
pypyt Documentation

Release 0.1.1

Julio Gajardo

Oct 08, 2019

1	Tutorial	3
1.1	How to install it	3
1.2	How to render content	3
2	How-To Guides	7
2.1	How to render paragraphs	8
2.2	How to render a table	10
2.3	How to render charts	15
2.4	How to render a picture	17
2.5	How to render a whole presentation	19
3	Library Documentation	21
3.1	Template Functions	22
3.2	Presentation Functions	23
3.3	Shape Functions	25
3.4	Render Functions	27
3.5	Utility Functions	31
3.6	Documentation Functions	31
	Python Module Index	33
	Index	35

Pypyt is a library to render PowerPoint presentations from python in an easy and intuitive way.

The code is open source, and available on [GitHub](#)

1.1 How to install it

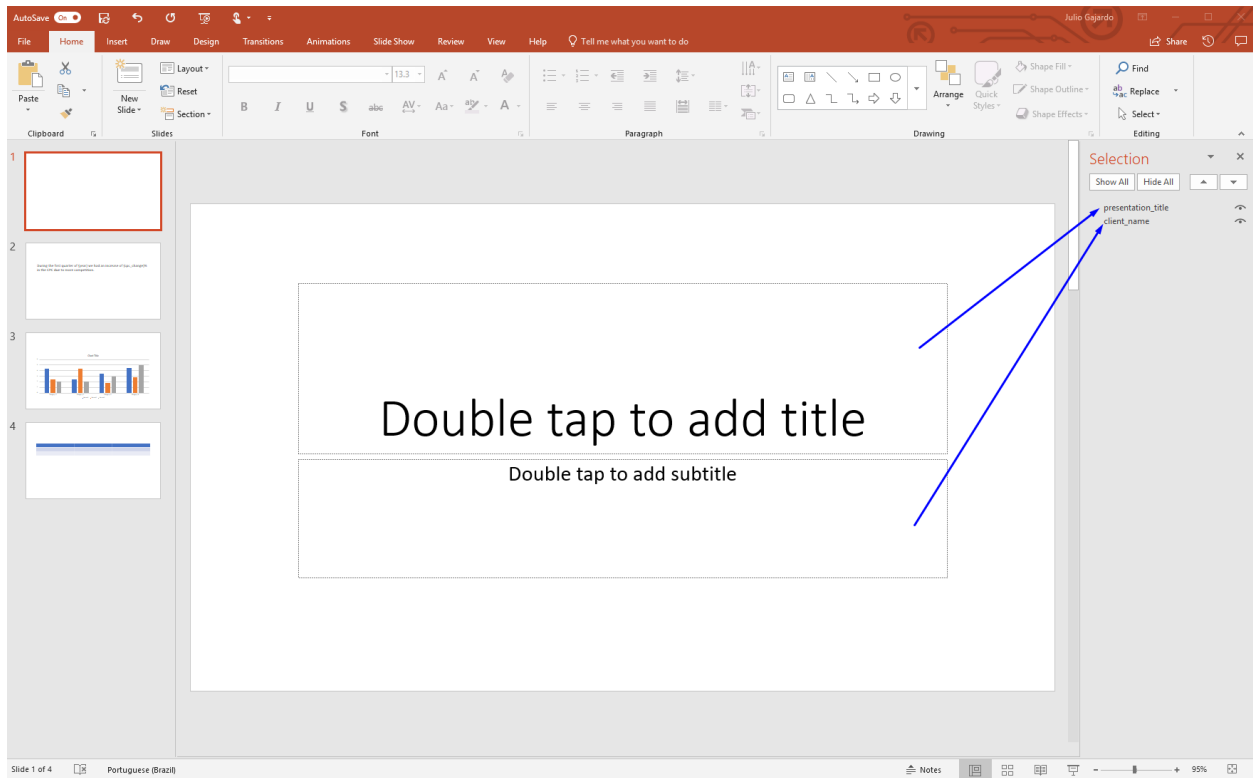
To install it, you just need to run the *pip* command:

```
pip install ppyt
```

1.2 How to render content

First of all, you need to create a template file with the objects names as shown in [this video](#)

Lets assume that you have a template file named `__template__.pptx` with two shapes: `presentation_title` and `client_name` as shown in the image below.



