Python的编辑器有很多比如 Visual Studio Code 、 Sublime Text 、 Atom 、 jupyter notebook 等 等

但是功能最强使用最多的还是 PyCharm,同时也是我使用的最顺手的一款IDE。它是由JetBrains打造的一款功能强大的Python IDE。比如代码调试、项目管理、代码跳转、智能提示、单元测试、版本控制等等。具有跨平台性,无论Windows、Mac、Linux都可以使用。

一款好的IDE能够帮我们快速入门一个新的语言,对于初次使用PyCharm的同学,可能无从下手,这也是我编写这份手册的初衷,希望能够帮助到大家更效率的使用PyCharm。



作者: 阿亮。公众号: Python极客专栏

声明:版权归个人所有。仅供学习使用,不允许用作商业及个人牟利/引流等用途

本使用手册为V1.0.0版本(2021年3月17日) 操作系统Window10 PyCharm版本 2020.3.3 Python解释器版本: 3.9.2

PyCharm的安装

当然时间2021年3月11日, 官方最新版本为 2020.3.3

安装PyCham系统要求:

- Microsoft Windows 10、8的64位版本
- 最低2 GB RAM, 建议8 GB RAM
- 2.5 GB硬盘空间,建议使用SSD
- 最低1024x768屏幕分辨率
- Python 2.7或Python 3.5或更高版本

官方下载地址: <u>https://www.jetbrains.com/pycharm/download</u>



从上图可以看到,有两个版本分别是 Professional 和 Community,他们的区别如下表所示。

	PyCharm专业版 (Professional)	PyCharm社区版本 (Community)
智能Python编辑器	支持	支持
图形化调试器	支持	支持
导航和重构	支持	支持
代码检查	支持	支持
VCS支持	支持	支持
科学工具	支持	
Web开发	支持	
Python Web框架	支持	
Python Profiler(性能分 析)	支持	
远程开发功能	支持	
数据库和SQL支持	支持	

社区版本是免费的,基本满足日常开发需求

如果下载PyCharm专业版,即Professional。但是只能免费试用30天,可以选择以下几种激活方式

- 购买正版
- 网上也有激活方法 (参考: <u>https://shimo.im/docs/9wGqjpDWwxYcrCkH</u>)
- 学生/老师可申请免费: 申请地址: <u>https://www.jetbrains.com/community/education/</u>

PyCharm下载

直接点击 Download 就可以进行下载了。下载完成之后双击打开 pycharm-professional-2020.3.3.exe 然后点击 Next ,中间记得选择安装路径, 不建议安装在C盘,如果你的C盘很大很大,那完全可以。

😰 PyCharm Setup			_		×
PC	Choose Inst Choose the fol	all Location der in which to install P	yCharm.		
Setup will install PyCha and select another fol Destination Folder	arm in the following fol der. Click Next to cont	der. To install in a diffe inue.	rent folder, d	lick Brows	e
E:\PyCharm\PyCh	arm 2020.3.3		B <u>r</u> ow	/se	
Space required: 1.1 G Space available: 11.8	B GB				
		< <u>B</u> ack	<u>N</u> ext >	Cano	el

然后点击 Next 会看到如下界面,建议勾选箭头所指的选项

🛐 PyCharm Setup			_		×
P	Installation Option Configure your PyCh	ns arm installation			
Create Desktop Shortcut	roject"	Update PATH va	riable (res s dir to the	tart neede PATH	ed)
.py					
		< <u>B</u> ack <u>N</u> ex	kt >	Cano	el

选项翻译如下:

- create desktop shortcut: 创建桌面快捷方式
- update path variable(restart needed):更新路径变量将路径添加到环境变量中
- update context menu: 更新通知
- create associations:关联.py文件,双击都是以pycharm打开



提示重启,不重启也没啥影响。

Project

≌ Welcome to PyCharm (Admin	istrator)		—	
PyCharm 2020.3.3		New Project Op		Get from VCS
Projects				
Customize				
Plugins				
Learn PyCharm				
A				
•				

- New Project: 创建项目
- Open: 导入并打开本地项目,
- Get from VCS: 配置GitHub/SVN仓库地址

Customize

醛 Welcome to PyCharm (Admin			
PyCharm 2020.3.3 Projects	Color theme Darcula Sync with OS		
Customize	Accessibility		
Plugins	IDE font: 12 💌		
Learn PyCharm	Editor font: 13 💌		
	Adjust colors for red-green vision deficiency How it works Requires restart. For protanopia and deuteranopia		
	Keymap		
	Windows Configure		
	Import Settings		
Ψ.			

- Color theme: 主题风格, 以我本机为例, 默认有 Darcula、 Intellij Light、 windows 10 Light、 High contrast
- IDE font: 修改界面字体大小, 建议默认12的就可以了
- Editor font: : 控制代码字体大小以及输出面板中的字体大小
- Adjust:针对红绿色盲用户设计的。
- Keymap: 快捷键风格, 有Windos、EMacs、Sublime Text 风格

Plugins

显示安装的插件,也可以在这个界面进行在线安装。



Learn PyCharm

官方教程,英文好的同学可以看一下

环境配置

这一篇是给新手朋友准备的,如果你本地已经安装配置,请自行跳过 Python代码运行,需要解释器,Python解释器下载地址:<u>https://www.python.org/</u>

