

API Documentation

API Documentation

January 7, 2015

Contents

Contents	1
1 Package pylottosimu	3
1.1 Modules	3
1.2 Variables	3
2 Package pylottosimu.dialog	4
2.1 Modules	4
2.2 Variables	4
3 Module pylottosimu.dialog.lottosystem	5
3.1 Functions	5
3.2 Variables	5
3.3 Class str	5
3.3.1 Methods	6
3.3.2 Properties	14
3.4 Class LottoSettingsDialog	14
3.4.1 Methods	14
3.4.2 Properties	17
3.4.3 Class Variables	17
3.5 Class lottosystemdata	18
3.5.1 Methods	18
4 Module pylottosimu.dialog.show_drawing	19
4.1 Variables	19
4.2 Class DlgShowDrawing	20
4.2.1 Methods	20
4.2.2 Properties	22
4.2.3 Class Variables	22
5 Module pylottosimu.lottokugeln_rc	24
5.1 Functions	24
5.2 Variables	24
6 Module pylottosimu.lottokugeln_rc3	25
6.1 Functions	25
6.2 Variables	25

7	Module pylottosimu.lottokugeln_rc3_qt5	26
7.1	Functions	26
7.2	Variables	26
8	Module pylottosimu.pylotto	27
8.1	Functions	27
8.2	Variables	27
8.3	Class str	27
8.3.1	Methods	27
8.3.2	Properties	37
8.4	Class LottoSimuDialog	38
8.4.1	Methods	38
8.4.2	Properties	42
8.4.3	Class Variables	42
	Index	43

1 Package pylottosimu

1.1 Modules

- **dialog** (*Section 2, p. 4*)
 - **lottosystem**: pyLottoSimu (*Section 3, p. 5*)
 - **show_drawing**: pyLottoSimu (*Section 4, p. 19*)
- **lottokugeln_rc** (*Section 5, p. 24*)
- **lottokugeln_rc3** (*Section 6, p. 25*)
- **lottokugeln_rc3_qt5** (*Section 7, p. 26*)
- **pylotto**: The signals for the GUI (*Section 8, p. 27*)

1.2 Variables

Name	Description
<code>__package__</code>	Value: None

2 Package pylottosimu.dialog

2.1 Modules

- **lottosystem:** pyLottoSimu
(Section 3, p. 5)
- **show_drawing:** pyLottoSimu
(Section 4, p. 19)

2.2 Variables

Name	Description
__package__	Value: None

3 Module pylottosimu.dialog.lottosystem

pyLottoSimu

Copyright (C) <2012-2015> Markus Hackspacher

This file is part of pyLottoSimu.

pyLottoSimu is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

pyLottoSimu is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU General Public License along with pyLottoSimu. If not, see <<http://www.gnu.org/licenses/>>.

3.1 Functions

gui(arguments, sysdat)

Open the GUI

Parameters

arguments: language (en, de)
(type=string)

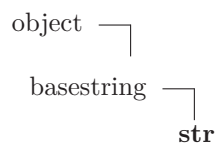
Return Value

none

3.2 Variables

Name	Description
__package__	Value: 'pylottosimu.dialog'

3.3 Class str



str(object) -> string

Return a nice string representation of the object. If the argument is a string, the return value is the same object.

3.3.1 Methods

<code>__add__</code> (<i>x</i> , <i>y</i>)
<code>x+y</code>

<code>__contains__</code> (<i>x</i> , <i>y</i>)
<code>y in x</code>

<code>__eq__</code> (<i>x</i> , <i>y</i>)
<code>x==y</code>

<code>__format__</code> (<i>S</i> , <i>format_spec</i>)
Return a formatted version of S as described by format_spec.
Return Value
string
Overrides: object.__format__

<code>__ge__</code> (<i>x</i> , <i>y</i>)
<code>x>=y</code>

<code>__getattr__</code> (...)
<code>x.__getattr__('name') <==> x.name</code>
Overrides: object.__getattr__

<code>__getitem__</code> (<i>x</i> , <i>y</i>)
<code>x[y]</code>

<code>__getnewargs__</code> (...)

<code>__getslice__</code> (<i>x</i> , <i>i</i> , <i>j</i>)
<code>x[i:j]</code>
Use of negative indices is not supported.

<code>__gt__</code> (<i>x</i> , <i>y</i>)
<code>x>y</code>

<code>__hash__</code> (<i>x</i>)
<code>hash(x)</code>
Overrides: object.__hash__

<code>__le__(x, y)</code>
<code>x<=y</code>

<code>__len__(x)</code>
<code>len(x)</code>

<code>__lt__(x, y)</code>
<code>x<y</code>

<code>__mod__(x, y)</code>
<code>x%y</code>

<code>__mul__(x, n)</code>
<code>x*n</code>

<code>__ne__(x, y)</code>
<code>x!=y</code>

<code>__new__(T, S, ...)</code>
Return Value a new object with type S, a subtype of T Overrides: object.__new__

<code>__repr__(x)</code>
<code>repr(x)</code> Overrides: object.__repr__

<code>__rmod__(x, y)</code>
<code>y%x</code>

<code>__rmul__(x, n)</code>
<code>n*x</code>

<code>__sizeof__(S)</code>
size of object in memory, in bytes
Return Value size of S in memory, in bytes Overrides: object.__sizeof__

__str__ (<i>x</i>)
<code>str(x)</code>
Overrides: <code>object.__str__</code>

capitalize (<i>S</i>)
Return a copy of the string <i>S</i> with only its first character capitalized.
Return Value string

center (<i>S, width, fillchar=...</i>)
Return <i>S</i> centered in a string of length <i>width</i> . Padding is done using the specified fill character (default is a space)
Return Value string

count (<i>S, sub, start=..., end=...</i>)
Return the number of non-overlapping occurrences of substring <i>sub</i> in string <i>S</i> [<i>start</i> : <i>end</i>]. Optional arguments <i>start</i> and <i>end</i> are interpreted as in slice notation.
Return Value int

decode (<i>S, encoding=..., errors=...</i>)
Decodes <i>S</i> using the codec registered for encoding. <i>encoding</i> defaults to the default encoding. <i>errors</i> may be given to set a different error handling scheme. Default is 'strict' meaning that encoding errors raise a <code>UnicodeDecodeError</code> . Other possible values are 'ignore' and 'replace' as well as any other name registered with <code>codecs.register_error</code> that is able to handle <code>UnicodeDecodeErrors</code> .
Return Value object

encode (<i>S, encoding=..., errors=...</i>)
Encodes <i>S</i> using the codec registered for encoding. <i>encoding</i> defaults to the default encoding. <i>errors</i> may be given to set a different error handling scheme. Default is 'strict' meaning that encoding errors raise a <code>UnicodeEncodeError</code> . Other possible values are 'ignore', 'replace' and 'xmlcharrefreplace' as well as any other name registered with <code>codecs.register_error</code> that is able to handle <code>UnicodeEncodeErrors</code> .
Return Value object

endswith(*S, suffix, start=..., end=...*)

Return True if S ends with the specified suffix, False otherwise. With optional start, test S beginning at that position. With optional end, stop comparing S at that position. suffix can also be a tuple of strings to try.

Return Value

bool

expandtabs(*S, tabsize=...*)

Return a copy of S where all tab characters are expanded using spaces. If tabsize is not given, a tab size of 8 characters is assumed.

Return Value

string

find(*S, sub, start=..., end=...*)

Return the lowest index in S where substring sub is found, such that sub is contained within S[start:end]. Optional arguments start and end are interpreted as in slice notation.

Return -1 on failure.

Return Value

int

format(*S, *args, **kwargs*)

Return a formatted version of S, using substitutions from args and kwargs. The substitutions are identified by braces ('{' and '}').