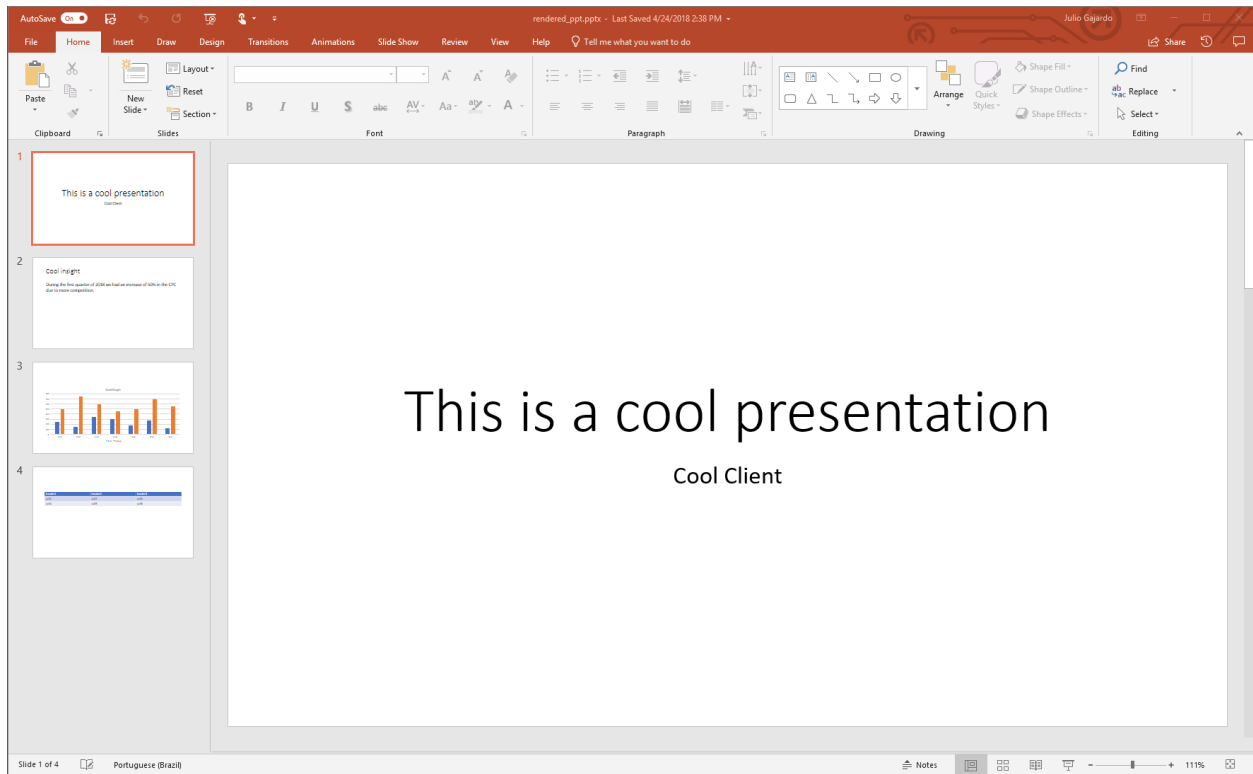
In order to render it you might use the following code:

```
from pypyt import render_and_save_template

values = {
    'presentation_title': "This is a cool presentation",
    'client_name': "Cool Client"
}

render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

This will render a presentation like the one below.



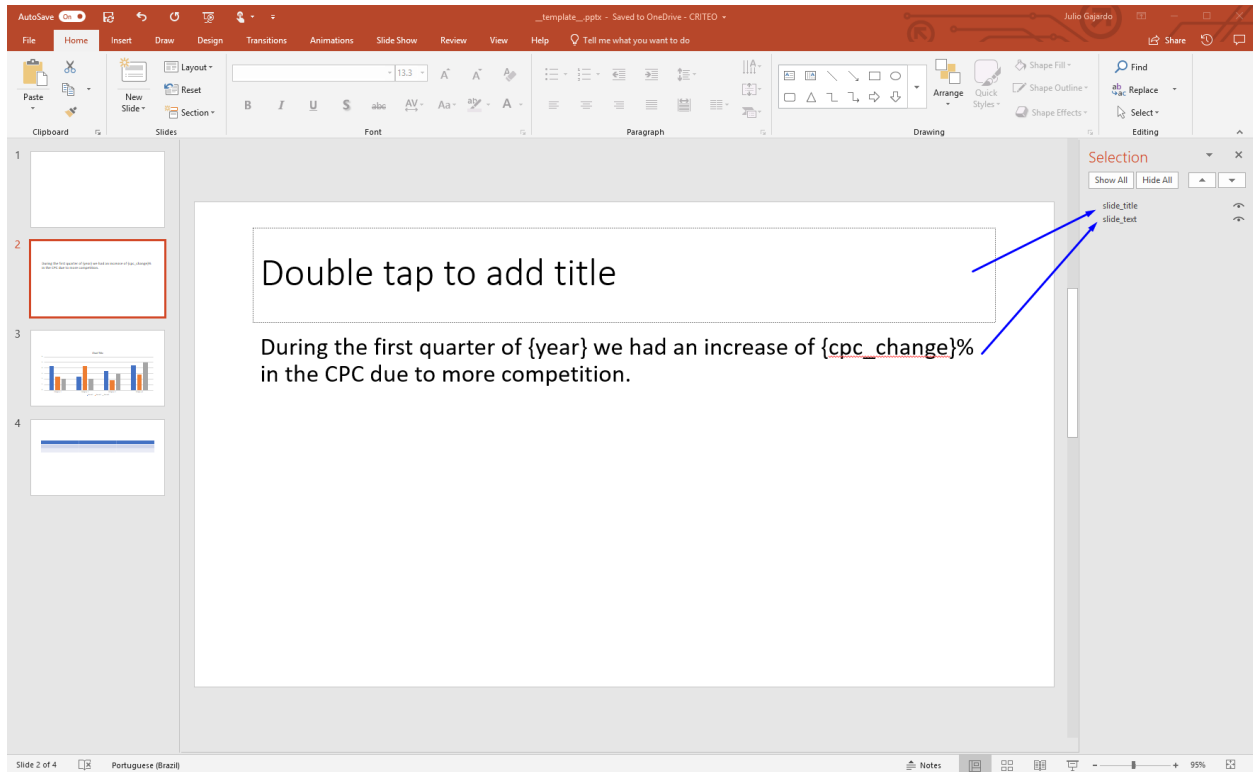
For more information about how to use this library, see the [How-To Guides](#).

For the full documentation, see the [Library Documentation](#).

- ***How to render paragraphs***
 - *How to replace the whole text of a paragraph*
 - *How to render placeholders within a paragraph*
- ***How to render a table***
 - *How to render a table from a python list*
 - *How to render a table from a pandas DataFrame*
 - *How to render a table from a pandas DataFrame using the header*
- ***How to render charts***
 - *How to render a chart from a python dictionary*
 - *How to render a chart from a pandas DataFrame*
- ***How to render a picture***
 - *How to render a picture using a file name*
 - *How to render a picture using a file-like object*
- *How to render a whole presentation*

2.1 How to render paragraphs

2.1.1 Template



2.1.2 How to replace the whole text of a paragraph

If you want to replace the whole text of the shape, you just need to define a dictionary, with the name of the shape as the key, and the text you want to render as the value.

Code:

```
>>> values = {
...     'slide_title': "Cool insight",
... }

>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

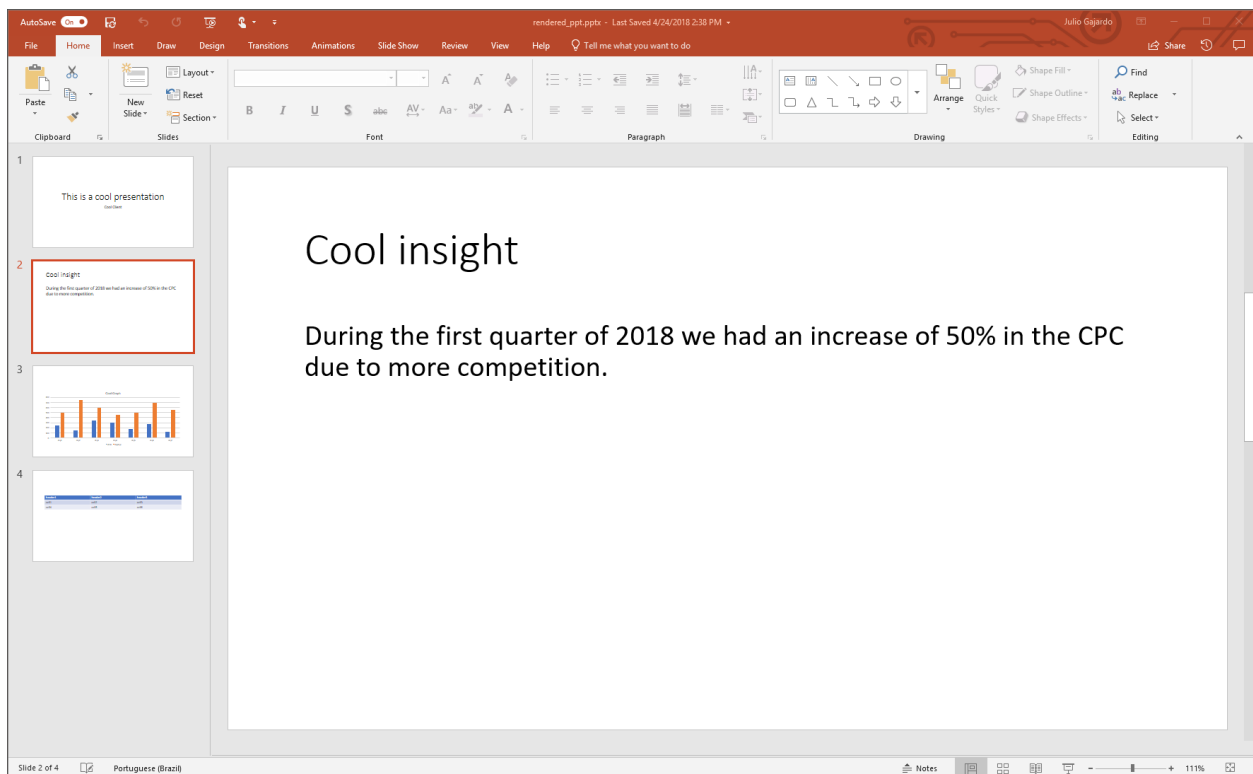
2.1.3 How to render placeholders within a paragraph

If you want to render placeholders within the paragraph, the value of the dictionary has to be another dictionary, with the placeholders as keys, and the text to be replaced as values. **The placeholder name has to be a valid python variable name.**

Code:

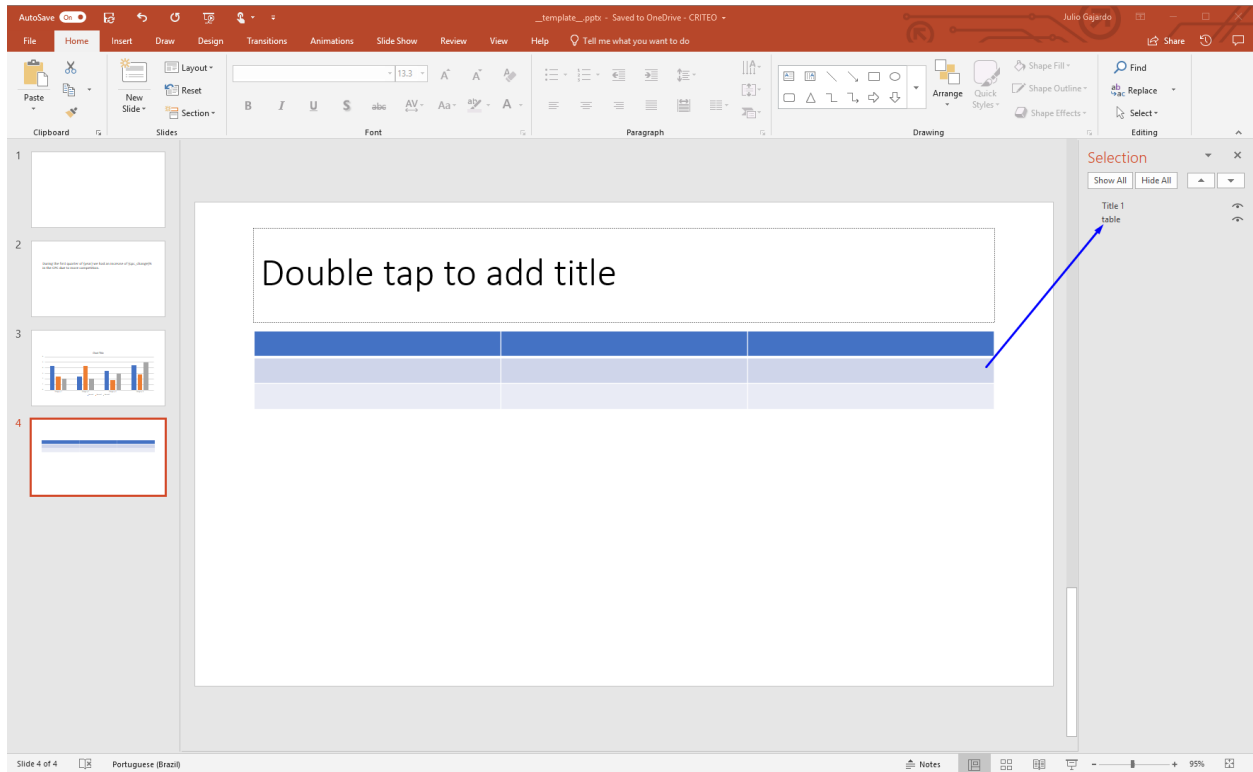
```
>>> values = {  
...     'slide_text': {  
...         'year': 2018,  
...         'cpc_change': 50  
...     }  
... }  
  
>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

2.1.4 Output



2.2 How to render a table

2.2.1 Template



2.2.2 How to render a table from a python list

In order to render a table, just pass the value of each cell as a list of lists.

Code:

```
>>> values = {
...     'table': [
...         ['header1', 'header2', 'header3'],
...         ['cell11', 'cell12', 'cell13'],
...         ['cell14', 'cell15', 'cell16']
...     ]
... }

>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

2.2.3 How to render a table from a pandas DataFrame

In the case you want to render the values of a DataFrame in the table, just pass the DataFrame instead of the list of lists.

Code:

```

>>> data = [
...     ['header', 'header2', 'header3'],
...     ['cell11', 'cell12', 'cell13'],
...     ['cell14', 'cell15', 'cell16']
... ]

>>> table_df = pd.DataFrame(data)
>>> table_df

   col1    col2    col3
0  header1  header2  header3
1   cell11   cell12   cell13
2   cell14   cell15   cell16

>>> values = {'table': table_df}
>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')

```

2.2.4 How to render a table from a pandas DataFrame using the header

If you want to use the DataFrame's column names as header, you should set the `header` attribute to `True`

Code:

```

>>> data = [
...     ['cell11', 'cell12', 'cell13'],
...     ['cell14', 'cell15', 'cell16']
... ]

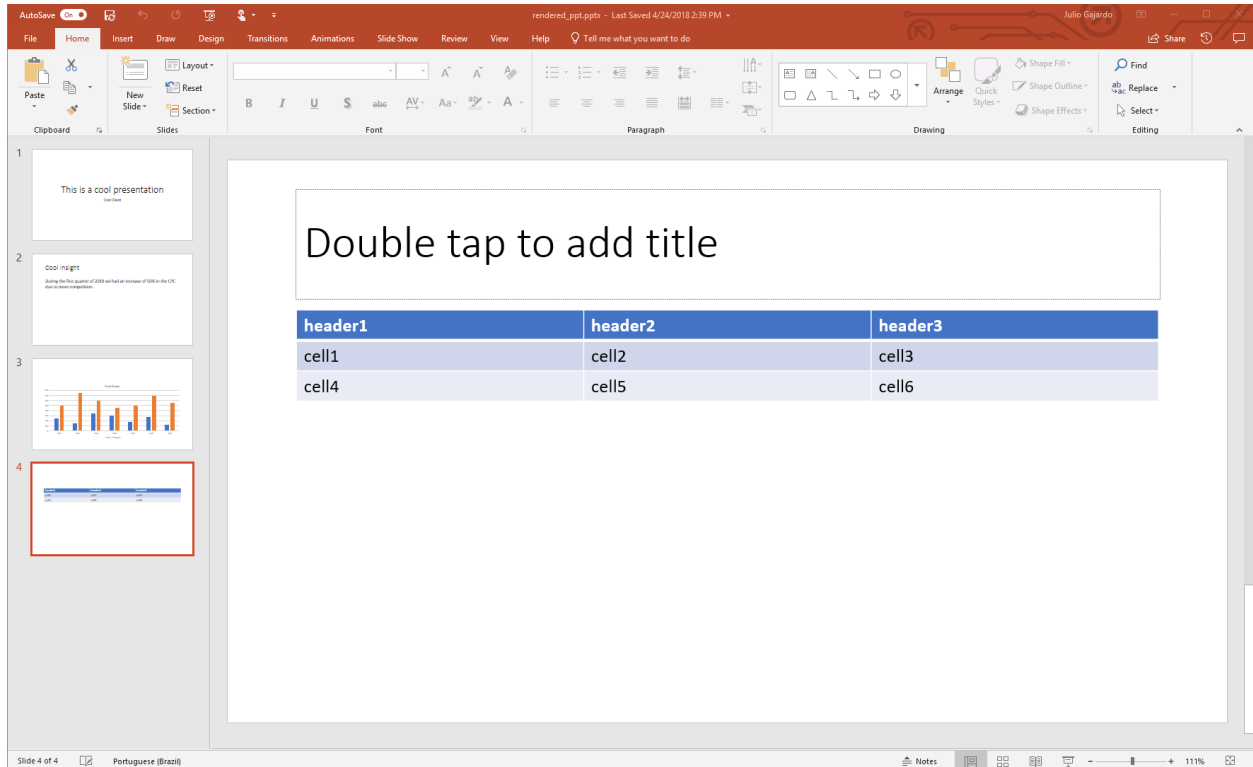
>>> table_df = pd.DataFrame(data, columns=['header', 'header2', 'header3'])
>>> table_df

   header1  header2  header3
0   cell11   cell12   cell13
1   cell14   cell15   cell16

>>> table_df.header = True
>>> values = {'table': table_df}
>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')

