

---

# **html\_nested\_tables Documentation**

***Release 0.2.0***

**Bertrand Bordage**

**Sep 27, 2017**



---

## Contents

---

**Python Module Index**

**3**



Source available on [GitHub](#).

**class** `html_nested_tables.base.TableDict (*args, **kws)`

TableDict objects are ordered dicts with methods that renders HTML tables.

Here is an ugly schema to define the terms I'm using:

	hh1		hh2		hh3		hh4	
	hh11	hh12	hh21	hh22	hh31	hh32	hh41	hh42
vh1	dA	dB	dC	dE	dF	dG	dH	dI
vh2	dJ	dK	dL	dM	dN	dO	dP	dQ
vh3	dR	dS	dT	dU	dV	dW	dX	dY

*hh1*, *hh11*, *hh2* [...] are horizontal headers, *vh1*, *vh2*, [...] are vertical headers, and *dA*, *dB*, [...] are data.

**generate\_html** ()

Generates an HTML table from the contents of `self`.

It creates an empty cell first, adds horizontal headers, then adds both vertical headers and data in the same time.

This whole thing could be implemented using BeautifulSoup and its outstanding inplace modification possibilities. That may be much more readable, but also much slower.

**Returns** A HTML table.

**Return type** unicode

**get\_ugliness** ()

Returns the ugliness of the current table.

The uglier a table is, the less readable it becomes.

**Returns** The ugliness of the current table.

**Return type** int

**class** `html_nested_tables.base.HorizontalTableDict (*args, **kws)`

Same as `TableDict`, but with a direction.

The direction is used to specify whether the headers of the first depth of the current object should be put on the horizontal or vertical axis.

**class** `html_nested_tables.base.VerticalTableDict (*args, **kws)`

Same as `TableDict`, but with a direction.

The direction is used to specify whether the headers of the first depth of the current object should be put on the horizontal or vertical axis.

`html_nested_tables.base.h`  
alias of `HorizontalTableDict`

`html_nested_tables.base.v`  
alias of `VerticalTableDict`

`html_nested_tables.base.get_all_structures (datadict)`

Returns all the possible structures for a datadict.

**Parameters** `datadict` – Nested dicts or association lists. Association lists have the advantage of being ordered.

**Returns** All possible structures.

**Return type** list

`html_nested_tables.base.build_table_dict(datadict, structure)`

Automatically builds a `TableDict` from `datadict` and `structure`.

**Parameters**

- **datadict** (*dict or tuple or list*) – Nested dicts or association lists. Association lists have the advantage of being ordered.
- **structure** (*list or tuple*) – Structure of the headers of the returned object. This must be a sequence of `h` and/or `v`, one per depth level of `datadict`.

**Returns** Nested `TableDict` s with horizontal and/or vertical structures applied, according to `structure`.

**Return type** `HorizontalTableDict` or `VerticalTableDict`

`html_nested_tables.base.build_optimal_table_dict(datadict)`

Automatically builds the less ugly table possible from `datadict`.

**Parameters** **datadict** (*dict or tuple or list*) – Nested dicts or association lists. Association lists have the advantage of being ordered.

**Returns** Nested `TableDict` with horizontal and/or vertical structures applied, according to `structure`.

**Return type** `HorizontalTableDict` or `VerticalTableDict`

## h

`html_nested_tables.base`, [1](#)





## B

`build_optimal_table_dict()` (in module `html_nested_tables.base`), 2  
`build_table_dict()` (in module `html_nested_tables.base`), 1

## G

`generate_html()` (`html_nested_tables.base.TableDict` method), 1  
`get_all_structures()` (in module `html_nested_tables.base`), 1  
`get_ugliness()` (`html_nested_tables.base.TableDict` method), 1

## H

`h` (in module `html_nested_tables.base`), 1  
`HorizontalTableDict` (class in `html_nested_tables.base`), 1  
`html_nested_tables.base` (module), 1

## T

`TableDict` (class in `html_nested_tables.base`), 1

## V

`v` (in module `html_nested_tables.base`), 1  
`VerticalTableDict` (class in `html_nested_tables.base`), 1