Return Value

string

index(*S, sub, start=..., end=...*)

Like S.find() but raise ValueError when the substring is not found.

Return Value

int

isalnum(*S*)

Return True if all characters in S are alphanumeric and there is at least one character in S, False otherwise.

Return Value

bool

isalpha(*S*)

Return True if all characters in S are alphabetic and there is at least one character in S, False otherwise.

Return Value

bool

isdigit(*S*)

Return True if all characters in *S* are digits and there is at least one character in *S*, False otherwise.

Return Value

bool

islower(*S*)

Return True if all cased characters in *S* are lowercase and there is at least one cased character in *S*, False otherwise.

Return Value

bool

isspace(*S*)

Return True if all characters in *S* are whitespace and there is at least one character in *S*, False otherwise.

Return Value

bool

istitle(*S*)

Return True if *S* is a titlecased string and there is at least one character in *S*, i.e. uppercase characters may only follow uncased characters and lowercase characters only cased ones. Return False otherwise.

Return Value

bool

isupper(*S*)

Return True if all cased characters in *S* are uppercase and there is at least one cased character in *S*, False otherwise.

Return Value

bool

join(*S*, *iterable*)

Return a string which is the concatenation of the strings in the iterable. The separator between elements is *S*.

Return Value

string

ljust(*S*, *width*, *fillchar*=...)

Return *S* left-justified in a string of length *width*. Padding is done using the specified fill character (default is a space).

Return Value

string

lower(*S*)

Return a copy of the string *S* converted to lowercase.

Return Value

string

lstrip(*S*, *chars*=...)

Return a copy of the string *S* with leading whitespace removed. If *chars* is given and not *None*, remove characters in *chars* instead. If *chars* is unicode, *S* will be converted to unicode before stripping

Return Value

string or unicode

partition(*S*, *sep*)

Search for the separator *sep* in *S*, and return the part before it, the separator itself, and the part after it. If the separator is not found, return *S* and two empty strings.

Return Value

(head, sep, tail)

replace(*S*, *old*, *new*, *count*=...)

Return a copy of string *S* with all occurrences of substring *old* replaced by *new*. If the optional argument *count* is given, only the first *count* occurrences are replaced.

Return Value

string

rfind(*S*, *sub*, *start*=... , *end*=...)

Return the highest index in *S* where substring *sub* is found, such that *sub* is contained within *S*[*start*:*end*]. Optional arguments *start* and *end* are interpreted as in slice notation.

Return -1 on failure.

Return Value

int

rindex(*S*, *sub*, *start*=... , *end*=...)

Like *S*.*rfind*() but raise *ValueError* when the substring is not found.

Return Value

int

rjust(*S*, *width*, *fillchar*=...)

Return *S* right-justified in a string of length *width*. Padding is done using the specified fill character (default is a space)

Return Value

string

rpartition(*S*, *sep*)

Search for the separator *sep* in *S*, starting at the end of *S*, and return the part before it, the separator itself, and the part after it. If the separator is not found, return two empty strings and *S*.

Return Value

(head, sep, tail)

rsplit(*S*, *sep*=... , *maxsplit*=...)

Return a list of the words in the string *S*, using *sep* as the delimiter string, starting at the end of the string and working to the front. If *maxsplit* is given, at most *maxsplit* splits are done. If *sep* is not specified or is *None*, any whitespace string is a separator.

Return Value

list of strings

rstrip(*S*, *chars*=...)

Return a copy of the string *S* with trailing whitespace removed. If *chars* is given and not *None*, remove characters in *chars* instead. If *chars* is unicode, *S* will be converted to unicode before stripping

Return Value

string or unicode

split(*S*, *sep*=... , *maxsplit*=...)

Return a list of the words in the string *S*, using *sep* as the delimiter string. If *maxsplit* is given, at most *maxsplit* splits are done. If *sep* is not specified or is *None*, any whitespace string is a separator and empty strings are removed from the result.

Return Value

list of strings

splitlines(*S*, *keepends*=**False**)

Return a list of the lines in *S*, breaking at line boundaries. Line breaks are not included in the resulting list unless *keepends* is given and true.

Return Value

list of strings

startswith(*S*, *prefix*, *start*=... , *end*=...)

Return True if *S* starts with the specified prefix, False otherwise. With optional *start*, test S beginning at that position. With optional *end*, stop comparing S at that position. *prefix* can also be a tuple of strings to try.

Return Value

bool

strip(*S*, *chars*=...)

Return a copy of the string *S* with leading and trailing whitespace removed. If *chars* is given and not None, remove characters in *chars* instead. If *chars* is unicode, *S* will be converted to unicode before stripping

Return Value

string or unicode

swapcase(*S*)

Return a copy of the string *S* with uppercase characters converted to lowercase and vice versa.

Return Value

string

title(*S*)

Return a titlecased version of *S*, i.e. words start with uppercase characters, all remaining cased characters have lowercase.

Return Value

string

translate(*S*, *table*, *deletechars*=...)

Return a copy of the string *S*, where all characters occurring in the optional argument *deletechars* are removed, and the remaining characters have been mapped through the given translation table, which must be a string of length 256 or None. If the *table* argument is None, no translation is applied and the operation simply removes the characters in *deletechars*.

Return Value

string

upper(*S*)

Return a copy of the string *S* converted to uppercase.

Return Value

string

zfill(*S*, *width*)

Pad a numeric string *S* with zeros on the left, to fill a field of the specified width. The string *S* is never truncated.

Return Value

string

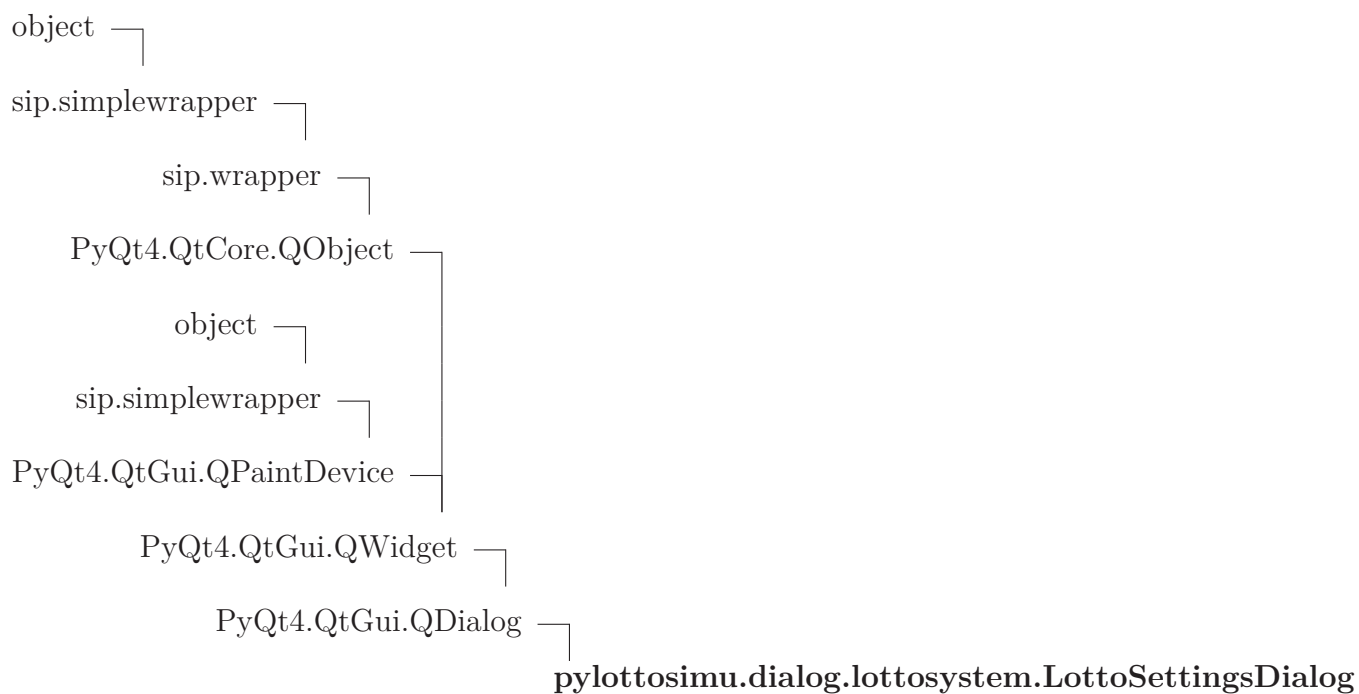
Inherited from object

`__delattr__()`, `__init__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`,
`__subclasshook__()`

3.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

3.4 Class `LottoSettingsDialog`



The GUI of Settings.