```

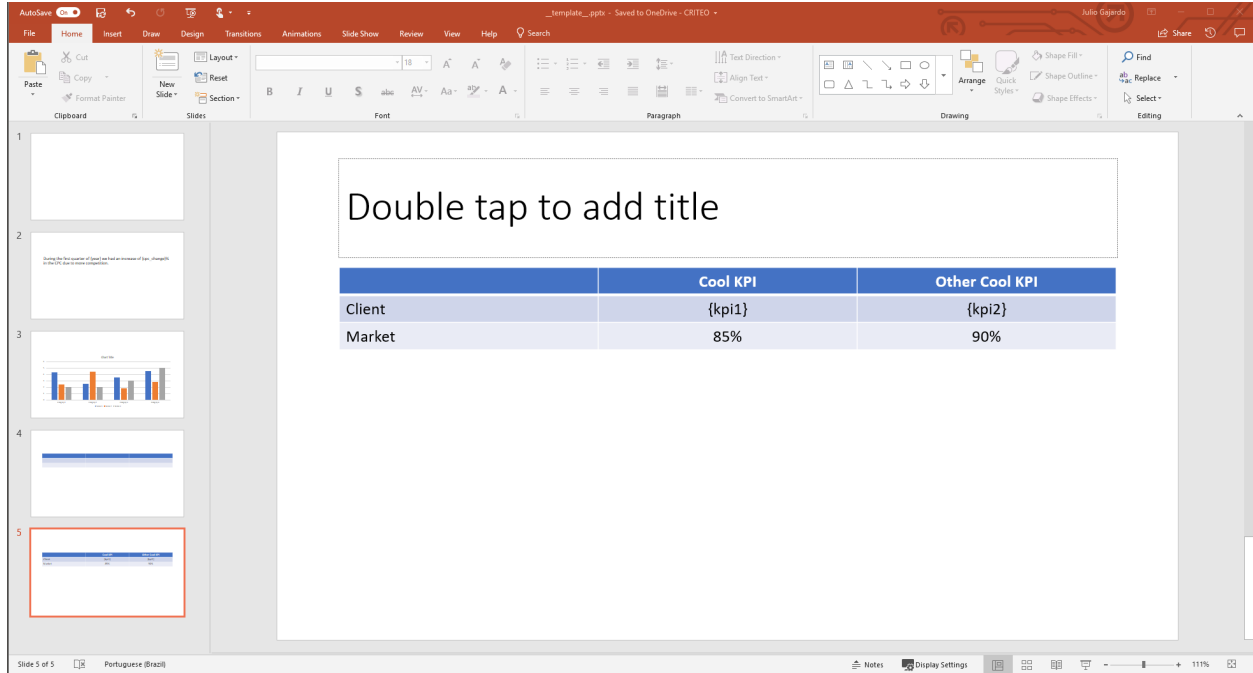
2.2.5 Output



2.2.6 How to render parts of a table using placeholders and a python dictionary

If you want to render partially a table, for example with some KPI's values, you can use a python dictionary.

2.2.7 Template

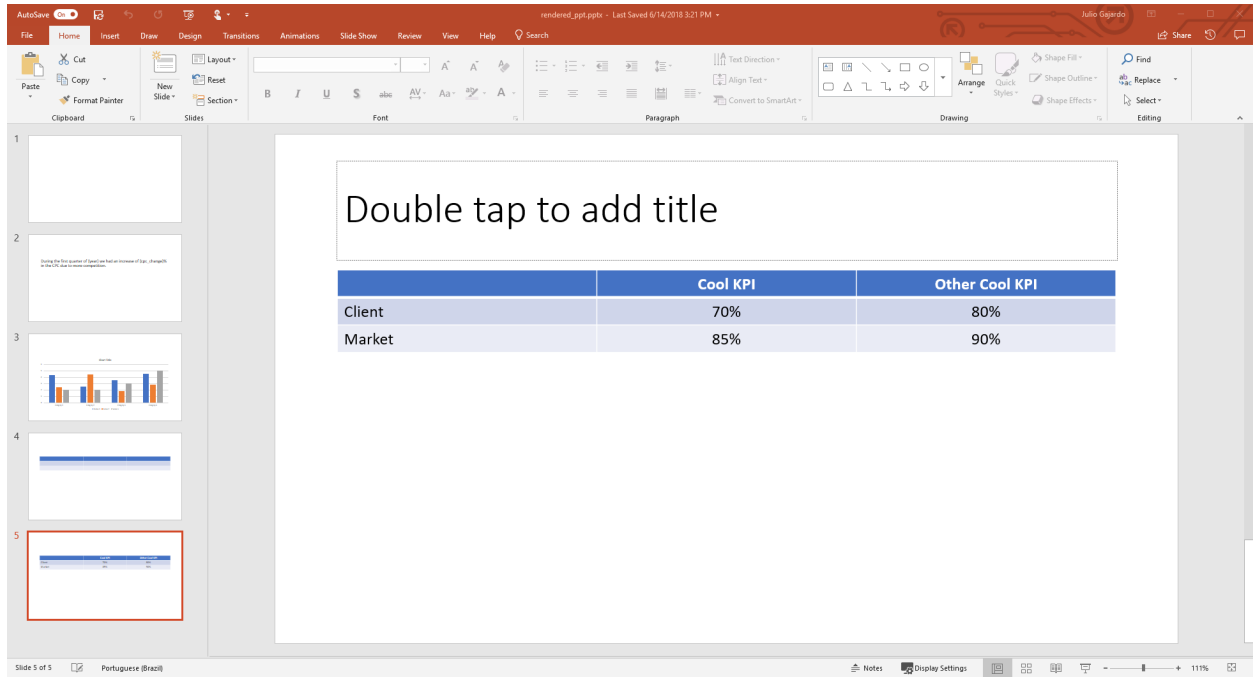


Code:

```
>>> values = {
...     'table': {
...         'kpi1': "70%",
...         'kpi2': "80%"
...     }
... }

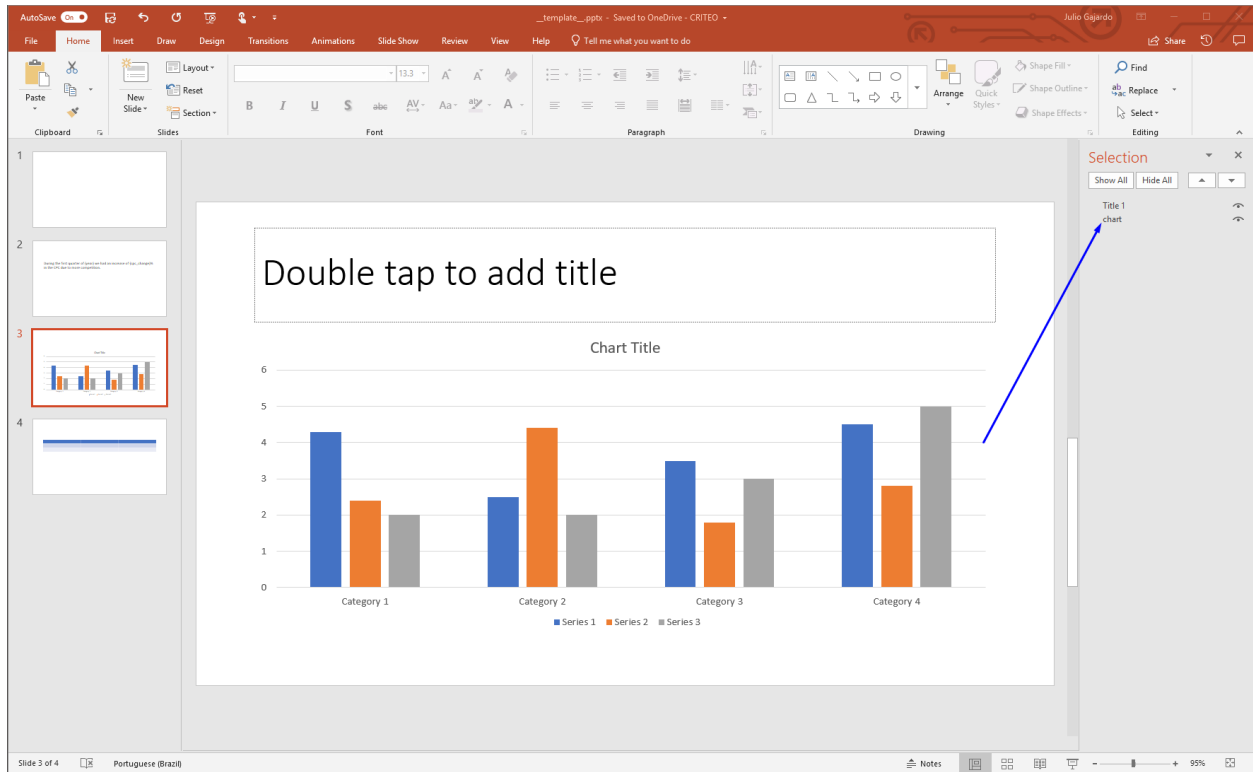
>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

2.2.8 Output



2.3 How to render charts

2.3.1 Template



2.3.2 How to render a chart from a python dictionary

In order to render a chart, you need to give a dictionary with three elements, the title, the data and the categories. The data is itself another dictionary, with the name of the series as the key and values of the series as the value. Code:

```
>>> values = {
...     'chart': {
...         'title': "Cool Graph",
...         'data': {
...             'displays': [500, 750, 600, 450, 500, 700, 550],
...             'clicks': [250, 150, 350, 300, 175, 275, 125]
...         },
...         'categories': ['day1', 'day2', 'day3', 'day4', 'day5', 'day6', 'day7']
...     }
... }

>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

2.3.3 How to render a chart from a pandas DataFrame

You can also build a chart from a DataFrame, the column names will be used as the series names, the values as it values and the index as the categories, if you want to set the title, you should add the attribute `title` to the DataFrame.

Code:

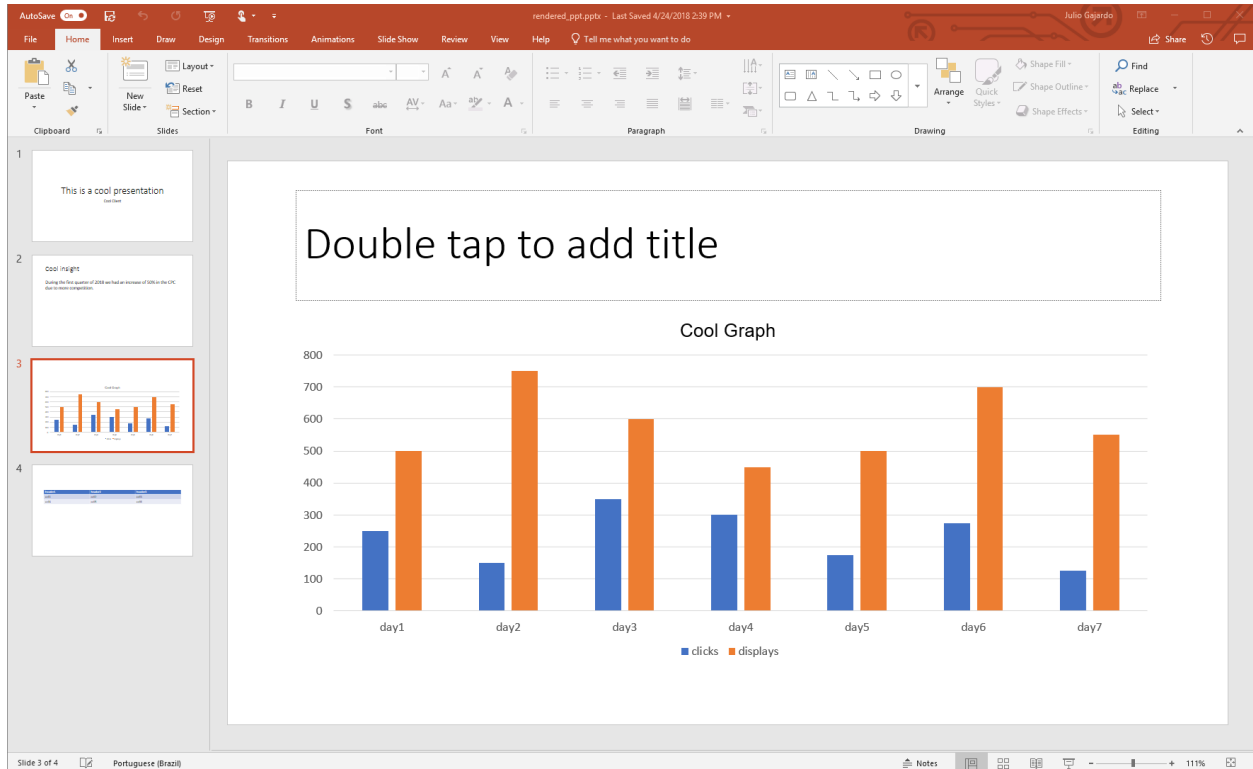
```
>>> data = [
...     [250, 500],
...     [150, 750],
...     [350, 600],
...     [300, 450],
...     [175, 500],
...     [275, 700],
...     [125, 550],
... ]

>>> pd_chart = pd.DataFrame(data,
...                          index=['day1', 'day2', 'day3', 'day4', 'day5', 'day6',
...                                ↪ 'day7'],
...                          columns=['clicks', 'displays'])

>>> pd_chart
   clicks  displays
0     250        500
1     150        750
2     350        600
3     300        500
4     175        500
5     275        700
6     125        550

>>> pd_chart.title = "Cool Graph"
>>> values = {
...     'chart': pd_chart
... }
>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

2.3.4 Output

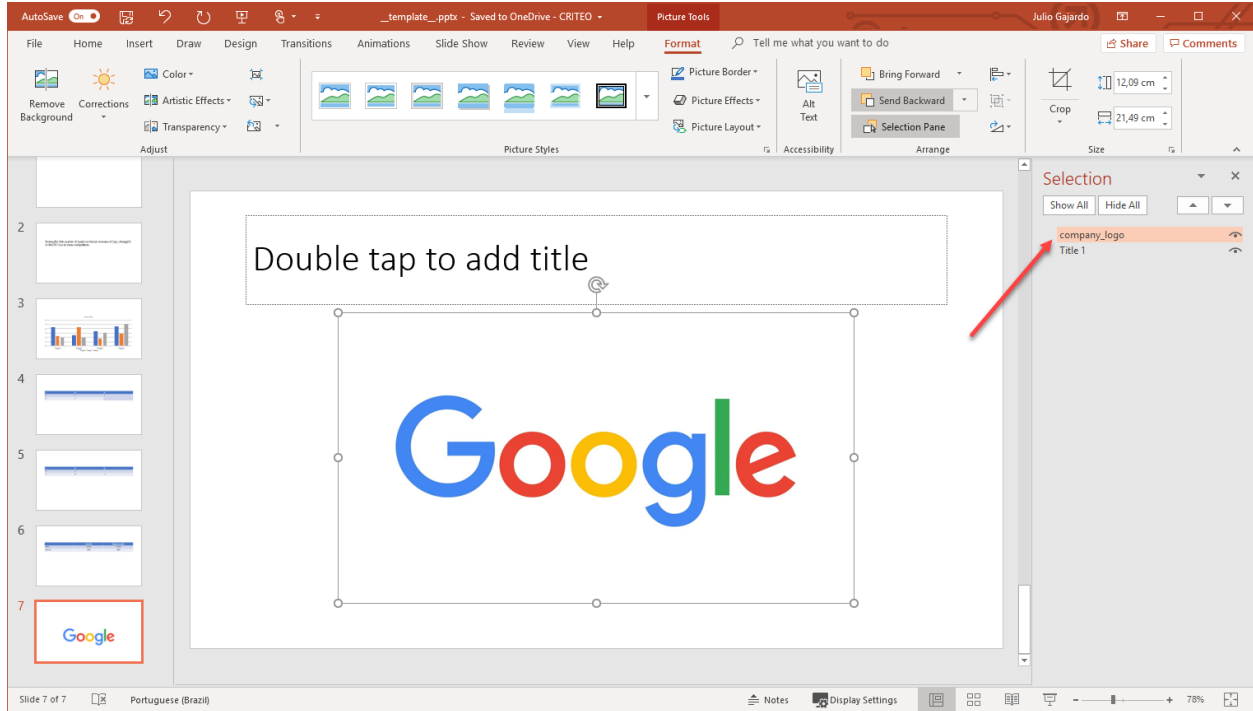


2.3.5 How to format a piechart to display percentages

2.4 How to render a picture

To render a picture, you should add one to your template, this image will be replaced by the selected one as described below:

2.4.1 Template



2.4.2 How to render a picture using a file name

To render from a filename, you just need to give the name of the image file to replace as a string.

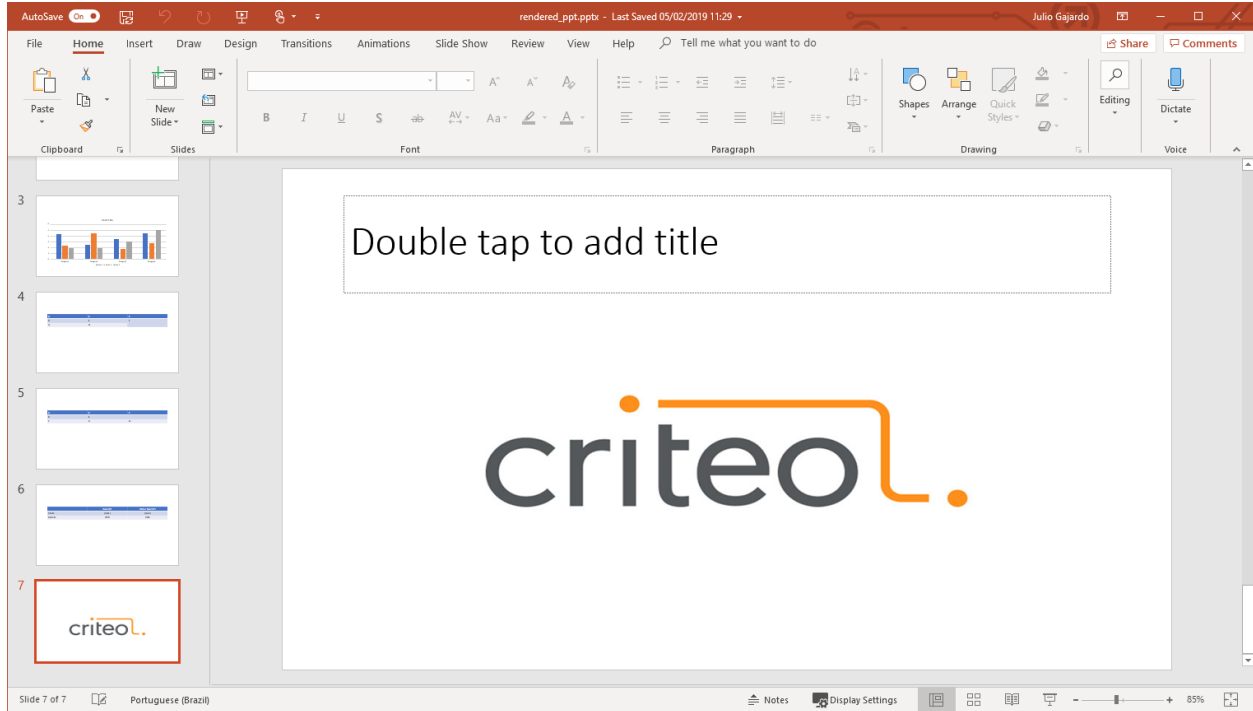
```
>>> values = {
...     'company_logo': 'criteo_logo.png'
... }
>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