鼠标悬停在 Downloads 上, 然后选择对应的操作系统, 点击版本号即可。我这里以 Python 3.9.2 为例

官网下载较慢,可以在公众号: Python极客专栏,后台回复【python392】获取安装包。

ѐ python"				onate Search	1		GO Socialize
About	Downloads	Documentatio	on Community	Success Stories	News	Events	
<pre># Python 3: Fib >>> def fib(n): >>> a, b = >>> while a >>> pri >>> a, >>> print() >>> fib(1000) 0 1 1 2 3 5 8 1</pre>	All releases Source ode Windows Mac OS X Other Platforms License Alternative Implement	nentations	Pownload for Window Python 3.9.2 Note that Python 3.9+ ca earlier. Not the OS you are looking many operating systems a /iew the full list of downlo	s <i>nnot</i> be used on Window g for? Python can be used nd environments. ads.	is 7 or on	ng functions. ments, keyword s. <u>More about</u>	
	Python is a and int	a programmi tegrate syste	ing language that ms more effectiv	t lets you work qı ely. <u>>>></u> <u>Learn Mo</u>	uickly <u>re</u>		

下载完毕,双击打开(建议以管理员身份运行)。

不建议按照默认的方式安装,参考下图



点击之后出现如下界面



浏览文件夹	< 🍫 Python 3.9.2 (64-bit) Setup —	
Python 3.9.2 (64-bit) Setup	Advanced Options	
 > Lust Epidemic v1.0 > MD交档 > MobaXterm MySQL > Navicate > Program Files (x86) > PyCharm > PyCharm_workspace 	 ☑ Associate files with Python (requires the py launcher) ☑ Create shortcuts for installed applications ☑ Add Python to gnvironment variables ☑ Precompile standard library ☑ Download debugging symbols ☑ Download debug binaries (requires VS 2017 or later) 修改安装路径 	
Python 固试文档 文件夹(F): Python	Customize instail location E:\Python You will require write permissions for the selected location. Windows Back	B <u>r</u> owse Cancel

- Install for all users: 为所有用户安装(选装)
- Associate files with Python: Python文件关联 (必选)
- Create shortcuts for Installed application:为已安装的应用程序创建快捷方式(必选)
- Add Python to environment variables:添加Python环境变量(必选)
- Precompile standard library: 预编译python标准文件(推荐勾选)
- Download debugging symbol:下载调试符号(推荐勾选)
- Download debug binaries(require VS 2017 or later):下载调试二进制文件(需要VS 2017或 更高版本) (**推荐勾选**)

勾选完毕之后点击 Install 进行安装,安装速度可能有点慢,耐心等耐即可。

눩 Python 3.9.2 (64-bit) Setup		-		\times
	Setup Progress			
_	Installing:			
	Python 3.9.2 Core Interpreter (64-bit)			
python windows		-	Canco	1
		L.		·)

安装完毕之后如下图所示。

눩 Python 3.9.2 (64-bit) Setup



查看是否安装成功

🖬 管理员: C:\Windows\System32\cmd.exe - python

C:\Windows\System32>python Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information.

创建项目

点击 New Project 创建一个新的项目

Setup was successful

New to Python? Start with the <u>online tutorial</u> and <u>documentation</u>. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See <u>what's new</u> in this release, or find more info about <u>using</u> <u>Python on Windows</u>.



<u>C</u>lose



界面如下

🖻 New Project	— C	ı x
🍦 Pure Python	Location: C:\Users\Administrator\PycharmProjects\pythonProject2 1	
d Django		
📞 Flask	 Python Interpreter: New Virtualenv environment 	
 Google App Engine Pyramid Web2Py Scientific Angular CLI AngularJS Bootstrap HTML5 Boilerplate React React Native 	 New environment using Virtualenv 2 Location: C:\Users\Administrator\pythonProject2 Base interpreter: Python 3.9 E:\Python\python.exe Inherit global site-packages Make available to all projects Previously configured interpreter 3 Interpreter: Python 3.9 E:\Python\python.exe 	
	Create a main.py welcome script Create a Python script that provides an entry point to coding in PyCharm.	
	Create	Cancel

2、选择项目依赖的Python库,会在项目中创建一个 venv 的虚拟环境

3、关联本地的Python解释器,如果不想使用venv可以选择本地解释器的可执行文件(也就是我们上面 安装的内容)

建议选择 New environment using , 然后在 Base interpreter 中选择前面安装的Python解释器的路 径。如下图所示



这样做的好处:每个项目都是独立的空间,不会存在版本依赖冲突的问题,冲分发挥了虚拟环境的灵活性。

修改完毕之后,点击 Creat,创建项目。

这样就创建了一个最初始的空项目



界面介绍



从大的方向来看PyCharm分为 菜单栏区域 / 项目结构区域 / 代码区域 / 运行信息区



菜单栏

提示:菜单栏快捷键为Alt+首字母,比如File的快捷键Alt+F,Edit的快捷键Alt+E

1、File (文件)



- New ...: 新建一些中间件配置,如MySQL、MongoDB、DDL等以及相关驱动
- New Scratch File: 划痕文档,也称为临时文件,可以**创建各种类型**的文件进行临时处理,在 里面"打草稿",可运行并且可调试(非常棒的一个功能,在最近的版本才有的)
- Open: 打开项目目录
- Save as: 另存为
- Close Project:关闭项目并回答创建项目页面
- Rename Project: 给项目重命名
- Settings: 设置选项, 重点☆☆☆☆☆
- File Properties: 文件的相关属性,包括编码等
- Invalidate Caches /Restart..: 是缓存失效, 并重启

2、Edit (编辑)