3.4.1 Methods

__init__ (<i>self</i> , <i>sysdat</i> , <i>parent=None</i>)
Initial user interface and slots
Return Value none
Overrides: <code>object.__init__</code>

init(*self*)

Initial variable

Return Value

none

sep__addit__numbers(*self*)**with__addit**(*self*)**setvalues**(*self*)

Set Values

values(*self*)

Values

getValues(*sysdat*, *parent*=None)

getValues

Inherited from PyQt4.QtGui.QDialog

accept(), accepted(), closeEvent(), contextMenuEvent(), done(), eventFilter(), exec_(), extension(), finished(), isSizeGripEnabled(), keyPressEvent(), minimumSizeHint(), open(), orientation(), reject(), rejected(), resizeEvent(), result(), setExtension(), setModal(), setOrientation(), setResult(), setSizeGripEnabled(), setVisible(), showEvent(), showExtension(), sizeHint()

Inherited from PyQt4.QtGui.QWidget

acceptDrops(), accessibleDescription(), accessibleName(), actionEvent(), actions(), activateWindow(), addAction(), addActions(), adjustSize(), autoFillBackground(), backgroundRole(), baseSize(), changeEvent(), childAt(), childrenRect(), childrenRegion(), clearFocus(), clearMask(), close(), contentsMargins(), contentsRect(), contextMenuPolicy(), create(), cursor(), customContextMenuRequested(), destroy(), devType(), dragEnterEvent(), dragLeaveEvent(), dragMoveEvent(), dropEvent(), effectiveWinId(), enabledChange(), ensurePolished(), enterEvent(), event(), find(), focusInEvent(), focusNextChild(), focusNextPrevChild(), focusOutEvent(), focusPolicy(), focusPreviousChild(), focusProxy(), focusWidget(), font(), fontChange(), fontInfo(), fontMetrics(), foregroundRole(), frameGeometry(), frameSize(), geometry(), getContentsMargins(), grabGesture(), grabKeyboard(), grabMouse(), grabShortcut(), graphicsEffect(), graphicsProxyWidget(), handle(), hasFocus(), hasMouseTracking(), height(), heightForWidth(), hide(), hideEvent(), inputContext(), inputMethodEvent(), inputMethodHints(), inputMethodQuery(), insertAction(),

insertActions(), isActiveWindow(), isAncestorOf(), isEnabled(), isEnabledTo(), isEnabledToTLW(), isFullScreen(), isHidden(), isLeftToRight(), isMaximized(), isMinimized(), isModal(), isRightToLeft(), isTopLevel(), isVisible(), isVisibleTo(), isWindow(), isWindowModified(), keyReleaseEvent(), keyboardGrabber(), languageChange(), layout(), layoutDirection(), leaveEvent(), locale(), lower(), mapFrom(), mapFromGlobal(), mapFromParent(), mapTo(), mapToGlobal(), mapToParent(), mask(), maximumHeight(), maximumSize(), maximumWidth(), metric(), minimumHeight(), minimumSize(), minimumWidth(), mouseDoubleClickEvent(), mouseGrabber(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), move(), moveEvent(), nativeParentWidget(), nextInFocusChain(), normalGeometry(), overrideWindowFlags(), overrideWindowState(), paintEngine(), paintEvent(), palette(), paletteChange(), parentWidget(), pos(), previousInFocusChain(), raise_(), rect(), releaseKeyboard(), releaseMouse(), releaseShortcut(), removeAction(), render(), repaint(), resetInputContext(), resize(), restoreGeometry(), saveGeometry(), scroll(), setAcceptDrops(), setAccessibleDescription(), setAccessibleName(), setAttribute(), setAutoFillBackground(), setBackgroundRole(), setBaseSize(), setContentsMargins(), setContextMenuPolicy(), setCursor(), setDisabled(), setEnabled(), setFixedHeight(), setFixedSize(), setFixedWidth(), setFocus(), setFocusPolicy(), setFocusProxy(), setFont(), setForegroundRole(), setGeometry(), setGraphicsEffect(), setHidden(), setInputContext(), setInputMethodHints(), setLayout(), setLayoutDirection(), setLocale(), setMask(), setMaximumHeight(), setMaximumSize(), setMaximumWidth(), setMinimumHeight(), setMinimumSize(), setMinimumWidth(), setMouseTracking(), setPalette(), setParent(), setShortcutAutoRepeat(), setShortcutEnabled(), setShown(), setSizeIncrement(), setSizePolicy(), setStatusTip(), setStyle(), setStyleSheet(), setTabOrder(), setToolTip(), setUpdatesEnabled(), setWhatsThis(), setWindowFilePath(), setWindowFlags(), setWindowIcon(), setWindowIconText(), setWindowModality(), setWindowModified(), setWindowOpacity(), setWindowRole(), setWindowState(), setWindowTitle(), show(), showFullScreen(), showMaximized(), showMinimized(), showNormal(), size(), sizeIncrement(), sizePolicy(), stackUnder(), statusTip(), style(), stylesheet(), tabletEvent(), testAttribute(), tooltip(), topLevelWidget(), underMouse(), ungrabGesture(), unsetCursor(), unsetLayoutDirection(), unsetLocale(), update(), updateGeometry(), updateMicroFocus(), updatesEnabled(), visibleRegion(), whatsThis(), wheelEvent(), width(), winId(), window(), windowActivationChange(), windowFilePath(), windowFlags(), windowIcon(), windowIconText(), windowModality(), windowOpacity(), windowRole(), windowState(), windowTitle(), windowType(), x(), x11Info(), x11PictureHandle(), y()

Inherited from PyQt4.QtCore.QObject

__getattr__(), blockSignals(), childEvent(), children(), connect(), connectNotify(), customEvent(), deleteLater(), destroyed(), disconnect(), disconnectNotify(), dumpObjectInfo(), dumpObjectTree(), dynamicPropertyNames(), emit(), findChild(), findChildren(), inherits(), installEventFilter(), isWidgetType(), killTimer(), metaObject(), moveToThread(), objectName(), parent(), property(), pyqtConfigure(), receivers(), removeEventFilter(), sender(), senderSignalIndex(), setObjectName(),

setProperty(), signalsBlocked(), startTimer(), thread(), timerEvent(), tr(), trUtf8()

Inherited from PyQt4.QtGui.QPaintDevice

colorCount(), depth(), heightMM(), logicalDpiX(), logicalDpiY(), numColors(), paintingActive(), physicalDpiX(), physicalDpiY(), widthMM()

Inherited from sip.simplewrapper

__new__()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

3.4.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

3.4.3 Class Variables

Name	Description
<i>Inherited from PyQt4.QtGui.QDialog</i> Accepted, Rejected	
<i>Inherited from PyQt4.QtGui.QWidget</i> DrawChildren, DrawWindowBackground, IgnoreMask	
<i>Inherited from PyQt4.QtCore.QObject</i> staticMetaObject	
<i>Inherited from PyQt4.QtGui.QPaintDevice</i> PdmDepth, PdmDpiX, PdmDpiY, PdmHeight, PdmHeightMM, PdmNumColors, PdmPhysicalDpiX, PdmPhysicalDpiY, PdmWidth, PdmWidthMM	

3.5 Class *lottosystemdata*

3.5.1 Methods

```
__init__(self, name='Lotto DE', max_draw=49, draw_numbers=6,  
with_addit=False, addit_numbers=0, sep_addit_numbers=False,  
max_addit=0)
```

```
writetofile(self)
```

4 Module `pylottosimu.dialog.show__drawing`

`pyLottoSimu`

Copyright (C) <2012-2014> Markus Hackspacher

This file is part of `pyLottoSimu`.

