[python 常用快捷键](https://www.cnblogs.com/fcc-123/p/10704048.html)

**最重要的快捷键**

1. ctrl+shift+A:万能命令行
2. shift两次:查看资源文件

**新建工程第一步操作**

1. module设置把空包分层去掉,compact empty middle package
2. 设置当前的工程是utf-8,设置的Editor-->File Encodings-->全部改成utf-8,

**注释**

1. 可以选中多行后注释ctrl+/:单行注释

**光标操作**

1. ctrl+alt+enter:向上插入
2. shift+enter:向下插入
3. end:光标

**操作代码**

1. ctrl+d:复制粘贴一行
2. ctrl+y:删除一行
3. shift+F6:重命令
4. ctrl+O:复写代码

**格式代码及其他功能**

1. ctrl+alt+L:格式代码
2. 在代码中使用alt+insert:Generate,可以get/set等操作
3. ctrl+alt+T:添加try/catch
4. ctrl+alt+M:抽取代码
5. ctrl+alt+F:变量抽取全局变量
6. 还需要设置前缀:Editor-->code style-->java-->code Genertion-->设置Field的前缘为m添加
7. ctrl+alt+v:方法体内值抽取成变量
8. 保存成模板:ctrl+shift+L,这个是自定义的(save as live Template)
9. 选中内容:tab进行退格
10. shift+tab:反向退格
11. alt+shift+上下键:选中代码移动
12. ctrl+shift+上下键:可以移动当前方法体,如果移动一行代码只能在代码体内移动
13. ctrl+shift+U:代码大小写
14. ctrl+shift+enter:补全代码(一行尾添加分号,如果是if等添加括号)

**进入代码**

1. ctrl+鼠标:进入代码
2. ctrl+B:进入代码
3. ctrl+shift+12:最大化窗口

**替换查找**

1. ctrl+r:替换
2. ctrl+F:查找
3. ctrl+shift+F:全局查找
4. ctrl+shift+R:全局替换
5. ctrl+shift+i:快捷查看方法实现的内容
6. ctrl+p:查看参数
7. ctrl+Q:查看文档描述
8. shift+F1:查看api文档
9. ctrl+F12:查看类的方法
10. ctrl+H:查看类的继承关系
11. 查看变量的赋值情况:
12. shift+ctrl+a:输入analyze data flow to Here
13. ctrl+alt+H:查看方法在那里被调用了
14. ctrl+{}:可以定位方法体的括号
15. F3:查看选中的内容
16. shift+F3:反向查看内容
17. ctrl+alt+B:查询那些类实现了光标所在的接口
18. ctrl+U:查看父类
19. ctrl+E:最近编辑的文件列表
20. ctrl+alt+home:查看布局与对应的类
21. ctrl+alt+H:查看当前方法在那里进行调用

**运行编译**

1. ctrl+F9:构建
2. shift+F10:运行

**工程目录操作**

1. 新建文件及工程:选中要创建目录使用alt+insert
2. ctrl+shift+a:输入show in explorer-->打开相应目录
3. ctrl+alt+s:打开软件设置
4. ctrl+alt+shift+s:打开module设置
5. alt+1:当前目录区
6. alt+7:当前类的方法列表查看
7. ctrl+tab:切换目录及视图
8. alt+shift+c:查看工程最近更改的地方
9. ctrl+J:livetemp模板查看

**代码快捷操作**

1. 没有操作完成操作可以先写todo进行,就可以在todo的窗口进行查看
2. F11定义书签
3. shift+F11:查看书签
4. ctrl+J:快捷调出模板
5. alt+点击断点:禁用断点
6. 调试状态下按下:alt查看变量能审查表达式的值

**组合快捷键**

1. F2:定位错误
2. alt+enter:修正错误
3. alt+鼠标:进入列编辑模式
4. ctrl+w:选中单词
5. 或其他组合操作

import turtle  
t = turtle.Pen()  
color = ["red", "yellow", "blue", "green", "orange"]  
for x in range(1000):  
 t.pencolor(color[x % 5])  
 t.forward(x)  
 t.left(91)