2.4.3 How to render a picture using a file-like object

To render from a file-like object, you just need to pass the object.

```
>>> import io
>>> with open('criteo_logo.png', 'br') as file:
>>>     picture = io.BytesIO(file.read())
>>> values = {
...     'company_logo': picture
... }
>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```

2.4.4 Output



2.5 How to render a whole presentation

In case you're wondering, you don't need to render one shape at the time. If you have a template like the previous one, you can render all the shapes with the following code:

```
>>> values = {
...     'slide_title': "Cool insight",
...     'slide_text': {
...         'year': 2018,
...         'cpc_change': 50
...     }
...     'table': [
...         ['header1', 'header2', 'header3'],
...         ['cell11', 'cell12', 'cell13'],
...         ['cell14', 'cell15', 'cell16']
...     ]
...     'chart': {
...         'title': "Cool Graph",
...         'data': {
...             'displays': [500, 750, 600, 450, 500, 700, 550],
...             'clicks': [250, 150, 350, 300, 175, 275, 125]
...         },
...         'categories': ['day1', 'day2', 'day3', 'day4', 'day5', 'day6', 'day7']
...     }
... }

>>> render_and_save_template('__template__.pptx', values, 'rendered_ppt.pptx')
```


For normal use, all you will have to do to get started is:

```
from pypyt import *
```

This will import the following functions:

Template Functions

- The `open_template()` function opens a template.
- The `render_template()` function renders a template.
- The `render_and_save_template()` function renders and saves a template.

Presentation Functions:

- The `get_shapes()` function gets all the shapes and its type in a presentation.
- The `get_shapes_by_name()` function gets all the shapes with the given name in a presentation.
- The `render_ppt()` function renders a presentation.
- The `render_and_save_ppt()` function renders and save a presentation.
- The `save_ppt()` function saves a presentation.

Shape Functions:

- The `get_shape_type()` function gets the type of a shape.
- The `is_chart()` function returns `True` if the given shape is a chart.
- The `is_hyperlink()` function returns `True` if the given shape is a hyperlink.
- The `is_paragraph()` function returns `True` if the given shape is a paragraph.
- The `is_picture()` function returns `True` if the given shape is a picture.
- The `is_table()` function returns `True` if the given shape is a table.

Render Functions:

- The `render_chart()` function renders a chart object.
- The `render_hyperlink()` function renders a hyperlink object.
- The `render_paragraph()` function renders a paragraph object.
- The `render_picture()` function renders a picture object.
- The `render_table()` function renders a table object.

Utility Functions:

- The `picture_from_url()` function gets an image from the given url.

Documentation Functions:

- The `pypyt_doc()` function opens this documentation.

3.1 Template Functions

`pypyt.open_template(template_name)`

Opens a pptx file given the `template_name` and returns it.

Parameters `template_name` (*str*) – The name of the file to be open.

Returns Presentation object

Return type pptx.presentation.Presentation

Examples

Open a ppt template for later use

```
>>> open_template('template.pptx')
<pptx.presentation.Presentation at ...>
```

`pypyt.render_template(template_name, values)`

Returns a rendered presentation given the template name and values to be rendered.

Parameters

- **template_name** (*str*) – Name of the presentation to be rendered.
- **values** (*dict*) – Dictionary with the values to render on the template.
- **raise_error** (*bool*) – Boolean that indicates whether an error has to be raised or not when is not possible to render a shape

Returns Rendered presentation

Return type pptx.presentation.Presentation

Examples

Render a template.