`pyLottoSimu` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

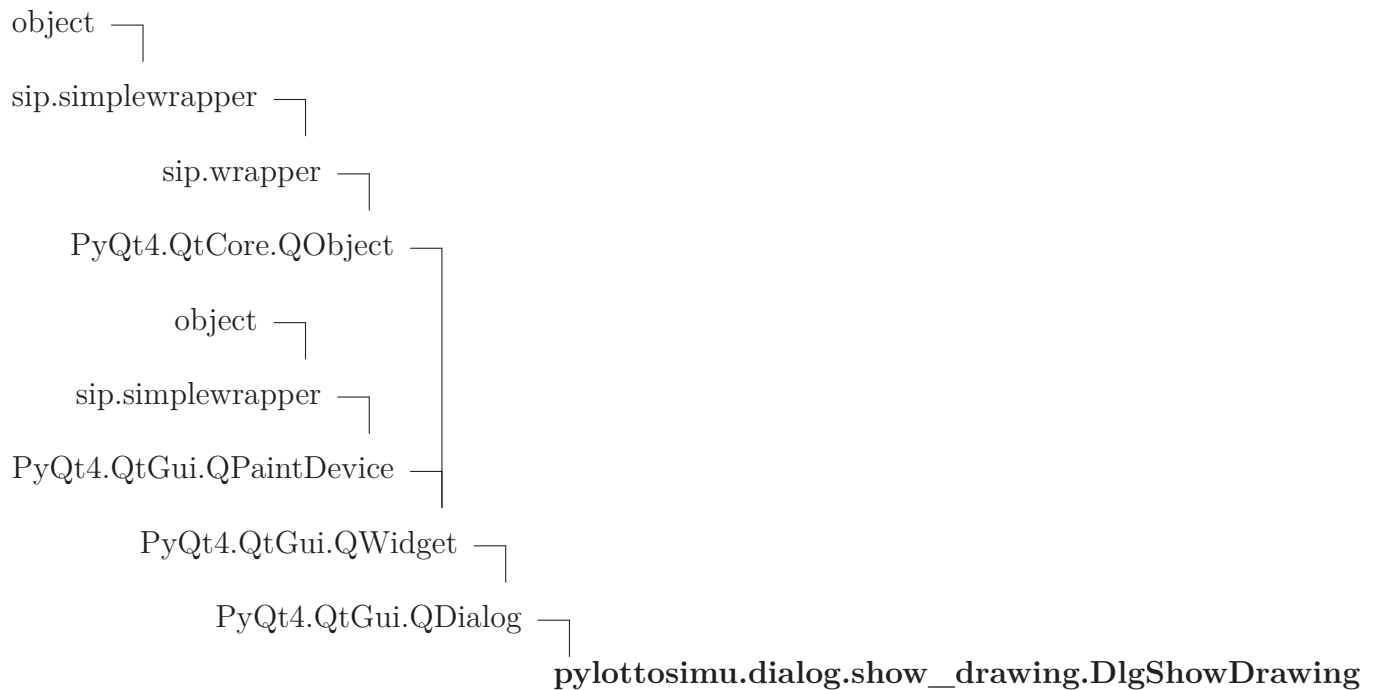
`pyLottoSimu` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU General Public License along with `pyLottoSimu`. If not, see <<http://www.gnu.org/licenses/>>.

4.1 Variables

Name	Description
<code>__package__</code>	Value: <code>'pylottosimu.dialog'</code>

4.2 Class DlgShowDrawing



Show the numbers in a dialog box

4.2.1 Methods

<code>__init__</code> (<i>self</i> , <i>draw_number</i> , <i>highest_number</i>)
x. <code>__init__</code> (...) initializes x; see <code>help(type(x))</code> for signature
Parameters
draw_number: the number of draw (<i>type=tuple of int</i>)
highest_number: the number of the PushButtons (<i>type=int</i>)
Return Value
none
Overrides: <code>object.__init__</code>

Inherited from `PyQt4.QtGui.QDialog`

`accept()`, `accepted()`, `closeEvent()`, `contextMenuEvent()`, `done()`, `eventFilter()`, `exec_()`, `extension()`, `finished()`, `isSizeGripEnabled()`, `keyPressEvent()`, `minimumSizeHint()`,

open(), orientation(), reject(), rejected(), resizeEvent(), result(), setExtension(), setModal(), setOrientation(), setResult(), setSizeGripEnabled(), setVisible(), showEvent(), showExtension(), sizeHint()

Inherited from PyQt4.QtGui.QWidget

acceptDrops(), accessibleDescription(), accessibleName(), actionEvent(), actions(), activateWindow(), addAction(), addActions(), adjustSize(), autoFillBackground(), backgroundRole(), baseSize(), changeEvent(), childAt(), childrenRect(), childrenRegion(), clearFocus(), clearMask(), close(), contentsMargins(), contentsRect(), contextMenuPolicy(), create(), cursor(), customContextMenuRequested(), destroy(), devType(), dragEnterEvent(), dragLeaveEvent(), dragMoveEvent(), dropEvent(), effectiveWinId(), enabledChange(), ensurePolished(), enterEvent(), event(), find(), focusInEvent(), focusNextChild(), focusNextPrevChild(), focusOutEvent(), focusPolicy(), focusPreviousChild(), focusProxy(), focusWidget(), font(), fontChange(), fontInfo(), fontMetrics(), foregroundRole(), frameGeometry(), frameSize(), geometry(), getContentsMargins(), grabGesture(), grabKeyboard(), grabMouse(), grabShortcut(), graphicsEffect(), graphicsProxyWidget(), handle(), hasFocus(), hasMouseTracking(), height(), heightForWidth(), hide(), hideEvent(), inputContext(), inputMethodEvent(), inputMethodHints(), inputMethodQuery(), insertAction(), insertActions(), isActiveWindow(), isAncestorOf(), isEnabled(), isEnabledTo(), isEnabledToTLW(), isFullScreen(), isHidden(), isLeftToRight(), isMaximized(), isMinimized(), isModal(), isRightToLeft(), isTopLevel(), isVisible(), isVisibleTo(), isWindow(), isWindowModified(), keyReleaseEvent(), keyboardGrabber(), languageChange(), layout(), layoutDirection(), leaveEvent(), locale(), lower(), mapFrom(), mapFromGlobal(), mapFromParent(), mapTo(), mapToGlobal(), mapToParent(), mask(), maximumHeight(), maximumSize(), maximumWidth(), metric(), minimumHeight(), minimumSize(), minimumWidth(), mouseDoubleClickEvent(), mouseGrabber(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), move(), moveEvent(), nativeParentWidget(), nextInFocusChain(), normalGeometry(), overrideWindowFlags(), overrideWindowState(), paintEngine(), paintEvent(), palette(), paletteChange(), parentWidget(), pos(), previousInFocusChain(), raise_(), rect(), releaseKeyboard(), releaseMouse(), releaseShortcut(), removeAction(), render(), repaint(), resetInputContext(), resize(), restoreGeometry(), saveGeometry(), scroll(), setAcceptDrops(), setAccessibleDescription(), setAccessibleName(), setAttribute(), setAutoFillBackground(), setBackgroundRole(), setBaseSize(), setContentsMargins(), setContextMenuPolicy(), setCursor(), setDisabled(), setEnabled(), setFixedHeight(), setFixedSize(), setFixedWidth(), setFocus(), setFocusPolicy(), setFocusProxy(), setFont(), setForegroundRole(), setGeometry(), setGraphicsEffect(), setHidden(), setInputContext(), setInputMethodHints(), setLayout(), setLayoutDirection(), setLocale(), setMask(), setMaximumHeight(), setMaximumSize(), setMaximumWidth(), setMinimumHeight(), setMinimumSize(), setMinimumWidth(), setMouseTracking(), setPalette(), setParent(), setShortcutAutoRepeat(), setShortcutEnabled(), setShown(), setSizeIncrement(), setSizePolicy(), setStatusTip(), setStyle(), setStyleSheet(), setTabOrder(), setToolTip(), setUpdatesEnabled(), setWhatsThis(), setWindowFilePath(),

