

---

# sphinx-plotly-directive

harupy

Feb 27, 2021



**CONTENTS:**

<b>1</b>	<b>Examples</b>	<b>1</b>
1.1	basic . . . . .	1
1.2	doctest . . . . .	1
1.3	function . . . . .	2
1.4	fig-vars . . . . .	2
1.5	precode . . . . .	3
1.6	iframe-size . . . . .	3
1.7	show . . . . .	4
<b>2</b>	<b>Indices and tables</b>	<b>5</b>



**EXAMPLES****1.1 basic**

Source:

```
.. plotly::  
  
    import plotly.express as px  
  
    px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

Output:

```
import plotly.express as px  
  
px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

**1.2 doctest**

Source:

```
.. plotly::  
  
    >>> import plotly.express as px  
    >>> px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

Output:

```
>>> import plotly.express as px  
>>> px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

## 1.3 function

test\_func.py

```
import plotly.express as px

def func():
    return px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

Source:

```
.. plotly:: examples/test_func.py func
```

Output:

```
import plotly.express as px

def func():
    return px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

## 1.4 fig-vars

### 1.4.1 Single

With *fig-vars* option, you can render a plotly figure assigned in a variable.

Source:

```
.. plotly::
   :fig-vars: fig1

   import plotly.express as px

   fig1 = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

Output:

```
import plotly.express as px

fig1 = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

### 1.4.2 Multiple

You can specify multiple variables.

Source:

```
.. plotly::
   :fig-vars: fig1, fig2

   import plotly.express as px
```

(continues on next page)

(continued from previous page)

```
fig1 = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
fig2 = px.scatter(x=[4, 3, 2, 1, 0], y=[0, 1, 4, 9, 16])
```

Output:

```
import plotly.express as px

fig1 = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
fig2 = px.scatter(x=[4, 3, 2, 1, 0], y=[0, 1, 4, 9, 16])
```

## 1.5 precode

By default, the following code will be executed before running each code block. This allows to use `np`, `plotly`, `go`, and `px` without importing them.

```
import numpy as np
import plotly
import plotly.graph_objects as go
import plotly.express as px
```

Source:

```
.. plotly::
    :fig-vars: fig1, fig2

    x = np.arange(5)
    y = x ** 2

    title = "plotly version: {}".format(plotly.__version__)
    fig1 = go.Figure(go.Scatter(x=x, y=y), layout=dict(title=title))
    fig2 = px.scatter(x=x, y=y, title=title)
```

Output:

```
x = np.arange(5)
y = x ** 2

title = "plotly version: {}".format(plotly.__version__)
fig1 = go.Figure(go.Scatter(x=x, y=y), layout=dict(title=title))
fig2 = px.scatter(x=x, y=y, title=title)
```

## 1.6 iframe-size

Source:

```
.. plotly::
    :iframe-width: 500px
    :iframe-height: 300px

    import plotly.express as px
```

(continues on next page)

(continued from previous page)

```
px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

Output:

```
import plotly.express as px

px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

You can set the default `iframe-width` and `iframe-height` by specifying `plotly_iframe_width` (default: "100%") and `plotly_iframe_height` (default: "500px") in `conf.py`.

```
# conf.py

plotly_iframe_width = "500px"
plotly_iframe_height = "300px"
```

## 1.7 show

Source:

```
.. plotly::

    import plotly.express as px

    fig = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
    fig.show()
```

Output:

```
import plotly.express as px

fig = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
fig.show()
```

Source:

```
.. plotly::
    :fig-vars: figure

    import plotly.express as px

    figure = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
    figure.show()
```

Output:

```
import plotly.express as px

figure = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
figure.show()
```



## INDICES AND TABLES

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