None) for autodetect
- **lazy\_min\_size** – data sets with a lower size will be always stored in the memory
- **lazy\_max\_size** – data sets with a larger size will never be stored in the memory

```
class pydons.LazyDataset (grp, name, squeeze=False, transpose=False, lazy_min_size=10,  

lazy_max_size=100000000)
```

NetCDF 4 / HDF5 data set object with lazy evaluation

```
class pydons.MatStruct (values=(), dedict=True, any_keys=False)
```

Matlab-like struct container

Features:

- Get and set fields as properties (`obj.field =` )
- String-only fields
- Save and load to/from Matlab-compatible HDF5 files
- Ipython customized output

### Parameters

- **values** – list/tuple of key, value pairs or a dict-like object
- **dedict** – convert dict members to MatStruct
- **any\_keys** – allow arbitrary keys, not only strings

```
diff (other, **kwargs)
```

Find numerical differences to another MatStruct, ignoring the keys order

Returns a structure with `diff_norm` = average norm of all numerical differences, `diff_max` = maximum of norm differences, `diff_uncomparable` = number of uncomparable fields.

**Parameters** **other** – MatStruct object to compare to

Keyword arguments

**Parameters**

- **norm** – norm function, default is `numpy.linalg.norm`
- **rel\_norm\_thold** – relative difference threshold above which relative difference is normalized by the norm of the field value

**insert\_after** (*existing\_key, key, value*)

Insert after an existing field

**Parameters**

- **existing\_key** – existing key
- **key** – new key
- **value** – inserted value

**insert\_before** (*existing\_key, key, value*)

Insert before an existing field

**Parameters**

- **existing\_key** – existing key
- **key** – new key
- **value** – inserted value

**classmethod loadh5** (*file\_name, path='/', matlab\_compatible=False, \*\*kwargs*)

Load from an HDF5 file

**Parameters**

- **file\_name** – file name
- **path** – path to read data from

**classmethod loadmat** (*file\_name, path='/', \*\*kwargs*)

Load from a Matlab (HDF5 format) file

**Parameters**

- **file\_name** – file name
- **path** – path to read data from

**merge** (*other*)

Merge fields from another MatStruct or any dict-like object

**Parameters** **other** – object to merge from

**saveh5** (*file\_name, path='/', truncate\_existing=False, matlab\_compatible=False, \*\*kwargs*)

Save to an HDF5 file

**Parameters**

- **file\_name** – output file name
- **path** – group path to store fields to

**savemat** (*file\_name, path='/', truncate\_existing=False, \*\*kwargs*)

Save to a Matlab (HDF5 format) file

**Parameters**

- **file\_name** – output file name



- **path** – group path to store fields to



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## p

pydons, [11](#)



## D

`diff()` (pydons.MatStruct method), [11](#)

## F

`FileBrowser` (class in pydons), [11](#)

## I

`insert_after()` (pydons.MatStruct method), [12](#)

`insert_before()` (pydons.MatStruct method), [12](#)

## L

`LazyDataset` (class in pydons), [11](#)

`loadh5()` (pydons.MatStruct class method), [12](#)

`loadmat()` (pydons.MatStruct class method), [12](#)

## M

`MatStruct` (class in pydons), [11](#)

`merge()` (pydons.MatStruct method), [12](#)

## P

`pydons` (module), [11](#)

## S

`saveh5()` (pydons.MatStruct method), [12](#)

`savemat()` (pydons.MatStruct method), [12](#)