setWindowFlags(), setWindowIcon(), setWindowIconText(), setWindowModality(), setWindowModified(), setWindowOpacity(), setWindowRole(), setWindowState(), setWindowTitle(), show(), showFullScreen(), showMaximized(), showMinimized(), showNormal(), size(), sizeIncrement(), sizePolicy(), stackUnder(), statusTip(), style(), styleSheet(), tabletEvent(), testAttribute(), toolTip(), topLevelWidget(), underMouse(), ungrabGesture(), unsetCursor(), unsetLayoutDirection(), unsetLocale(), update(), updateGeometry(), updateMicroFocus(), updatesEnabled(), visibleRegion(), whatsThis(), wheelEvent(), width(), winId(), window(), windowActivationChange(), windowFilePath(), windowFlags(), windowIcon(), windowIconText(), windowModality(), windowOpacity(), windowRole(), windowState(), windowTitle(), windowType(), x(), x11Info(), x11PictureHandle(), y()

Inherited from PyQt4.QtCore.QObject

__getattr__(), blockSignals(), childEvent(), children(), connect(), connectNotify(), customEvent(), deleteLater(), destroyed(), disconnect(), disconnectNotify(), dumpObjectInfo(), dumpObjectTree(), dynamicPropertyNames(), emit(), findChild(), findChildren(), inherits(), installEventFilter(), isWidgetType(), killTimer(), metaObject(), moveToThread(), objectName(), parent(), property(), pyqtConfigure(), receivers(), removeEventFilter(), sender(), senderSignalIndex(), setObjectName(), setProperty(), signalsBlocked(), startTimer(), thread(), timerEvent(), tr(), trUtf8()

Inherited from PyQt4.QtGui.QPaintDevice

colorCount(), depth(), heightMM(), logicalDpiX(), logicalDpiY(), numColors(), paintingActive(), physicalDpiX(), physicalDpiY(), widthMM()

Inherited from sip.simplewrapper

__new__()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.2.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

4.2.3 Class Variables

Name	Description
<i>Inherited from PyQt4.QtGui.QDialog</i> Accepted, Rejected	
<i>Inherited from PyQt4.QtGui.QWidget</i> DrawChildren, DrawWindowBackground, IgnoreMask	
<i>Inherited from PyQt4.QtCore.QObject</i> staticMetaObject	
<i>Inherited from PyQt4.QtGui.QPaintDevice</i> PdmDepth, PdmDpiX, PdmDpiY, PdmHeight, PdmHeightMM, PdmNumColors, PdmPhysicalDpiX, PdmPhysicalDpiY, PdmWidth, PdmWidthMM	

5 Module pylottosimu.lottokugeln_rc

5.1 Functions

qInitResources()

qCleanupResources()

5.2 Variables

Name	Description
qt_resource_data	Value: '\x00\x01\x94\x94\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\.
qt_resource_name	Value: '\x00\x0e\x00\xc9\x8e\xe7\x001\x00o\x00t\x00t\x00o\x00k\x.
qt_resource_struct	Value: '\x00\x00\x00\x00\x00\x02\x00\x00\x00\x01\x00\x00\x00\x01.
__package__	Value: 'pylottosimu'

6 Module pylottosimu.lottokugeln_rc3

6.1 Functions

qInitResources()

qCleanupResources()

6.2 Variables

Name	Description
qt_resource_data	Value: '\x00\x01\x94\x94\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\.
qt_resource_name	Value: '\x00\x0e\x00\xc9\x8e\xe7\x001\x00o\x00t\x00t\x00o\x00k\x.
qt_resource_struct	Value: '\x00\x00\x00\x00\x00\x02\x00\x00\x00\x01\x00\x00\x00\x01.
__package__	Value: 'pylottosimu'

7 Module pylottosimu.lottokugeln_rc3_qt5

7.1 Functions

<code>qInitResources()</code>

<code>qCleanupResources()</code>

7.2 Variables

Name	Description
qt_resource_data	Value: ...
qt_resource_name	Value: ...
qt_resource_struct	Value: ...

8 Module pylottosimu.pylotto

The signals for the GUI

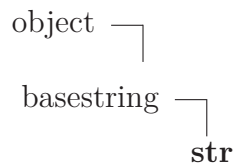
8.1 Functions

gui (<i>arguments</i>)
Open the GUI
Parameters
<i>arguments</i> : language (en, de) (<i>type=string</i>)
Return Value
none

8.2 Variables

Name	Description
<code>__doc__</code>	Value: "The signals for the GUI"
<code>__package__</code>	Value: 'pylottosimu'

8.3 Class str



`str(object) -> string`

Return a nice string representation of the object. If the argument is a string, the return value is the same object.

8.3.1 Methods

__add__ (<i>x, y</i>)
<code>x+y</code>

<code>__contains__</code> (<i>x</i> , <i>y</i>) y in x
<code>__eq__</code> (<i>x</i> , <i>y</i>) x==y
<code>__format__</code> (<i>S</i> , <i>format_spec</i>) Return a formatted version of S as described by format_spec. Return Value string Overrides: object.__format__
<code>__ge__</code> (<i>x</i> , <i>y</i>) x>=y
<code>__getattr__</code> (...) x.__getattr__('name') <==> x.name Overrides: object.__getattr__
<code>__getitem__</code> (<i>x</i> , <i>y</i>) x[y]
<code>__getnewargs__</code> (...)
<code>__getslice__</code> (<i>x</i> , <i>i</i> , <i>j</i>) x[i:j] Use of negative indices is not supported.
<code>__gt__</code> (<i>x</i> , <i>y</i>) x>y
<code>__hash__</code> (<i>x</i>) hash(x) Overrides: object.__hash__

<code>__le__(x, y)</code>
<code>x<=y</code>

<code>__len__(x)</code>
<code>len(x)</code>

<code>__lt__(x, y)</code>
<code>x<y</code>

<code>__mod__(x, y)</code>
<code>x%y</code>

<code>__mul__(x, n)</code>
<code>x*n</code>

<code>__ne__(x, y)</code>
<code>x!=y</code>

<code>__new__(T, S, ...)</code>
Return Value a new object with type S, a subtype of T
Overrides: object.__new__

<code>__repr__(x)</code>
<code>repr(x)</code>
Overrides: object.__repr__

<code>__rmod__(x, y)</code>
<code>y%x</code>

<code>__rmul__(x, n)</code>
<code>n*x</code>

__sizeof__(*S*)

size of object in memory, in bytes

Return Valuesize of *S* in memory, in bytes

Overrides: object.__sizeof__

__str__(*x*)

str(*x*)

Overrides: object.__str__

capitalize(*S*)Return a copy of the string *S* with only its first character capitalized.**Return Value**

string

center(*S*, *width*, *fillchar*=...)Return *S* centered in a string of length *width*. Padding is done using the specified fill character (default is a space)**Return Value**

string

count(*S*, *sub*, *start*=..., *end*=...)Return the number of non-overlapping occurrences of substring *sub* in string *S*[*start*:*end*]. Optional arguments *start* and *end* are interpreted as in slice notation.**Return Value**

int

decode(*S*, *encoding*=..., *errors*=...)