<u>E</u> d	<u>E</u> dit <u>V</u> iew		<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	R <u>u</u> n	<u>T</u> ools	VCS
	Ur	ndo						
	<u>R</u> ε	edo					trl+Shif	
ж	Cu	ı <u>t</u>					Ctr	l+X
뎹	<u>C</u> c	ру					Ctr	l+C
	Co	opy Pat	th					
	Pa	ste						►
	D€	elete					De	ete
	<u>F</u> ir	nd						►
	Fir	nd Usa	ges					►
	Co	olumn s	Selection <u>M</u>	lode		Alt+9	Shift+Ins	sert
	Se	lect <u>A</u> l	I				Ctr	l+A
	Ac	ld Car	ets to Ends	of Sele	cted Lines	1	Alt+Shift	t+G
	Ex	tend S	election				Ctrl	+W
	Sh	rink Se	election			Ct	trl+Shift	+W
	То	oggle (Case			С	trl+Shift	t+U
	Jo	in Line				(Ctrl+Shif	ft+J
	<u>D</u> ı	uplicate	e Line				Ctr	+D
	Fil	l Parag	graph					
	Sc	ort Line	s					
	Re	everse	Lines					
	Tr	anspo	se					
		dent S	election					Tab
	Ur	nindent	t Line or Se	lection			Shift+1	Гаb
	Co	onvert	Indents					•
	M	acro <u>s</u>						►
		(code	XML/HTML	Special	Characters			
	Ec	lit as T	able					

Find:编辑窗口中用的最多的就是Find选项中的,例如Ctrl + F 文件内查找,Ctrl +Shift + F 项目中搜索,以及Ctrl + R 文件内替换,Ctrl+Shift+R 全文替换(慎用!)

windows下ctrl+shift+F快捷键如果无效,大概率是因为装了搜狗输入法,快捷键冲突导致的。只需要修改输入法中对应的快捷键即可,或者修改PyCharm的快捷键。

3、View (视图)

<u>V</u> ie	w	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	R <u>u</u> n
	Ιo	ol Window			►
	<u>A</u> p	pearance			►
	Qu	iic <u>k</u> Definiti	on	Ctrl+Shi	ft+I
	Qu	iick Type D	efinition		
	Qu	iick <u>D</u> ocum	entation	ı Ctr	l+Q
	<u>P</u> a	rameter In	fo	Ctr	rl+P
	Ту	p <u>e</u> Info		Ctrl+Shif	ft+P
	<u>C</u> o	ntext Info	Alt+Q		
	Re	ce <u>n</u> t Files		Ctr	rl+E
	Re	cently Cha	nged File	es	
	Re	cent Locati	ons	Ctrl+Shif	ft+E
	R <u>e</u>	cent Chang	ges	Alt+Shif	t+C
→ +	Co	mpare Wit	:h	Ctr	l+D
₽	Co	mpare wit	h Clip <u>b</u> o	ard	
	Qu	iick Switch	Scheme	Ct	rl+`
	Ac	tive Editor			►
	Bio	di Text Bas	e Directi	on	►
	Sc	ientific Mo	de		

- Tool windows: 工具窗口,如果主页面中某些窗口不小心关了,可以在这里面重新找到。
- Appearance: : 外观设置,除了基本的布局调整,最强大的莫过于这四种模式(在阅读代码的时候真的很爽!)

<u>T</u> ool \	Nindows	►			
<u>A</u> ppe	arance	Þ		Enter Presentation Mode	
Quic <u>k</u>	Definition	Ctrl+Shift+I		Enter Distraction Free Mode	
Quick	Type Definition			Enter Full Screen	
Quick	<u>D</u> ocumentation	Ctrl+Q		Enter Zen Mode	
<u>P</u> aran	neter Info	Ctrl+P	\checkmark	Main Menu	
Тур <u>е</u>	Info	Ctrl+Shift+P		<u>T</u> oolbar	
<u>C</u> onte	xt Info	Alt+Q	~	Na <u>v</u> igation Bar	
Recer	t Files	Ctrl+E	~	Tool Window Bars	
Recer	tly Changed File		~	<u>S</u> tatus Bar	
Recer	t Locations	Ctrl+Shift+E		Status Bar <u>W</u> idgets	►
R <u>e</u> cer	t Changes	Alt+Shift+C		Details in Tree View	
t Com	ara With	Ctrl+D	\checkmark	Members in Navigation Bar	

Enter/Exit Presentation Mode: 进入/退出 展示模式

Enter/Exit Distraction Free Mode: 进入/退出 免打扰模式

Enter/Exit Full Screen: 进入/退出 全屏模式

Enter/Exit Zen Mode: 进入/退出 禅模式 (一个终极模式,包含以上3种模式)

其次 Toolbar 也是一个不错的功能,开启之后,会在菜单栏有一个导航



- Recent Files:: 最近打开的文件, 快捷键 Ctrl + E
- Recent Locations:最近修改的内容
- Compare with: :比较文件之间的差异



- Compare with Cliboard: 与剪切板上的内容做比较
- 4、Code (编码)