```
>>> values = {'presentation_title': "My Cool Presentation"}
>>> render_template('template.pptx', values)
<pptx.presentation.Presentation at ...>
```

`pypyt.render_and_save_template(template_name, values, filename)`

Renders and save a presentation with the given template name, values to be rendered, and filename.

Parameters

- **template_name** (*str*) – Name of the presentation to be saved.
- **values** (*dict*) – Dictionary with the values to render on the template.
- **raise_error** (*bool*) – Boolean that indicates whether an error has to be raised or not when is not possible to render a shape
- **filename** (*str*) – Name of the file to be saved.

Examples

Render and save a template

```
>>> values = {'presentation_title': "My Cool Presentation"}
>>> render_and_save_template('template.pptx', values, 'presentation.pptx')
```

3.2 Presentation Functions

`pypyt.get_shapes(prs)`

Returns a dictionary to be filled with the values to be rendered in the presentation.

Parameters

- **prs** (*pptx.presentation.Presentation*) – Presentation to get the shapes from.
- **get_all** (*bool*) – If False, filters out the shapes that are believed to be created automatically

Returns Dictionary with the shape names as keys and an empty data structure to fill the values.

Return type dict

Examples

Get all the shapes in a presentation.

```
>>> prs = open_template('template.pptx')
>>> get_shapes(prs)
{'chart': {'categories': [], 'data': [], 'title': ''},
 'client_name': '',
 'presentation_title': '',
 'slide_text': {'cpc_change': '', 'year': ''},
 'slide_title': '',
 'table': [[None, None, None], [None, None, None], [None, None, None]]}
```

```
>>> get_shapes(prs, all=True)
{'Title 1': '',
 'chart': {'categories': [], 'data': [], 'title': ''},
 'client_name': '',
 'presentation_title': '',
```

(continues on next page)

(continued from previous page)

```
'slide_text': {'cpc_change': '', 'year': ''},
'slide_title': '',
'table': [[None, None, None], [None, None, None], [None, None, None]]}
```

`pypyt.get_shapes_by_name` (*prs, name*)

Returns a list of shapes with the given name in the given presentation.

Parameters

- **prs** (*pptx.presentation.Presentation*) – Presentation to be saved.
- **name** (*str*) – Name of the shape(s) to be returned.

Examples

Get all the shapes named ‘chart’ in a presentation.

```
>>> prs = open_template('template.pptx')
>>> get_shapes_by_name(prs, 'chart')
[<pptx.shapes.placeholder.PlaceholderGraphicFrame at ...>]
```

`pypyt.render_ppt` (*prs, values*)

Returns a rendered presentation given the template name and values to be rendered.

Parameters

- **prs** (*pptx.presentation.Presentation*) – Presentation to be rendered.
- **values** (*dict*) – Dictionary with the values to render on the template.
- **raise_error** (*bool*) – Boolean that indicates whether an error has to be raised or not when is not possible to render a shape

Returns Rendered presentation

Return type *pptx.presentation.Presentation*

Examples

Open a template and render it.

```
>>> prs = open_template('template.pptx')
>>> values = {'presentation_title': "My Cool Presentation"}
>>> render_ppt(prs, values)
<pptx.presentation.Presentation at ...>
```

`pypyt.render_and_save_ppt` (*template_name, values, filename*)

Renders and save a presentation with the given template name, values to be rendered, and filename.

Parameters

- **prs** (*Presentation*) – Name of the presentation to be saved.
- **values** (*dict*) – Dictionary with the values to render on the template.
- **filename** (*str*) – Name of the file to be saved.
- **raise_error** (*bool*) – Boolean that indicates whether an error has to be raised or not when is not possible to render a shape

Examples

Open a template, render it and save it.

```
>>> values = {'presentation_title': "My Cool Presentation"}
>>> prs = open_template('template.pptx')
>>> render_and_save_ppt(prs, values, 'presentation.pptx')
```

`pypyt.save_ppt(prs, filename)`

Saves the given presentation with the given filename.

Parameters

- **prs** (`pptx.presentation.Presentation`) – Presentation to be saved.
- **filename** (`str`) – Name of the file to be saved.

Examples

Render a template and save it.

```
>>> values = {'presentation_title': "My Cool Presentation"}
>>> rendered_prs = render_template('template.pptx', values)
>>> save_ppt(rendered_prs, 'presentation.pptx')
```

3.3 Shape Functions

`pypyt.get_shape_type(shape)`

Returns a string with the kind of the given shape.

Parameters **shape** (`pptx.shapes.BaseShape`) – Shape to get the type from.

Returns String representing the type of the shape

Return type string

Examples

Get the type of a shape.

```
>>> prs = open_template('template.pptx')
>>> shapes = get_shapes_by_name(prs, 'client_name')
>>> get_shape_type(shapes[0])
'paragraph'
```

`pypyt.is_chart(shape)`

Checks whether the given shape is a chart.

Parameters **shape** (`pptx.shapes.base.BaseShape`) – Shape to get whether is a chart or not.

Returns Boolean representing whether the given shape is a table or no.

Return type bool

Examples

Check whether the given shape is a table.

```
>>> prs = open_template('template.pptx')
>>> shapes = get_shapes_by_name(prs, 'client_name')
>>> is_chart(shapes[0])
True
```

`pypyt.is_hyperlink(shape)`

Checks whether the given shape is a hyperlink

Parameters **shape** (`pptx.shapes.base.BaseShape`) – Shape to get whether is a hyperlink or not.

Returns Boolean representing whether the given shape is a hyperlink or not.

Return type bool

Examples

Check whether the given shape is a hyperlink.

```
>>> prs = open_template('template.pptx')
>>> shapes = get_shapes_by_name(prs, 'client_name')
>>> is_hyperlink(shapes[0])
False
```

`pypyt.is_paragraph(shape)`

Checks whether the given shape is a paragraph.

Parameters **shape** (`pptx.shapes.base.BaseShape`) – Shape to get whether is a paragraph or not.

Returns Boolean representing whether the given shape is a table or no.

Return type bool

Examples

Check whether the given shape is a table.

```
>>> prs = open_template('template.pptx')
>>> shapes = get_shapes_by_name(prs, 'client_name')
>>> is_paragraph(shapes[0])
True
```

`pypyt.is_picture(shape)`

Checks whether the given shape is a picture.

Parameters **shape** (`pptx.shapes.base.BaseShape`) – Shape to get whether is a picture or not.

Returns Boolean representing whether the given shape is a picture or not.

Return type bool

Examples

Check whether the given shape is a picture.

```
>>> prs = open_template('template.pptx')
>>> shapes = get_shapes_by_name(prs, 'client_name')
>>> is_picture(shapes[0])
False
```

`pypyt.is_table(shape)`

Checks whether the given shape is a table.

Parameters **shape** (*pptx.shapes.base.BaseShape*) – Shape to get whether is a table or not.

Returns Boolean representing whether the given shape is a table or no.

Return type bool

Examples

Check whether the given shape is a table.