Decodes *S* using the codec registered for encoding. *encoding* defaults to the default encoding. *errors* may be given to set a different error handling scheme. Default is 'strict' meaning that encoding errors raise a `UnicodeDecodeError`. Other possible values are 'ignore' and 'replace' as well as any other name registered with `codecs.register_error` that is able to handle `UnicodeDecodeErrors`.

Return Value

object

encode(*S*, *encoding*=..., *errors*=...)

Encodes *S* using the codec registered for encoding. *encoding* defaults to the default encoding. *errors* may be given to set a different error handling scheme. Default is 'strict' meaning that encoding errors raise a UnicodeEncodeError. Other possible values are 'ignore', 'replace' and 'xmlcharrefreplace' as well as any other name registered with codecs.register_error that is able to handle UnicodeEncodeErrors.

Return Value

object

endswith(*S*, *suffix*, *start*=..., *end*=...)

Return True if *S* ends with the specified suffix, False otherwise. With optional *start*, test *S* beginning at that position. With optional *end*, stop comparing *S* at that position. *suffix* can also be a tuple of strings to try.

Return Value

bool

expandtabs(*S*, *tabsize*=...)

Return a copy of *S* where all tab characters are expanded using spaces. If *tabsize* is not given, a tab size of 8 characters is assumed.

Return Value

string

find(*S*, *sub*, *start*=..., *end*=...)

Return the lowest index in *S* where substring *sub* is found, such that *sub* is contained within *S*[*start*:*end*]. Optional arguments *start* and *end* are interpreted as in slice notation.

Return -1 on failure.

Return Value

int

format(*S*, **args*, ***kwargs*)

Return a formatted version of *S*, using substitutions from *args* and *kwargs*. The substitutions are identified by braces ('{' and '}').

Return Value

string

index(*S*, *sub*, *start*=... , *end*=...)

Like *S*.find() but raise ValueError when the substring is not found.

Return Value

int

isalnum(*S*)

Return True if all characters in *S* are alphanumeric and there is at least one character in *S*, False otherwise.

Return Value

bool

isalpha(*S*)

Return True if all characters in *S* are alphabetic and there is at least one character in *S*, False otherwise.

Return Value

bool

isdigit(*S*)

Return True if all characters in *S* are digits and there is at least one character in *S*, False otherwise.

Return Value

bool

islower(*S*)

Return True if all cased characters in *S* are lowercase and there is at least one cased character in *S*, False otherwise.

Return Value

bool

isspace(*S*)

Return True if all characters in *S* are whitespace and there is at least one character in *S*, False otherwise.

Return Value

bool

istitle(*S*)

Return True if *S* is a titlecased string and there is at least one character in *S*, i.e. uppercase characters may only follow uncased characters and lowercase characters only cased ones. Return False otherwise.

Return Value

bool

isupper(*S*)

Return True if all cased characters in *S* are uppercase and there is at least one cased character in *S*, False otherwise.

Return Value

bool

join(*S*, *iterable*)

Return a string which is the concatenation of the strings in the iterable. The separator between elements is *S*.

Return Value

string

ljust(*S*, *width*, *fillchar*=...)

Return *S* left-justified in a string of length *width*. Padding is done using the specified fill character (default is a space).

Return Value

string

lower(*S*)

Return a copy of the string *S* converted to lowercase.

Return Value

string

lstrip(*S*, *chars*=...)

Return a copy of the string *S* with leading whitespace removed. If *chars* is given and not None, remove characters in *chars* instead. If *chars* is unicode, *S* will be converted to unicode before stripping

Return Value

string or unicode

partition(*S*, *sep*)

Search for the separator *sep* in *S*, and return the part before it, the separator itself, and the part after it. If the separator is not found, return *S* and two empty strings.

Return Value

(head, sep, tail)

replace(*S*, *old*, *new*, *count*=...)

Return a copy of string *S* with all occurrences of substring *old* replaced by *new*. If the optional argument *count* is given, only the first *count* occurrences are replaced.

Return Value

string

rfind(*S*, *sub*, *start*=... , *end*=...)

Return the highest index in *S* where substring *sub* is found, such that *sub* is contained within *S*[*start*:*end*]. Optional arguments *start* and *end* are interpreted as in slice notation.

Return -1 on failure.

Return Value

int

rindex(*S*, *sub*, *start*=... , *end*=...)

Like *S*.*rfind*() but raise *ValueError* when the substring is not found.

Return Value

int

rjust(*S*, *width*, *fillchar*=...)

Return *S* right-justified in a string of length *width*. Padding is done using the specified fill character (default is a space)

Return Value

string

rpartition(*S*, *sep*)

Search for the separator *sep* in *S*, starting at the end of *S*, and return the part before it, the separator itself, and the part after it. If the separator is not found, return two empty strings and *S*.

Return Value

(head, sep, tail)

rsplit(*S*, *sep*=... , *maxsplit*=...)

Return a list of the words in the string *S*, using *sep* as the delimiter string, starting at the end of the string and working to the front. If *maxsplit* is given, at most *maxsplit* splits are done. If *sep* is not specified or is *None*, any whitespace string is a separator.

Return Value

list of strings

rstrip(*S*, *chars*=...)

Return a copy of the string *S* with trailing whitespace removed. If *chars* is given and not *None*, remove characters in *chars* instead. If *chars* is unicode, *S* will be converted to unicode before stripping

Return Value

string or unicode

split(*S*, *sep*=... , *maxsplit*=...)

Return a list of the words in the string *S*, using *sep* as the delimiter string. If *maxsplit* is given, at most *maxsplit* splits are done. If *sep* is not specified or is *None*, any whitespace string is a separator and empty strings are removed from the result.

Return Value

list of strings

splitlines(*S*, *keepends*=**False**)

Return a list of the lines in *S*, breaking at line boundaries. Line breaks are not included in the resulting list unless *keepends* is given and true.

Return Value

list of strings

startswith(*S*, *prefix*, *start*=..., *end*=...)

Return True if *S* starts with the specified prefix, False otherwise. With optional *start*, test *S* beginning at that position. With optional *end*, stop comparing *S* at that position. *prefix* can also be a tuple of strings to try.

Return Value

bool

strip(*S*, *chars*=...)

Return a copy of the string *S* with leading and trailing whitespace removed. If *chars* is given and not None, remove characters in *chars* instead. If *chars* is unicode, *S* will be converted to unicode before stripping

Return Value

string or unicode

swapcase(*S*)

Return a copy of the string *S* with uppercase characters converted to lowercase and vice versa.

Return Value

string

title(*S*)

Return a titlecased version of *S*, i.e. words start with uppercase characters, all remaining cased characters have lowercase.

Return Value

string

translate(*S*, *table*, *deletechars*=...)

Return a copy of the string *S*, where all characters occurring in the optional argument *deletechars* are removed, and the remaining characters have been mapped through the given translation table, which must be a string of length 256 or None. If the *table* argument is None, no translation is applied and the operation simply removes the characters in *deletechars*.

Return Value

string

upper(*S*)

Return a copy of the string *S* converted to uppercase.