<u>C</u> ode <u>R</u> efacto	r R <u>u</u> n	Tools	VCS	<u>W</u> indow	<u>H</u> elp	pyth
<u>O</u> verride Me <u>I</u> mplement I Generate	ethods Methods				Ctrl Ctr Alt+Ins	+O ·l+I sert
Code <u>C</u> omp	letion					•
Insert Live <u>T</u> <u>S</u> urround W Un <u>w</u> rap/Rer	emplate ith nove			Ctrl+S	Ctr Ctrl+Alt hift+Del	l+J t+T ete
Folding						•
Comment w Comment w <u>R</u> eformat Co Reformat Fil <u>A</u> uto-Indent Optimi <u>z</u> e Im Rearrange (Move Stater Move Stater Move Eleme	ith <u>L</u> ine (ith <u>B</u> lock ode e Lines ports Code ment Do ment <u>Up</u> ent Left ent Right	Comme : Comm <u>w</u> n	nt ient Ct: Ct:	(Ctrl+ Ctrl+Shi Ctrl+Shi rl+Alt+Shi rl+Alt+Shi	Ctrl+Shif Ctrl+Ah Alt+Shiff Ctrl+Ah Ctrl+Ah Ctrl+Ah ft+向下育 ft+向上育 ft+向左育 ft+向右省	- +/ . t+/ . t+L +L
Move Line [Move Line l)o <u>w</u> n Jp			Alt+Shi Alt+Shi	ft+向下育 ft+向上备	新头 新头
Inspect Cod Code Clean Silent Code	-r le up Cleanup					1
<u>R</u> un Inspect Configure C View <u>O</u> ffline <u>L</u> ocate Dupl	ion by N urrent Fi Inspecti icates	ame le Analy on Res	/sis ults	Ctrl+ Ctrl+,	-Alt+Shif Alt+Shift	t+I +H
Data Flow <u>t</u> o Data Flow <u>f</u> r	o Here om Here					
Update Cop	yright					

• Code Completion: 代码补全,不过可以进行全局设置,每次敲入字母时会自动提示进行补全 设置步骤如下: File -> settings



在同样的窗口,可以设置忽略大小写补全

🖻 Settings		
Q	Editor > General > Code Completion	Reset
Appearance & Behavior Scopes Notifications Quick Lists Path Variables Keymap V Editor V General Auto Import	 Match case: First letter only ● All letters R消沟选,可以忽略大小写自动提示 Automatically insert single suggestions for: Smart Type Completion Ctrl+Shift+空格 Sort suggestions alphabetically Show suggestions as you type Insert selected suggestion by pressing space, dot, or other context-dependent keys Show the documentation popup in 1000 ms H示弹窗时间,越小越快 Machine Learning-Assisted Completion 	
Appearance Breadcrumbs Code Completion Code Folding Console Editor Tabs Gutter Icons Postfix Completion > Smart Keys Code Editing Font > Color Scheme > Code Style Inspections	 Services aggression buside on medianic iterating () JavaScript Mark position changes in completion popup 14 JavaScript Only type-based completion Show fewer completion suggestions based on type information. May significantly improve performance. Suggest items with optional chaining for nullable types Expand method bodies in completion for overrides Completion of names 	
?	ок Cancel <u>А</u> р	ply

• Insert Live Template: 快速插入模板。

默认模板,路径 File -> Settings,也可以通过点击+号自己添加



以上面的 flask 下的 route 为例,写代码时,直接输入 route 就可以完成预先设置的模板内容了



• surround with: 将选择的代码进行包裹, 如 if/while/for/try..catch 包裹住。快捷键 Ctrl +Alt + T



- Reformat Code:格式化代码,快捷键 Ctrl +Alt +L
- Auto-Indent Lines: 自动缩进, 快捷键 Ctrl + Alt + I
- Move Statement/Line Down/Up: 向上向下移动, 快捷键 Ctrl +Shift + 向上箭头/向下箭头

5、Refactor (重构)



- Refactor This..: 重构当前
- Rename: 重命名, 快捷键 Shift + F6
- Move: 移动文件, 快捷键 F6
- Copy: 拷贝文件, 快捷键 F5
- Safe Detele: 安全删除, 快速删除py文件, 快捷键 Alt + Delete

6、Run

	R <u>u</u>		<u>T</u> ools	VCS	<u>W</u> indow	<u>H</u> elp	python-test - r	
		Ru	<u>ı</u> n 'main				Shift+F10	
1	ŧ	<u>D</u> €	ebug 'm	ain'			Shift+F9	
•	G	Ru	ın 'main	' with	Co <u>v</u> erage			
	¢	Pr	ofile				►	
		Ru	ın				Alt+Shift+F10	
	ŧ	De	ebug				Alt+Shift+F9	
	Ť,	At	tach to l	Proce	55		Ctrl+Alt+F5	
		Ed	lit Confi	gu <u>r</u> ati	ons			
		Ste	ор					
		Ste	op Back	grour	nd Process		Ctrl+Shift+F2	
		Sh	iow Run	ning L				
		De	ebuggin	g Acti	ons		۲	
		То	oggle Br	eakpo	oint		►	
	۲	Vie	ew Brea	<u>k</u> poin	ts		Ctrl+Shift+F8	
			st Histo					
	Ľ	Im	port Te	sts fro	om File			
		Sh	iow Co <u>v</u>	erage	Data		Ctrl+Alt+F6	

- Run 'xxx': 运行当前文件
- **Debug 'xxx':** 通过Debug模式运行该文件
- Run 'xxx' with Coverage: 以统计覆盖的形式运行当前文件
- Run ...: 选择文件运行
- Debug ...: 选择文件Debug运行
- Edit Configurations..: 编辑配置内容

🖺 Run/Debug Configurations					
+ - 🖬 🖌 🔺 🖛 🔩	<u>N</u> ame: main	Allow parallel r <u>u</u> n <u>S</u> tore as project file			
🗸 🍦 Python					
 main <i>F</i> Templates 	Configuration Logs				
	Script path: 🔻	E:\PyCharm_workspace\python-test\main.py 2 🗁			
	<u>P</u> arameters:				
	▼ Environment				
	<u>E</u> nvironment variables:	PYTHONUNBUFFERED=1			
	<u>Python interpreter:</u>	🗬 Project Default (Python 3.9 (python-test) (2)) E:\PyCh 🔻			
	Interpreter options:				
	<u>W</u> orking directory:	E:\PyCharm_workspace\python-test			
	✓ Add content roots to PYTHONPATH				
	✓ Add source roots to PYTHONPATH				
	▼ Execution				
	Emulate terminal in output console				
	Run with Python Console				
	Redirect input from:				
	<u>B</u> efore launch				
?		OK Cancel Apply			

