qtap

Release 0.1.0

Aug 23, 2020

Contents:

CHAPTER 1

Function

Classes used for defining functions as Qt elements. Create a single widget using the `Function` class or combine multiple functions into a single widget using the `Functions` class

1.1 Function

class qtap.**Function**(*func: callable*, *arg_opts: dict = None*, *parent: Optional[PyQt5.QtWidgets.QWidget] = None*, *kwarg_entry: bool = False*)

__init__(*func: callable*, *arg_opts: dict = None*, *parent: Optional[PyQt5.QtWidgets.QWidget] = None*, *kwarg_entry: bool = False*)

Creates a widget based on the function signature

Parameters

- **func** (*callable*) – A function with type annotations
- **arg_opts** (*dict*) – manually set certain features of an Arg
- **parent** (*Optional[QtWidgets.QWidget]*) – parent QWidget
- **kwarg_entry** (*bool*) – Not yet implemented. include a text box for kwargs entry

sig_changed

Emitted when an argument value changes. Emits dict for all function arguments. See `get_data()` for details on the dict.

Type dict

sig_set_clicked

Emitted when the "Set" button is clicked. Emits dict for all function arguments. See `get_data()` for details on the dict.

Type dict

sig_arg_changed

Emitted when specific argument value changes. Emits argument name and argument value

Type str, object

Examples

Basic

```
from PyQt5 import QtWidgets
from qtap import Function

# annotated function
def f(a: int = 1, b: float = 3.14, c: str = 'yay', d: bool = True):
    pass

app = QtWidgets.QApplication([])

# basic
func = Function(f)
func.widget.show()

app.exec()
```

Opt Args

```
from PyQt5 import QtWidgets
from qtap import Function
from pyqtgraph.console import ConsoleWidget

def f(a: int = 1, b: float = 3.14, c: str = 'yay', d: bool = True):
    pass


if __name__ == '__main__':
    app = QtWidgets.QApplication([])

    # opt args dict
    opts =                      {
            'b':
                {
                    'use_slider': True,
                    'minmax': (0, 100),
                    'step': 1,
                    'suffix': '%',
                    'typ': int,
                    'tooltip': 'yay tooltips'
                }
        }

    func = Function(f, arg_opts=opts)
    func.widget.show()

    console = ConsoleWidget(parent=func.widget, namespace={'this': func})
    func.vlayout.addWidget(console)

    app.exec()
```

get_data() → Dict[str, object]

Get the data from the function arguments

Returns dict keys are the argument names, dict values are the argument vals

Return type dict

1.2 Functions

class qtap.**Functions**(*functions: List[callable], arg_opts: Optional[List[dict]] = None, parent: Optional[PyQt5.QtWidgets.QWidget] = None, scroll: bool = False, orient: str = 'V', columns: bool = False, **kwargs*)

__init__(*functions: List[callable], arg_opts: Optional[List[dict]] = None, parent: Optional[PyQt5.QtWidgets.QWidget] = None, scroll: bool = False, orient: str = 'V', columns: bool = False, **kwargs*)

Parameters

- **functions** (*List[callable]*) – list of functions
- **arg_opts** (*List[dict], optional*) – optional list of dicts to manually set features of an argument. passed to `Function`
- **parent** (*QtWidgets.QWidget, optional*) – parent widget
- **scroll** (*bool*) – Not yet implemented
- **orient** (*str*) – orientation of the individual functions. One of `V` or `H`. Default orientation is `V` (vertical)
- **columns** (*bool*) – Not yet implemented
- ****kwargs** – passed to QtWidgets.QWidget.__init__()

sig_changed
Emitted when an underlying function emits sig_changed(). The emitted dict comes from `get_data()`, see the docstring for `get_data()` for details.
Type dict

sig_set_clicked
Emitted when an underlying function emits sig_set_clicked(). The emitted dict comes from `get_data()`, see the docstring for `get_data()` for details.
Type dict