Return Value

string

zfill(*S*, *width*)

Pad a numeric string *S* with zeros on the left, to fill a field of the specified width. The string *S* is never truncated.

Return Value

string

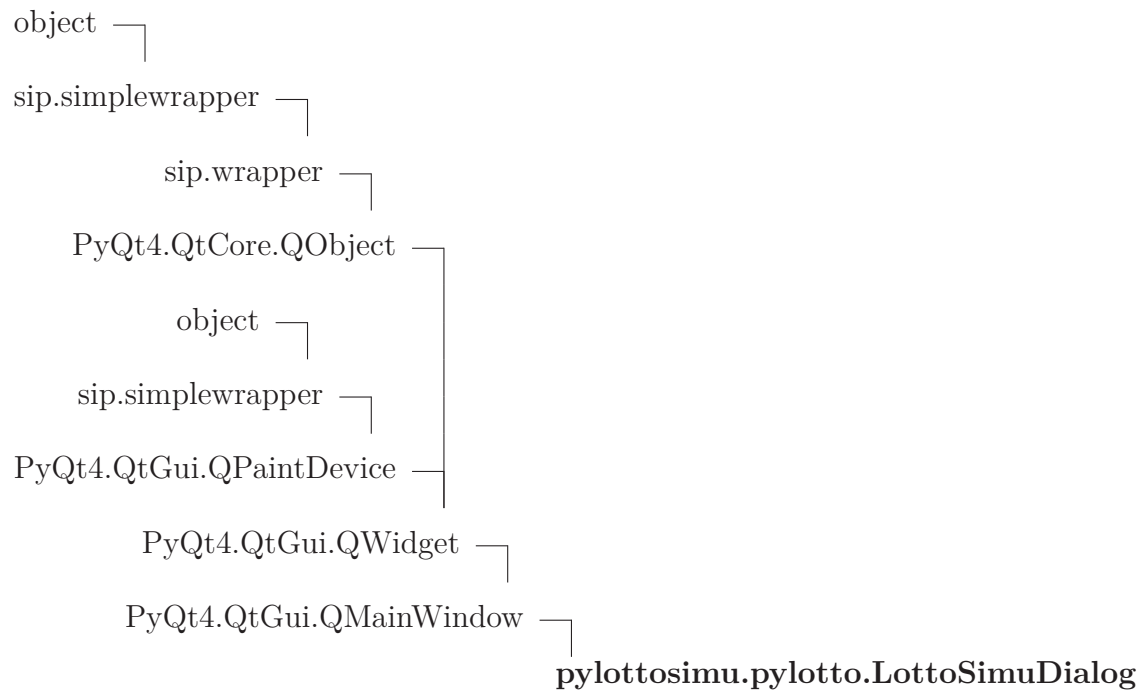
Inherited from object

`__delattr__()`, `__init__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`,
`__subclasshook__()`

8.3.2 Properties

Name	Description
<i>Inherited from object</i> <code>__class__</code>	

8.4 Class **LottoSimuDialog**



The GUI and programm of the pyLottoSimu.

8.4.1 Methods

<code>__init__(self)</code>
Initial user interface and slots
Return Value
none
Overrides: <code>object.__init__</code>

<code>init(self)</code>
Initial variable
Return Value
none

ontimer(*self*)

Start time to show a number.

Return Value

none

show__next__number(*self*)

Simulation of the draw and show the next Number on the Screen.

Return Value

none

onbtn__draw__overview(*self*)

show dialog of the draw

onsystem(*self*)

show dialog of the draw

onbtn__start(*self*)

Start simulation with the first drawing init timer with the valve from the Scrollbar the next drawing starts with the timer event.

Return Value

none

action__lottosim(*self*)

Changing the layout for simulation or generation Move the textedit and change the visible.

Return Value

none

onrandom__numbers__generator(*self*)

Show the output from the random number generator.

Return Value

none

onclean__output__text(*self*)

Clean the output text

Return Value

none

oninfo (<i>self</i>)
info message box
Return Value
none

onwebsite (<i>self</i>)
Open website
Return Value
none

onclose (<i>self</i>)
Close the GUI
Return Value
none

Inherited from PyQt4.QtGui.QMainWindow

addDockWidget(), addToolBar(), addToolBarBreak(), centralWidget(), contextMenuEvent(), corner(), createPopupMenu(), dockOptions(), dockWidgetArea(), documentMode(), event(), iconSize(), iconSizeChanged(), insertToolBar(), insertToolBarBreak(), isAnimated(), isDockNestingEnabled(), isSeparator(), menuBar(), menuWidget(), removeDockWidget(), removeToolBar(), removeToolBarBreak(), restoreDockWidget(), restoreState(), saveState(), setAnimated(), setCentralWidget(), setCorner(), setDockNestingEnabled(), setDockOptions(), setDocumentMode(), setIconSize(), setMenuBar(), setMenuWidget(), setStatusBar(), setTabPosition(), setTabShape(), setToolButtonStyle(), setUnifiedTitleAndToolBarOnMac(), splitDockWidget(), statusBar(), tabPosition(), tabShape(), tabifiedDockWidgets(), tabifyDockWidget(), toolBarArea(), toolBarBreak(), toolButtonStyle(), toolButtonStyleChanged(), unifiedTitleAndToolBarOnMac()

Inherited from PyQt4.QtGui.QWidget

acceptDrops(), accessibleDescription(), accessibleName(), actionEvent(), actions(), activateWindow(), addAction(), addActions(), adjustSize(), autoFillBackground(), backgroundRole(), baseSize(), changeEvent(), childAt(), childrenRect(), childrenRegion(), clearFocus(), clearMask(), close(), closeEvent(), contentsMargins(), contentsRect(), contextMenuPolicy(), create(), cursor(), customContextMenuRequested(), destroy(), devType(), dragEnterEvent(), dragLeaveEvent(), dragMoveEvent(), dropEvent(), effectiveWinId(), enabledChange(), ensurePolished(), enterEvent(), find(), focusInEvent(), focusNextChild(), focusNextPrevChild(), focusOutEvent(), focusPolicy(), focusPreviousChild(), focusProxy(), focusWidget(), font(), fontChange(), fontInfo(), fontMetrics(), foregroundRole(), frameGeometry(), frameSize(), geometry(), getContentsMargins(), grabGesture(), grabKeyboard(), grabMouse(), grabShortcut(),