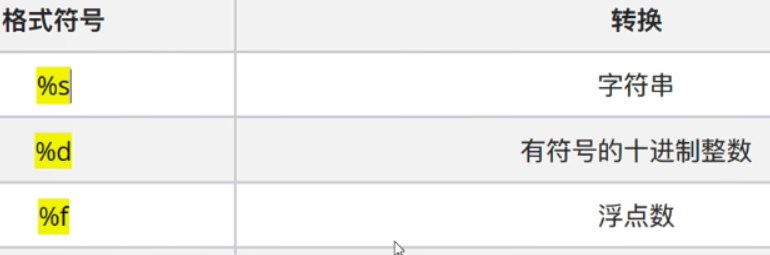
import turtle  
t = turtle.Pen()  
turtle.bgcolor("green")  
color = ("rad", "green")  
for a in range(100):  
 turtle.speed(700)  
 t.penup()  
 t.forward(a\*10)  
 t.write(a, font=("HOT-Ninja Std R", 20, "bold"))  
 t.pendown()  
 t.left(90)

1起名习惯 大小驼峰以及下划线

2注释#及六引号

3变量类型

4格式变量（简洁）



位数（整数%03d（三位整数） 小数%.3f（三位小数））

多个值时%（，，，）

5#表示注释

6\n表示换行符号

7eval（数字）

Round（）转换为最近整数并赋值

链式调用

Student student = new Student().setStudentName("jhon")  
              .seAge(18)  
              .setSex("M")  
              .setNumber("11011010");

Setbackground必须得在单元格有内容的情况下才能使用

PYQT5

气泡提示：引入QToolTip

设置程序图标：引入QIcon

def initUI(self):

QToolTip.setFont(QFont('SansSerif', 10)) #设置气泡提示字体的格式

self.setToolTip("这是一个<b>气泡提示</b>")

self.setWindowIcon(QIcon("Python2.ico"))

##### 快速定位指定行

* 当表格中的行数过多时，可以通过直接输入行号定位并显示，示例：
* import sys
* from PyQt5.QtCore import \*
* from PyQt5.QtGui import \*
* from PyQt5.QtWidgets import \*
* class Table(QWidget):
* def \_\_init\_\_(self):
* super(Table, self).\_\_init\_\_()
* self.initUI()
* def initUI(self):
* self.setWindowTitle("QTableWidget 示例2")
* self.setWindowIcon(QIcon("./images/Python2.ico"))
* self.resize(600, 800)
* conLayout = QHBoxLayout()
* tableWidget = QTableWidget()
* tableWidget.setRowCount(30)
* tableWidget.setColumnCount(4)
* conLayout.addWidget(tableWidget)
* for i in range(30):
* for j in range(4):
* itemContent = '(%d, %d)' % (i, j)
* tableWidget.setItem(i, j, QTableWidgetItem(itemContent))
* self.setLayout(conLayout)
* # 遍历表格查询对应项
* text = "(10, 1)"
* items = tableWidget.findItems(text, Qt.MatchExactly)
* item = items[0]
* # 选中单元格
* item.setSelected(True)
* # 设置单元格的背景色为红色
* item.setForeground(QBrush(QColor(255, 0, 0)))
* row = item.row()
* # 通过鼠标滚轮定位，快速定位到 11 行
* tableWidget.verticalScrollBar().setSliderPosition(row)
* if \_\_name\_\_ == "\_\_main\_\_":
* app = QApplication(sys.argv)
* example = Table()
* example.show()

sys.exit(app.exec\_())

本文实例讲述了python保存字符串到文件的方法。分享给大家供大家参考。具体实现方法如下：

|  |  |
| --- | --- |
| 1  2  3  4  5 | def save(filename, contents):    fh = open(filename, 'w')    fh.write(contents)    fh.close()  save('file.name', 'some stuff') |

插入一列或者一行

1. self.tableWidget.insertColumn(0) *#插入到第一列*
2. self.tableWidget.insertRow(0) *#插入到第一行*

tableWidget.sortItems(2, QtCore.Qt.DescendingOrder)

1）Qt.DescendingOrder 降序排序  
2）Qt.AscendingOrder 升序排序

Pyecharts的使用

<https://zhuanlan.zhihu.com/p/95625996>

<https://www.jianshu.com/p/554d64470ec9>

若想输出图像pyecharts 提供了 selenium 和 phantomjs 两种方式。

第一种方法需要安装[snapshot-selenium](https://github.com/pyecharts/snapshot-selenium)，并且需要做一些其他的配置。（[snapshot-selenium](https://github.com/pyecharts/snapshot-selenium) 是 pyecharts + selenium 渲染图片的扩展，使用 selenium 需要配置 browser driver，这部分可以参考 [selenium-python](https://selenium-python.readthedocs.io/installation.html#drivers) 相关介绍，推荐使用 Chrome 浏览器，可以开启 headless 模式。目前支持 Chrome, Safari。）

