PaperSize Documentation

Release 0.1.2

Louis Paternault

June 10, 2015

Contents

Paper size related data and functions.

This module provides tools to manipulate paper sizes, that is:

- a dictionary of several named standard names (e.g. A4, letter) , with their respective sizes (with and height);
- functions to convert sizes between units;
- functions to manipulate paper orientation (portrait or landscape);
- tools to parse paper sizes, so that you do not have to worry about the format of paper sizes provided by your user, it being *a4* or *21cm x 29.7cm*.

CHAPTER 1

Download and install

See the main project page for instructions.

CHAPTER 2

Module documentation

Paper size related data and functions

In this module:

- the default unit (input and output) is point (`pt`);
- every numbers are returned as `decimal.Decimal` objects.

2.1 Constants

`papersize.`**`UNITS`**
Dictionary of units.

Keys are unit abbreviation (e.g. `pt` or `cm`), and values are their value in points (e.g. `UNITS['pt']` is 1, `UNITS['pc']` is 12), as `decimal.Decimal` objects.

`papersize.`**`SIZES`**
Dictionary of named sizes.

Keys are names (e.g. `a4`, `letter`) and values are strings, human-readable, and parsable by *`parse_papersize()`* (e.g. `21cm x 29.7cm`).

`papersize.`**`PORTRAIT`**
Constant corresponding to the portrait orientation

That is, height greater than width.

`papersize.`**`LANDSCAPE`**
Constant corresponding to the landscape orientation

That is, width greater than height.

2.2 Unit conversion

`papersize.`**`convert_length`**(*length*, *orig*, *dest*)
Convert length from one unit to another.

Parameters

- **`length`** (*decimal.Decimal*) – Length to convert, as any object convertible to a `decimal.Decimal`.
- **`orig`** (*str*) – Unit of `length`, as a string which is a key of *`UNITS`*.

- **dest** (*str*) – Unit in which `length` will be converted, as a string which is a key of *UNITS*.

Due to floating point arithmetic, there can be small rounding errors.

```
>>> convert_length(0.1, "cm", "mm")
Decimal('1.000000000000000055511151231')
```

2.3 Parsers

papersize.**parse_length**(*string*, *unit='pt'*)

Return a length corresponding to the string.

Parameters

- **string** (*str*) – The string to parse, as a length and a unit, for instance `10.2cm`.
- **unit** (*str*) – The unit of the return value, as a key of *UNITS*.

Returns The length, in an unit given by the `unit` argument.

Return type `decimal.Decimal`

```
>>> parse_length("1cm", "mm")
Decimal('1E+1')
>>> parse_length("1cm", "cm")
Decimal('1')
>>> parse_length("10cm")
Decimal('284.52755910')
```

papersize.**parse_couple**(*string*, *unit='pt'*)

Return a tuple of dimensions.

Parameters **string** (*str*) – The string to parse, as "LENGTHxLENGTH" (where LENGTH are length, parsable by *parse_length()*). Example: `21cm x 29.7cm`. The separator can be `x`, `×` or empty, surrounded by an arbitrary number of spaces. For instance: `2cmx3cm`, `2cm x 3cm`, `2cm×3cm`, `2cm 3cm`.

Return type `tuple`

Returns A tuple of `decimal.Decimal`, representing the dimensions.

```
>>> parse_couple("1cm 10cm", "mm")
(Decimal('1E+1'), Decimal('1.0E+2'))
>>> parse_couple("1mm 10mm", "cm")
(Decimal('0.1'), Decimal('1.0'))
```

papersize.**parse_papersize**(*string*, *unit='pt'*)

Return the papersize corresponding to string.

Parameters

- **string** (*str*) – The string to parse. It can be either a named size (as keys of constant *SIZES*), or a couple of lengths (that will be processed by *parse_couple()*). The named paper sizes are case insensitive. The following strings return the same size: `a4`, `A4`, `21cm 29.7cm`, `210mmx297mm`, `21cm × 297mm`...
- **unit** (*str*) – The unit of the return values.

Returns The paper size, as a couple of `decimal.Decimal`.

Return type `tuple`

```
>>> parse_papersize("A4", "cm")
(Decimal('21.0'), Decimal('29.7'))
>>> parse_papersize("21cm x 29.7cm", "mm")
(Decimal('2.1E+2'), Decimal('297'))
>>> parse_papersize("10 100")
(Decimal('10'), Decimal('100'))
```

2.4 Paper orientation

papersize.**is_portrait**(*width*, *height*)

Return whether paper orientation is portrait

That is, height greater or equal to width.

Parameters

- **width** – Width of paper, as any sortable object.
- **height** – Height of paper, as any sortable object.

```
>>> is_portrait(11, 10)
False
>>> is_portrait(10, 10)
True
>>> is_portrait(10, 11)
True
```

papersize.**is_landscape**(*width*, *height*)

Return whether paper orientation is landscape

That is, width greater or equal to height.

Parameters

- **width** – Width of paper, as any sortable object.
- **height** – Height of paper, as any sortable object.

```
>>> is_landscape(11, 10)
True
>>> is_landscape(10, 10)
True
>>> is_landscape(10, 11)
False
```

papersize.**is_square**(*width*, *height*)

Return whether paper is a square (width equals height).

Parameters

- **width** – Width of paper, as any sortable object.
- **height** – Height of paper, as any sortable object.

```
>>> is_square(11, 10)
False
>>> is_square(10, 10)
True
>>> is_square(10, 11)
False
```

papersize.**rotate**(*size*, *orientation*)
Return the size, rotated if necessary to make it portrait or landscape.

Parameters

- **size** (*tuple*) – Couple paper of dimension, as sortable objects (`int`, `float`, `decimal.Decimal`...).
- **orientation** – Return format, one of `PORTRAIT` or `LANDSCAPE`.

Returns The size, as a couple of dimensions, of the same type of the `size` parameter.

Return type `tuple`

```
>>> rotate((21, 29.7), PORTRAIT)
(21, 29.7)
>>> rotate((21, 29.7), LANDSCAPE)
(29.7, 21)
```

2.5 Exceptions

class papersize.**PapersizeException**
All exceptions of this module inherit from this one.

class papersize.**CouldNotParse**(*string*)
Raised when a string could not be parsed.

Parameters **string** (*str*) – String that could not be parsed.

class papersize.**UnknownOrientation**(*string*)
Raised when a string could not be parsed.

Parameters **string** (*obj*) – Object wrongly provided as an orientation.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

p

C

I

L

P

R

S

U