- 1、Name:可以自己随意起名
- 2、Script Path: 项目的文件路径
- 3、Python interpreter: Python解释器的路径
- 4、Workding directory:项目路径

7、Tools (工具)

保存一些文件/项目模板。

8、VCS (版本控制)



- Enable Version Control Integration:选择相应的版本控制工具
- VCS Operation: 版本控制操作窗口
- Get from version Control...: 从版本控制中获取(比如从GitHub上导入项目时,可在这个 模块中完成)
- 9、Window (窗体)



- Store Current Layout as Default: 存储当前PyCharm布局
- Restore Default Layout: 窗口布局复位 (有时候窗口比较乱的时候,可以进行还原)
- 其他补充: 主要就是控制窗口布局, 以及tab显示的

10、Help



- Find Action: 通过键入快捷键唤出想要的功能(非常强大,适合键盘流)
- Keymap Reference: 查看快捷键清单 ()

PyCharm

Find any action inside the IDE	Ctrl + Shift + A
CREATE AND EDIT	
Show intention actions	Alt + Enter
Basic code completion	Ctrl + Space
Smart code completion	Ctrl + Shift + Space
Type name completion	Ctrl + Alt + Space
Complete statement	Ctrl + Shift + Enter
Parameter information / context info	Ctrl + P / Alt + Q
Quick definition	Ctrl + Shift + I
Quick / external documentation	Ctrl + Q / Shift + F1
Generate code	Alt + Insert
Override / implement members	Ctrl + O / Ctrl + I
Surround with	Ctrl + Alt + T
Comment with line comment	Ctrl+/
Extend / shrink selection	Ctrl + W / Ctrl + Shift + W
Optimize imports	Ctrl + Alt + O
Auto-indent lines	Ctrl + Alt + I
Cut / Copy / Paste	Ctrl + X / Ctrl + C / Ctrl + V
Copy document path	Ctrl + Shift + C
Paste from clipboard history	Ctrl + Shift + V
Duplicate current line or selection	Ctrl + D
Move line up / down	Ctrl + Shift + Up / Down
Delete line at caret	Ctrl + Y
Join / split line	Ctrl + Shift + J / Ctrl + Enter
Start new line	Shift + Enter
Toggle case	Ctrl + Shift + U
Expand / collapse code block	Ctrl + NumPad + / -
Expand / collapse all	Ctrl + Shift + NumPad + / -
Save all	Ctrl + S
VERSION CONTROL	
VCS operations popup	Alt+`
Commit	Ctrl+K
Update project	Ctrl+T
Recent changes	Alt + Shift + C
Revert	Ctrl + Alt + Z
Push	Ctrl + Shift + K
Next / previous change	Ctrl + Alt + Shift + Down / Ur
ronci protiono onningo	Sur , ut Sunt Down Of

Find action	Ctrl + Shift + A
Open a tool window	Alt + [0-9]
Synchronize	Ctrl + Alt + Y
Quick switch scheme	Ctrl + `
Settings	Ctrl + Alt + S
Jump to source / navigation bar	F4 / Alt + Home
Jump to last tool window	F12
Hide active / all tool windows	Shift + Esc / Ctrl + Shift + F12
Go to next / previous editor tab	Alt + Right / Alt + Left
Go to editor (from a tool window)	Esc
Close active tab / window	Ctrl + Shift + F4 / Ctrl + F4

FIND EVERYTHING	
Search everywhere	Double Shift
Find / replace	Ctrl + F / R
Find in path / Replace in path	Ctrl + Shift + F / R
Next / previous occurence	F3 / Shift + F3
Find word at caret	Ctrl + F3
Go to class / file	Ctrl + N / Ctrl + Shift + N
Go to file member	Ctrl + F12
Go to symbol	Ctrl + Alt + Shift + N
NAVIGATE FROM SYMBOLS	
Declaration	Ctrl + B
Type declaration (JavaScript only)	Ctrl + Shift + B

Super method	Ctrl + U
Implementation(s)	Ctrl + Alt + B
Find usages / Find usages in file	Alt + F7 / Ctrl + F7
Highlight usages in file	Ctrl + Shift + F7
Show usages	Ctrl + Alt + F7

REFACTOR AND CLEAN UP		_
Refactor this	Ctrl + Alt + Shift + T	
Copy/Move	F5/F6	
Safe delete	Alt + Delete	
Rename	Shift + F6	
Change signature	Ctrl + F6	
Inline	Ctrl + Alt + N	
Extract method	Ctrl + Alt + M	
Introduce variable/ parameter	Ctrl + Alt + V / P	
Introduce field / constant	Ctrl + Alt + F / C	
Reformat code	Ctrl + Alt + L	

ANALYZE	AND	EXPLOR	E

Show error description	Ctrl + F1
Next / previous highlighted error	F2 / Shift + F2
Run inspection by name	Ctrl + Alt + Shift + I
Type/ call hierarchy	Ctrl + H / Ctrl + Alt + H
NAVIGATE IN CONTEXT	
Select in	Alt + F1
Recently viewed / Recent locations	Ctrl + E / Ctrl + Shift + E
Last edit location	Ctrl + Shift + Back
Navigate back / forward	Ctrl + Alt + Left / Right
Go to previous / next method	Alt + Up / Down
Go to line / column	Ctrl + G
Go to code block end / start	Ctrl +] / [
Add to favorites	Alt + Shift + F
Toggle bookmark	F11
Toggle bookmark with mnemonic	Ctrl + F11
Go to numbered bookmark	Ctrl + [0-9]
Show bookmarks	Shift + F11
BUILD, RUN, AND DEBUG	
Run context configuration	Ctrl + Shift + F10
Run / debug selected configuration	Alt + Shift + F10 / F9
Run / debug current configuration	Shift+F10/F9
Step over / into	F8 / F7
Smart step into	Shift + F7