```
>>> prs = open_template('template.pptx')
>>> shapes = get_shapes_by_name(prs, 'client_name')
>>> is_table(shapes[0])
False
```

3.4 Render Functions

`pypyt.render_chart(values, chart)`

Renders the given values into the given chart.

Parameters

- **values** (*dict* or *pandas.DataFrame*) – Values to render the chart.
- **chart** (*pptx.chart.chart.Chart*) – Chart object to be rendered.

Examples

Render a chart from a dictionary

```
>>> prs = open_template('template.pptx')
>>> chart_values = {
...     'title': "My Cool Graph",
...     'categories': ['d1', 'd2', 'd3'],
...     'data': {
...         'displays': [500, 750, 600],
...         'clicks': [150, 250, 200]
...     }
... }
>>> shapes = get_shapes_by_name(prs, 'chart')
```

(continues on next page)

(continued from previous page)

```
>>> shape = shapes[0]
>>> render_chart(chart_values, shape.chart)
```

Render a chart for a pandas DataFrame

```
>>> prs = open_template('template.pptx')
>>> data = [
...     [250, 500],
...     [150, 750],
...     [350, 600],
...     [300, 450],
...     [175, 500],
...     [275, 700],
...     [125, 550],
... ]
>>> pd_chart = pd.DataFrame(data,
...                          index=['day1', 'day2', 'day3', 'day4', 'day5', 'day6',
...                                ↪ 'day7'],
...                          columns=['clicks', 'displays'])
>>> pd_chart
   clicks  displays
0      250        500
1      150        750
2      350        600
3      300        500
4      175        500
5      275        700
6      125        550
>>> pd_chart.title = "Cool Graph"
>>> shapes = get_shapes_by_name(prs, 'chart')
>>> shape = shapes[0]
>>> render_chart(pd_chart, shape.chart)
```

`pypyt.render_hyperlink(values, hyperlink)`

Renders the given values into the given hyperlink.

Parameters

- **values** (*str*) – Address of the link.
- **hyperlink** – Hyperlink object to be rendered.

Examples

Render a hyperlink

```
>>> prs = open_template('template.pptx')
>>> hyperlink_value = {
...     'https://pypyt.readthedocs.io'
... }
>>> shapes = get_shapes_by_name(prs, 'hyperlink')
>>> shape = shapes[0]
>>> render_chart(hyperlink_value, shape)
```

`pypyt.render_paragraph(values, text_frame)`

In the case you want to replace the whole text.

Parameters

- **values** (*dict, str, int or float*) – Values to render the text.
- **text_frame** (*pptx.text.text.TextFrame*) – TextFrame object to be rendered.

Examples

Replace full text.

```
>>> prs = open_template('template.pptx')
>>> paragraph = {
...     'slide_title': "Cool insight",
... }
>>> shapes = get_shapes_by_name(prs, 'slide_title')
>>> shape = shapes[0]
>>> render_table(paragraph, shape.text_frame)
```

Replace placeholders within text.

```
>>> prs = open_template('template.pptx')
>>> paragraph = {
...     'slide_text': {
...         'year': 2018,
...         'cpc_change': 50
...     }
... }
>>> shapes = get_shapes_by_name(prs, 'slide_text')
>>> shape = shapes[0]
>>> render_table(paragraph, shape.text_frame)
```

`pypyt.render_picture(values, table)`

Renders an image with the given values.

Parameters

- **values** (*string (filename) or file-like object*) – Reference to the image
- **image** (*pptx.shapes.picture.Picture*) – Shape where to render the image

Examples

Render image from image filename

```
>>> prs = open_template('template.pptx')
>>> picture = 'image_file.jpg'
>>> shapes = get_shapes_by_name(prs, 'image')
>>> shape = shapes[0]
>>> render_picture(picture, shape)
```

Render image from file-like object

```
>>> import io
>>> prs = open_template('template.pptx')
>>> with open('image_file.jpg', 'br') as file:
>>>     picture = io.BytesIO(file.read())
```

(continues on next page)

(continued from previous page)

```
>>> shapes = get_shapes_by_name(prs, 'image')
>>> shape = shapes[0]
>>> render_picture(picture, shape)
```

pypyt.**render_table**(*values*, *table*)

Renders a table with the given values.

Parameters

- **values** (*dict*, *list* or *pandas.DataFrame*) – Values to render the table
- **table** (*pptx.shapes.table.Table*) – Table object to be rendered.

Examples

Render table from python list

```
>>> prs = open_template('template.pptx')
>>> table: [
...     ['header1', 'header2', 'header3'],
...     ['cell11', 'cell12', 'cell13'],
...     ['cell14', 'cell15', 'cell16']
... ]
>>> shapes = get_shapes_by_name(prs, 'table')
>>> shape = shapes[0]
>>> render_table(table, shape.table)
```

Render table from pandas DataFrame without header

```
>>> prs = open_template('template.pptx')
>>> data = [
...     ['header', 'header2', 'header3'],
...     ['cell11', 'cell12', 'cell13'],
...     ['cell14', 'cell15', 'cell16']
... ]
>>> table_df = pd.DataFrame(data)
>>> table_df
   col1  col2  col3
0  header1 header2 header3
1   cell11  cell12  cell13
2   cell14  cell15  cell16
>>> shapes = get_shapes_by_name(prs, 'table')
>>> shape = shapes[0]
>>> render_chart(table_df, shape.table)
```

Render a table from a pandas DataFrame with header.

```
>>> prs = open_template('template.pptx')
>>> data = [
...     ['cell11', 'cell12', 'cell13'],
...     ['cell14', 'cell15', 'cell16']
... ]
>>> table_df = pd.DataFrame(data, columns=['header', 'header2', 'header3'])
>>> table_df
   header1 header2 header3
0   cell11  cell12  cell13
```

(continues on next page)

(continued from previous page)

```
1  cell4    cell5    cell6
>>> table_df.header = True
>>> shapes = get_shapes_by_name(prs, 'table')
>>> shape = shapes[0]
>>> render_chart(table_df, shape.table)
```

3.5 Utility Functions

`pypyt.picture_from_url(url)`

Loads an image from a given url.

Parameters `url` (*string*) – Url from the image

Returns Byte object with the image.

Return type `io.BytesIO`

Examples

Render image from image url

```
>>> prs = open_template('template.pptx')
>>> url = 'http://yoursite.com/image_file.jpg'
>>> shapes = get_shapes_by_name(prs, 'image')
>>> shape = shapes[0]
>>> render_picture(picture_from_url(url), shape)
```

3.6 Documentation Functions

`pypyt.pypyt_doc()`

Opens pypyt's documentation in the browser.

p

`pypyt`, [21](#)

G

`get_shape_type()` (in module `pypyt`), 25
`get_shapes()` (in module `pypyt`), 23
`get_shapes_by_name()` (in module `pypyt`), 24

I

`is_chart()` (in module `pypyt`), 25
`is_hyperlink()` (in module `pypyt`), 26
`is_paragraph()` (in module `pypyt`), 26
`is_picture()` (in module `pypyt`), 26
`is_table()` (in module `pypyt`), 27

O

`open_template()` (in module `pypyt`), 22

P

`picture_from_url()` (in module `pypyt`), 31
`pypyt` (module), 21
`pypyt_doc()` (in module `pypyt`), 31

R

`render_and_save_ppt()` (in module `pypyt`), 24
`render_and_save_template()` (in module `pypyt`), 22
`render_chart()` (in module `pypyt`), 27
`render_hyperlink()` (in module `pypyt`), 28
`render_paragraph()` (in module `pypyt`), 28
`render_picture()` (in module `pypyt`), 29
`render_ppt()` (in module `pypyt`), 24
`render_table()` (in module `pypyt`), 30
`render_template()` (in module `pypyt`), 22

S

`save_ppt()` (in module `pypyt`), 25