graphicsEffect(), graphicsProxyWidget(), handle(), hasFocus(), hasMouseTracking(), height(), heightForWidth(), hide(), hideEvent(), inputContext(), inputMethodEvent(), inputMethodHints(), inputMethodQuery(), insertAction(), insertActions(), isActiveWindow(), isAncestorOf(), isEnabled(), isEnabledTo(), isEnabledToTLW(), isFullScreen(), isHidden(), isLeftToRight(), isMaximized(), isMinimized(), isModal(), isRightToLeft(), isTopLevel(), isVisible(), isVisibleTo(), isWindow(), isWindowModified(), keyPressEvent(), keyReleaseEvent(), keyboardGrabber(), languageChange(), layout(), layoutDirection(), leaveEvent(), locale(), lower(), mapFrom(), mapFromGlobal(), mapFromParent(), mapTo(), mapToGlobal(), mapToParent(), mask(), maximumHeight(), maximumSize(), maximumWidth(), metric(), minimumHeight(), minimumSize(), minimumSizeHint(), minimumWidth(), mouseDoubleClickEvent(), mouseGrabber(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), move(), moveEvent(), nativeParentWidget(), nextInFocusChain(), normalGeometry(), overrideWindowFlags(), overrideWindowState(), paintEngine(), paintEvent(), palette(), paletteChange(), parentWidget(), pos(), previousInFocusChain(), raise_(), rect(), releaseKeyboard(), releaseMouse(), releaseShortcut(), removeAction(), render(), repaint(), resetInputContext(), resize(), resizeEvent(), restoreGeometry(), saveGeometry(), scroll(), setAcceptDrops(), setAccessibleDescription(), setAccessibleName(), setAttribute(), setAutoFillBackground(), setBackgroundRole(), setBaseSize(), setContentsMargins(), setContextMenuPolicy(), setCursor(), setDisabled(), setEnabled(), setFixedHeight(), setFixedSize(), setFixedWidth(), setFocus(), setFocusPolicy(), setFocusProxy(), setFont(), setForegroundRole(), setGeometry(), setGraphicsEffect(), setHidden(), setInputContext(), setInputMethodHints(), setLayout(), setLayoutDirection(), setLocale(), setMask(), setMaximumHeight(), setMaximumSize(), setMaximumWidth(), setMinimumHeight(), setMinimumSize(), setMinimumWidth(), setMouseTracking(), setPalette(), setParent(), setShortcutAutoRepeat(), setShortcutEnabled(), setShown(), setSizeIncrement(), setSizePolicy(), setStatusTip(), setStyle(), setStyleSheet(), setTabOrder(), setToolTip(), setUpdatesEnabled(), setVisible(), setWhatsThis(), setWindowFilePath(), setWindowFlags(), setWindowIcon(), setWindowIconText(), setWindowModality(), setWindowModified(), setWindowOpacity(), setWindowRole(), setWindowState(), setWindowTitle(), show(), showEvent(), showFullScreen(), showMaximized(), showMinimized(), showNormal(), size(), sizeHint(), sizeIncrement(), sizePolicy(), stackUnder(), statusTip(), style(), styleSheet(), tabletEvent(), testAttribute(), tooltip(), topLevelWidget(), underMouse(), ungrabGesture(), unsetCursor(), unsetLayoutDirection(), unsetLocale(), update(), updateGeometry(), updateMicroFocus(), updatesEnabled(), visibleRegion(), whatsThis(), wheelEvent(), width(), winId(), window(), windowActivationChange(), windowFilePath(), windowFlags(), windowIcon(), windowIconText(), windowModality(), windowOpacity(), windowRole(), windowState(), windowTitle(), windowType(), x(), x11Info(), x11PictureHandle(), y()

Inherited from PyQt4.QtCore.QObject

__getattr__(), blockSignals(), childEvent(), children(), connect(), connectNo-

tify(), customEvent(), deleteLater(), destroyed(), disconnect(), disconnectNotify(), dumpObjectInfo(), dumpObjectTree(), dynamicPropertyNames(), emit(), eventFilter(), findChild(), findChildren(), inherits(), installEventFilter(), isWidgetType(), killTimer(), metaObject(), moveToThread(), objectName(), parent(), property(), pyqtConfigure(), receivers(), removeEventFilter(), sender(), senderSignalIndex(), setObjectName(), setProperty(), signalsBlocked(), startTimer(), thread(), timerEvent(), tr(), trUtf8()

Inherited from PyQt4.QtGui.QPaintDevice

colorCount(), depth(), heightMM(), logicalDpiX(), logicalDpiY(), numColors(), paintingActive(), physicalDpiX(), physicalDpiY(), widthMM()

Inherited from sip.simplewrapper

__new__()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

8.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

8.4.3 Class Variables

Name	Description
<i>Inherited from PyQt4.QtGui.QMainWindow</i>	
AllowNestedDocks, AllowTabbedDocks, AnimatedDocks, ForceTabbedDocks, VerticalTabs	
<i>Inherited from PyQt4.QtGui.QWidget</i>	
DrawChildren, DrawWindowBackground, IgnoreMask	
<i>Inherited from PyQt4.QtCore.QObject</i>	
staticMetaObject	
<i>Inherited from PyQt4.QtGui.QPaintDevice</i>	
PdmDepth, PdmDpiX, PdmDpiY, PdmHeight, PdmHeightMM, PdmNumColors, PdmPhysicalDpiX, PdmPhysicalDpiY, PdmWidth, PdmWidthMM	

Index

- pylottosimu (*package*), 3
 - pylottosimu.dialog (*package*), 4
 - pylottosimu.dialog.lottosystem (*module*), 5–18
 - pylottosimu.dialog.show_drawing (*module*), 19–23
 - pylottosimu.lottokugeln_rc (*module*), 24
 - pylottosimu.lottokugeln_rc.qCleanupResources (*function*), 24
 - pylottosimu.lottokugeln_rc.qInitResources (*function*), 24
 - pylottosimu.lottokugeln_rc3 (*module*), 25
 - pylottosimu.lottokugeln_rc3.qCleanupResources (*function*), 25
 - pylottosimu.lottokugeln_rc3.qInitResources (*function*), 25
 - pylottosimu.lottokugeln_rc3_qt5 (*module*), 26
 - pylottosimu.lottokugeln_rc3_qt5.qCleanupResources (*function*), 26
 - pylottosimu.lottokugeln_rc3_qt5.qInitResources (*function*), 26
 - pylottosimu.pylotto (*module*), 27–42
 - pylottosimu.pylotto.gui (*function*), 27
 - pylottosimu.pylotto.LottoSimuDialog (*class*), 37–42
- str (*class*), 5–14, 27–37
 - str.__add__ (*function*), 6, 27
 - str.__contains__ (*function*), 6, 27
 - str.__eq__ (*function*), 6, 28
 - str.__ge__ (*function*), 6, 28
 - str.__getitem__ (*function*), 6, 28
 - str.__getnewargs__ (*function*), 6, 28
 - str.__getslice__ (*function*), 6, 28
 - str.__gt__ (*function*), 6, 28
 - str.__le__ (*function*), 6, 28
 - str.__len__ (*function*), 7, 29
 - str.__lt__ (*function*), 7, 29
 - str.__mod__ (*function*), 7, 29
 - str.__mul__ (*function*), 7, 29
 - str.__ne__ (*function*), 7, 29
 - str.__rmod__ (*function*), 7, 29
 - str.__rmul__ (*function*), 7, 29
 - str.capitalize (*function*), 8, 30
 - str.center (*function*), 8, 30
 - str.count (*function*), 8, 30
 - str.decode (*function*), 8, 30
 - str.encode (*function*), 8, 30
 - str.endswith (*function*), 8, 31
 - str.expandtabs (*function*), 9, 31
 - str.find (*function*), 9, 31
 - str.format (*function*), 9, 31
 - str.index (*function*), 9, 31
 - str.isalnum (*function*), 9, 32
 - str.isalpha (*function*), 9, 32
 - str.isdigit (*function*), 9, 32
 - str.islower (*function*), 10, 32
 - str.isspace (*function*), 10, 32
 - str.istitle (*function*), 10, 32
 - str.isupper (*function*), 10, 33
 - str.join (*function*), 10, 33
 - str.ljust (*function*), 10, 33
 - str.lower (*function*), 10, 33
 - str.lstrip (*function*), 11, 33
 - str.partition (*function*), 11, 33
 - str.replace (*function*), 11, 34
 - str.rfind (*function*), 11, 34
 - str.rindex (*function*), 11, 34
 - str.rjust (*function*), 11, 34
 - str.rpartition (*function*), 11, 34
 - str.rsplitleft (*function*), 12, 35
 - str.rstrip (*function*), 12, 35
 - str.split (*function*), 12, 35
 - str.splitlines (*function*), 12, 35
 - str.startswith (*function*), 12, 35
 - str.strip (*function*), 12, 36
 - str.swapcase (*function*), 13, 36
 - str.title (*function*), 13, 36
 - str.translate (*function*), 13, 36
 - str.upper (*function*), 13, 36
 - str.zfill (*function*), 13, 37