Step over / into	F8 / F7
Smart step into	Shift + F7
Step out	Shift + F8
Run to cursor / Force run to cursor	Alt + F9 / Ctrl + Alt + F9
Show execution point	Alt+F10
valuate expression	Alt+F8
Stop	Ctrl + F2
Stop background processes	Ctrl + Shift + F2
Resume program	F9
Foggle line breakpoint	Ctrl + F8
Toggle temporary line breakpoint	Ctrl + Alt + Shift + F8
dit / view breakpoint	Ctrl + Shift + E8

jetbrains.com/pycharm
jetbrains.com/help/pycharm
blog.jetbrains.com/pycharm
@nvcharm



- Tip of the Day: PyCharm的每日小技巧 •
- Edit Custom Properties: 在idea.properties中添加个人配置
- Edit Custom VM Options: 在pycharm64.exe.vmoptions中添加启动配置
- Register: 注册 .

Check for Update: 检查更新 •

PyCharm基础配置



PyCharm安装完毕之后,我们需要修改一些常用的配置让视觉上更加享受,比如修改PyCharm的主题以及字体显示呢

基础配置在 File - > Settings 中进行

修改主题

在 Settings 中 选择 Appearance & Behavior-> Appearance ,如下图所示

E Settings		
Q.	Appearance & Behavior > Appearance	
 Appearance & Behavior 	Theme: Darcula 示いていたので、 Sync with OS	
Appearance Menus and Toolbars > System Settings File Colors © Scopes © Notifications Quick Lists Path Variables Keymap > Editor Plugins • Pugins • > Version Control © > Project: python-test © > Build, Execution, Deployment > Languages & Frameworks > Tools	Intellul Light Size: 12 Darcula Accessil High contrast Support screen readers Requires restart Use contrast scrollbars Adjust colors for red-green vision deficiency How it works Requires restart. For protanopia and deuteranopia UI Options Image: Size: Smooth scrolling (?) Use smaller indents in trees Drag-and-drop with Alt pressed only Enable mnemonics in menu Merge main menu with window title Requires restart. Always show full path in window header Display icons in menu items Always show full path in window header Display icons in menu items Editor: DE: Subpixel Editor: Subpixel Editor: Subpixel	
?	OK Cancel Apply	

我个人习惯使用 Darcula 的黑色主题,除了默认的主题,我们也可以自己安装其他主题风格。

如何安装主题

主题可以从网站 http://www.themesmap.com/ 上进行下载。



下载自己喜欢的主题,然后在 Settings 界面中 Editor -> color Scheme 中进行安装,如下图





设置字体

仍然是在 Settings 界面,在 Editor->Font 中修改字体及大小

PC Settings		×	<
Q+	Editor > Font		
> Appearance & Behavior	Font:	字体 LetBrains Mono	
Keymap			
~ Editor	Size:		
> General	Line spacing:		
Con Editing			
Font	Fallback font:	<none> For symbols not supported by the main font</none>	
> Color Scheme			
> Code Style	Enable fon	it ligatures	
Inspections @	Restore Def		
File and Code Templates			
File Encodings 🛛 🖻	1 PyCha	arm is a full-featured IDE	

字体建议使用 JetBrains Mono,它是JetBrains 公司开发的一款开源字体,也被称为最适合程序员的字体。

代码模板

通常情况,我们创建.py 或者.html等文件时,需要指定一些信息,比如编码,开发人员信息,时间等..

这个时候就需要修改一下对应的文件模板。

```
在 File - > Settings 下, 找到 Editor - > File and Code Templates
```



Example

我在 Python Script 中进行如下配置

😫 Settings			×	
Q,	Editor $ ightarrow$ File and Code Templat	es l		(× Kamodel.py ×
> Appearance & Behavior	Scheme: Default			
Keymap Editor General Code Editing Fort Color Scheme Color Scheme Inspections	Files Includes Code + - • • - <th></th> <th></th> <th></th>			
File and Code Templates File Encodings @	الله Sass File الله SCSS File الله SCSS File الله Stylus File			
File Types > Copyright @ > Inlay Hints @ Dublicates	TypeScript File TypeScript JSX File tsconfigjson packagejson			
> Emmet Intentions Language Injections ତ > Proofreading	CoffeeScript Class	Apache Velocity template language is used		
Reader Mode @ TextMate Bundles TODO	Python Script Python Unit Test Setup Script Flask Main			

模板其他变量信息

\${PROJECT_NAME} - 当前Project名称;

\${NAME} - 在创建文件的对话框中指定的文件名;
\${USER} - 当前用户名;
\${DATE} - 当前系统日期;
\${TIME} - 当前系统时间;
\${YEAR} - 年;
\${MONTH} - 月;
\${DAY} - 日;
\${HOUR} - 小时;
\${MINUTE} - 分钟;
\${PRODUCT_NAME} - 创建文件的IDE名称;
\${MONTH_NAME_SHORT} - 英文月份缩写, 如: Jan, Feb, etc;
\${MONTH_NAME_FULL} - 英文月份全称, 如: January, February, etc:

解释器配置

执行Python代码必须要用到解释器,在环境配置章节讲到过下载解释器。如果本地安装了多个解释器版本,在不同的项目中需要进行切换时。

可以在 Fiel -> settings 中的 Project 项目名 - > Project Interpreter 中进行设置,如下图

<u>₽</u> Settings ×								
Q	Project: python-test > Python Inter	preter lle For current project						
> Appearance & Behavior	Python Interpreter: 🙀 Python 3.9 (python-test) (2) E\PyCharm_workspace\python-test\venv\Scripts\python.exe 🚬 🕇							
Keymap								
> Editor	Package		Latest version;此外切撞已安					
Plugins 🚺 🖻	certifi	2020.12.5	2020.12.5 过于自己的主义中国					
> Version Control	chardet	4.0.0	4.0.0					
🗸 Project: python-test 🛛 🖻	idna	2.10	▲ 3.1					
Python Interpreter 🛛 🖻	pip	21.0.1	21.0.1					
Project Structure @	requests	2.20.1 54 1 1	54.1.1					
> Build, Execution, Deployment	urllib3	1.26.3	1.26.3					
> Languages & Frameworks								
> Tools	÷ 0							
	+ − ▲ ⊙							
?			OK Cancel Apply					

点击小齿轮,点击`add,选择其他版本的解释器即可。如何



代码运行



编写好代码肯定是需要去运行的,在Pycharm中运行代码的方式有以下

1、右键Run运行

直接在要运行的py文件中右键,点击 Run 即可,或者使用快捷键 Ctrl + Shify + F10

樻 test.py 🗦				
1 ⊏f 2 3 4 ⊖	<pre>ior i in range(1₁10): for j in range(i₁10): print(str(i)+"*"+s print()</pre>	tr(j)+"="+str(i*j)+")
5	bi fuc()			
		Show Context Actions		
		<u>Paste</u> Copy / Paste Special Column Selection <u>M</u> ode	Ctrl+V ► Alt+Shift+Insert	
		Find <u>U</u> sages <u>R</u> efactor		
		Folding		
		Go To Generate	► Alt+Insert	
		<mark>₩u</mark> n 'test'	Ctrl+Shift+F10	
		Open In		
		Local <u>History</u> Bun File in Python Console		

2、通过导航栏的Run执行

PC	<u>F</u> ile <u>E</u> o	dit <u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	R <u>u</u> n	<u>T</u> ools	VCS	<u>W</u> indow	<u>H</u> elp	python-te	est -	test.py		—			×
P	ython-tes	st 🔪 💑 te	st.py										🍦 test 🔻	► ĕ	G			Q
g	📥 test.p	py ×											Edit Confi	iguratio	ns			
E.	1	for	i in r	ande	(1 10	۱.							H Save 'test	Config	juratio	on	~	~ A
			<u>т</u> ти і	ange	(T [*] TO).							🔚 rest-api#					tab
			for j	in r	ange (i <u>,</u> 10):						🍦 main					
			pr	int(str(i)+"*	*" + s1	tr(j)+"="	+str((i*j)+'		🍦 test					

如果之前执行过某个程序,在这里是可以看到的,选择要执行的程序,点击绿色的三角箭头即可运行。

如果是一个新的程序,则点击 Edit Configurations 进行配置,这个在前面的 菜单栏 Run 章节中有讲过

🖺 Run/Debug Configurations			
+ - 恒日 / 🎍 🗸 📭 » Add New Configuration 😤	<u>N</u> ame: test	Allow parallel r <u>u</u> n <u>S</u> tore as	project file 🔍
≊s ∪atabase Script ☑ Django Server ☑ Diango tests	Configuration Logs		
✓ ➡ Docker	Script path: 🔹	E:\PyCharm_workspace\python-test\test.py	
Docker Image Docker-compose	<u>P</u> arameters:		
Dockerfile	▼ Environment		
Flask server	<u>E</u> nvironment variables:	PYTHONUNBUFFERED=1	
& Grunt.js Gulp.js	Python	Roject Default (Python 3.9 (python-test) (2))	
HTTP Request	Interpreter options:		
a st	<u>W</u> orking directory:	E:\PyCharm_workspace\python-test	
◎ N V.js ● Pr_tracto的建一个Pytho	Add content roots to	pythonpath pythonpath 代码工程路径	
Python	▼ Execution		
 ☑ Python Debug Server ✓ I Python docs 	Emulate terminal in o	utput console	
	Run with Python Cons	ole	
	Redirect input from:		
	▼ <u>B</u> efore launch		
?		OK Cancel	

3、通过main启动

如果程序中有 main 函数, 会在左侧有个绿色的三角箭头, 点击选择 Run 项目名即可启动

tost p	$r \times$	
1. ▶ ६	if _	_name == 'main':
2 🗧		for i in range(1, 10):
3		for j in range(i, 10):
4		print(str(i) + "*" + str(j) + "=" + str(i * j) + "
5 ¢		print()

DeBug运行/调试

Bug 大家都知道是程序中的错误,导致程序不能正常运行。而 DeBug 的字面意思就是解决Bug。 DeBug 执行的方式也是有三种,与上面的 代码运行 章节类似,

1、右键DeBug

🐉 test.py 🗡			
<pre>1 ▶ oifname == 'main': 2 for i in range(1, 10): 3 for j in range(i, 1 4 print(str(i) + 5 print() 6</pre>	0): "*" + str(j) + "="	+ str(i * j) + " ", er	✓ (d="")
	 Show Context Actions Paste Copy / Paste Special Column Selection Mode Find Usages Refactor Folding Go To Generate Run 'test' Debug 'test' More Run/Debug Open In 	Alt+Enter Ctrl+V Alt+Shift+Insert Alt+F7 Alt+Insert	
	Local <u>H</u> istory	•	

2、导航栏DeBug



点击导航栏绿色的蜘蛛图标即可DeBug启动。

3、通过main Debug执行



如果程序有 main 函数入口,可以点击左侧的绿色小三角,然后选择上图标识的 Debug 项目名即可。

4、断点

如果Debug的程序没有断点,则跟正常的执行没有区别。

断点:一个断点标注一个代码行,当程序执行到断点所在行时,会被挂起。我们可以查看项目中各参数的值,运行结果等信息

如何打断点?