pip install snapshot-selenium

第二种方法需要安装[snapshot-phantomjs](https://github.com/pyecharts/snapshot-phantomjs)。（[snapshot-phantomjs](https://github.com/pyecharts/snapshot-phantomjs) 是 pyecharts + phantomjs 渲染图片的扩展，需要先安装 phantomjs，安装方法请参照官网 [phantomjs.org/download.html](http://phantomjs.org/download.html)。）

pip install snapshot-phantomjs

pantomjs 下载好后还需要将bin目录添加到PATH

当用connect时，里面调用函数时，需要给他用lambda定义

<https://blog.csdn.net/f156207495/article/details/77540027>

# PyQt的QTableWidget的全面总结与归纳

<https://www.jb51.net/article/181326.htm>

# python GUI库图形界面开发之PyQt5控件QTableWidget详细使用方法与属性

<https://zhuanlan.zhihu.com/p/34532247>

# PYQT5中QTableWidget的使用！

<https://blog.csdn.net/gefeng0510/article/details/108007307>

# pyqt5的QTableWidget动态添加数据并获取单元格内容

<https://blog.csdn.net/a429367172/article/details/85328315>

# pyqt5练习笔记——QTableWidget数据表的增删改

<https://blog.csdn.net/weixin_39964552/article/details/82937144>

# 基于python3.6+pyQT5利用Graphics View 控件显示图像并实现其缩放

<https://blog.csdn.net/weixin_41888503/article/details/79802443?utm_source=blogxgwz6>

# Python将二维列表（list）的数据输出（TXT，Excel）

<https://www.cnblogs.com/cloud-ken/p/10019253.html>

# python创建与遍历List二维列表

<https://www.jb51.net/article/142196.htm>

# python中的二维列表实例详解

<https://www.cnblogs.com/liming19680104/p/11294589.html>

# [pyqt5-表格TableWidGet](https://www.cnblogs.com/liming19680104/p/11294589.html)

<https://blog.csdn.net/panrenlong/article/details/79948261>

# PyQt5的输入对话框使用（QInputDialog）

<https://c.m.163.com/news/a/F9S7E0RG0545C5L0.html>

# Python用PyQt5实现多种弹出消息框，QMessageBox控件使用详解

<https://m.zhezhier.com/view/164408152/>

# 解决PyQt5的tableWidget数据行不能完全删除问题

<http://www.manongjc.com/detail/14-cadzjxhnfrpitbk.html>

# pyqt5 tableWidget用法

<https://blog.csdn.net/qq_38161040/article/details/109181602>

# PyQt5 技术篇-QTableWidget表格组件指定行的隐藏与显示控制实例演示，设置表格指定列的列宽方法

<https://blog.csdn.net/weixin_44731100/article/details/98876755>

# python中如何判断一个字符串是否可以转换为数字



table = QtGui.QTableWidget(5, 3, self)

for row in range(5):

for col in range(3):

table.setItem(row, col, QtGui.QTableWidgetItem("(%d, %d)" % (row, col)))

print("1,0: %s" % table.item(1, 0).text())

Python中符合序列的有序序列都支持切片（slice），例如列表，字符串，元组。

     格式：【start:end:step】

     start:起始索引，从0开始，-1表示结束

     end：结束索引

     step：步长，end-start，步长为正时，从左向右取值。步长为负时，反向取值

    注意切片的结果不包含结束索引，即不包含最后的一位，-1代表列表的最后一个位置索引

a=[1,2,3,4,5,6]

b1=a[:] #省略全部，代表截取全部内容，可以用来将一个列表拷给另一个列表  
  
print(b1)

结果：[1, 2, 3, 4, 5, 6]

b=a[0:-1:1] #从位置0开始到结束，每次增加1，截取。不包含结束索引位置

print(b)

结果：[1, 2, 3, 4, 5]

c1=a[:3] #省略起始位置的索引，以及步长。默认起始位置从头开始，默认步长为1，结束位置索引为3

print(c1)

结果：[1, 2, 3]

c=a[0:5:3] #从第一个位置到第6个位置，每3个取一个值

print(c)

结果：[1, 4]

[复制代码](javascript:void(0);)

d=a[5:0:-1] #反向取值

print(d)

结果：[6, 5, 4, 3, 2]

d1=a[::-1]

print(d1)

结果：[6, 5, 4, 3, 2, 1]