Examples

Basic

```
from PyQt5 import QtWidgets
from qtap import Functions
from pyqtgraph.console import ConsoleWidget

def func_A(a: int = 1, b: float = 3.14, c: str = 'yay', d: bool = True):
    pass


def func_B(x: float = 50, y: int = 2.7, u: str = 'bah'):
    pass


if __name__ == '__main__':
    app = QtWidgets.QApplication([])

    functions = Functions([func_A, func_B])

```

(continues on next page)

(continued from previous page)

```
    console = ConsoleWidget(parent=functions, namespace={'this': functions})
    functions.main_layout.addWidget(console)

    functions.show()

    app.exec()
```

Opt Args

```
from PyQt5 import QtWidgets
from qtap import Functions
from pyqtgraph.console import ConsoleWidget


def func_A(a: int = 1, b: float = 3.14, c: str = 'yay', d: bool = True):
    pass


def func_B(x: float = 50, y: int = 2.7, u: str = 'bah'):
    pass


if __name__ == '__main__':
    app = QtWidgets.QApplication([])

    # opt args for ``func_A``
    opts_A =                    {
            'b':
                {
                    'use_slider': True,
                    'minmax': (0, 100),
                    'step': 1,
                    'suffix': '%',
                    'typ': int,
                    'tooltip': 'yay tooltips'
                }
        }

    # functions where one has ``opt_args``
    functions = Functions(
        functions=[func_A, func_B],
        arg_opts=[opts_A, None], # opt_args in same order as functions
    )

    console = ConsoleWidget(parent=functions, namespace={'this': functions})
    functions.main_layout.addWidget(console)

    functions.show()

    app.exec()
```

get_data() → Dict[callable, dict]

Returns dict keys are the functions, each dict values is a kwargs dict

Return type dict

CHAPTER 2

Argument

Classes defining individual arguments as Qt elements.

2.1 Arg

class qtap.argument.**Arg**(*name: str, typ: type, val: Union[int, float, str, bool], parent: PyQt5.QtWidgets.QWidget, vlayout: PyQt5.QtWidgets.QVBoxLayout, tooltip: Optional[str] = None, **kwargs*)

__init__(*name: str, typ: type, val: Union[int, float, str, bool], parent: PyQt5.QtWidgets.QWidget, vlayout: PyQt5.QtWidgets.QVBoxLayout, tooltip: Optional[str] = None, **kwargs*)

Creates the appropriate QWidget interface.

Parameters

- **name** (*str*) – argument name
- **typ** (*type*) – function type, one of `int`, `float`, `str` or `bool`. Used for determining the correct QWidget to be used
- **val** (*Union[int, float, str, bool]*) – default value for the widget
- **parent** (*QtWidgets.QWidget*) – parent widget
- **vlayout** (*QtWidgets.QVBoxLayout*) – parent VBoxLayout
- **tooltip** (*str*) – toolTip

sig_changed

emits `self.val` when GUI value is changed.

Type object

name

argument name

val

current argument value

2.2 ArgNumeric

class qtap.argument.**ArgNumeric**(*name: str, typ: type, val: Union[int, float], parent: PyQt5.QtWidgets.QWidget, vlayout: PyQt5.QtWidgets.QVBoxLayout, minmax: tuple = (-1, 999), step: Union[int, float] = 1, use_slider: bool = False, suffix: str = None, **kwargs*)

Bases: *qtap.argument.Arg*

__init__(*name: str, typ: type, val: Union[int, float], parent: PyQt5.QtWidgets.QWidget, vlayout: PyQt5.QtWidgets.QVBoxLayout, minmax: tuple = (-1, 999), step: Union[int, float] = 1, use_slider: bool = False, suffix: str = None, **kwargs*)

Creates numerical QWidget interface

Parameters

- **minmax** (*tuple*) – min & max values
- **step** (*Union[int, float]*) – step size for the spin box
- **use_slider** (*Optional[bool]*) – adds a slider below the spin box
- **suffix** (*Optional[str]*) – text suffix for the spin box, like data units
- ****kwargs** – passed to Arg

max
max value limit for the widget

min
min value limit for the widget

minmax
minmax limits for the widget

name
argument name

step
step size for the widget

val
current argument value

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index

Symbols

A

F

G

M

N

S

V