如下图所示,在红框标注的地方单击一下就可以打上断点,**可以标记多个断点**,或者快捷键 Ctr1 +F8 可快速在光标所在行打上断点。



如下图,我在第7行打了一个断点,通过 DeBug 执行程序。



通过 变量面板 的信息我们可以得到以下信息

- 执行到第6行, index 的值为 1,
- 当前循环为第一次循环, i的值为0
- checkcode 还未被赋值,依旧是空字符串"""
5、调试按钮

Deb	ug: 🤤 test	<u> </u>						
đ	Debugger	Console		¥	<u>₹↓</u>	± <u>↑</u>	×1	⊟
¥	Frames					Variab	es <mark>6</mark>	

调试面板如上图所示,每个按钮的意思如下,将鼠标悬停在对应的图标上,可查看快捷键

1、跳转到当前程序所执行的地方(快捷键Alt + F10);比如你打开了很多窗口,当前界面在别的代码页面,只需要按一下这个按钮就会回到程序所执行的地方。

2、依次往下执行代码(快捷键F8);不进入函数;顺序执行,如果某一行代码调用了别的函数,则不 会进入那个函数

3、顺序执行,会进入函数(快捷键F7);当某一行代码调用了其他函数,则会进入那个函数或者源码中。

4、顺序执行,进入自己写的代码中(快捷键Alt + Shift + F7);与上面的区域是只会进入自己写的代码中,不会进入源代码中

5、跳出当前函数(快捷键Shift + F8);配合编号 3、4 当程序运行到函数体中,可以迅速跳出当前函数,回到程序执行的地方

6、运行到光标位置并暂停(快捷键Alt + F9);此时你的光标在哪,程序就会运行到那个地方,并挂起 (暂停)

7、计算表达式(快捷键Alt + F8); 鼠标选中一段代码,点击 7 号按钮,或者右键选择 Evaluate Expression,然后在弹窗中,点击 Evaluate 就可以计算出当前选中表达式的值。

		• I	- 11	<u> </u>	ne == "main_	': #_四位変	I.F.F.F	学验证妈的生成			
				cheo	ckcode = "" # (
				for	<pre>i in range(4):</pre>						
	10				index = random.	randrange(0	4)	# 牛成一个0~3	<u>中的数 in</u>	dex:	
					print_test()		Show	Context Actions	Alt+	Enter	
					if index != i a	nd index +	κ cu <u>r</u> ΓΩΩοργ		(trl+∧	
					checkcode +	∙= chr(<mark>rand</mark> c	D <u>P</u> aste		c	trl+V	
					elif index + 1	== i:	Сору	/ Paste Special	Alto Chife a		
					checkcode +	-= chr(randc	Eind I	Isages	Ait+Sniπ+I	Insert	
					else:		<u>R</u> efac	tor			
					checkcode +	-= str(randc	Foldin	ng			
				prir	nt(checkcode)		<u>S</u> earc	h with Google			
			if na	name =:	="main") foriin ran ge	(4)					
							Gener	rate	Alt+I	lt+F8	
± <u>↑</u>	× _r I	▦					≚r Run to	o <u>C</u> ursor	A	lt+F9	
Variab								Run to Cur <u>s</u> or	Ctrl+A	lt+F9	
+	Control checkcode = (str) "						<u>J</u> ump 蒙 Add t	To Cursor to Watches			
	01 = {										

6、DeBug控制面板



- 1、重新DeBug启动当前程序
- 2、跳过当前断点,直接运行到下一断点处,快捷键:F9
- 3、停止并关闭当前DeBug程序
- 4、查看当前所有设置的断点
- 5、使所有断点都失效(此时断点由红色变为灰白色)

7、DeBug Console面板

在DeBug执行的时候,在 Console 中有个按 钮如下图蓝色箭头标注所示 Show DeBug Console ,可以 在右侧终端中输入变量的值或者表达式,按回车之后,会返回相应的结果。

1 import random	
2	
3 d <u>def print_test():</u>	
4 a = 1	
5 c pront(a)	-
7 ▶ <pre></pre>	
8 checkcode = "" # 保存验证码的变量 checkcode: ''	
9 👳 for 1 in rang (4): _i: 0	
10 ● index = random_randrange(0, 4) # 生成一个0~3中的数	
11 print_test()	
12 • if nde ! i i and index + 1 != i:	
13 // Cheby Code += chr(random.randint(97, 122)) # 生成a~z中的一个小写字母	
14 elt f i dex + 1 == i:	
if_name_== '_nain_	
Debug: 🔶 test ×	
🕆 Debugger 🗩 Consy 🖉 三 🛆 生 注 主 主 注 🗐	=
F = E:\PyCharm_wovkspace\python-test\venv\Scripts\python.exe "E:\PyCharm\PyCharm 2020.3.3\plugins\python\helpers\py	/dev\pydevd
▶ ↓ Convected to pydev debugger (build 203.7148.72)	
a 🖶 /> print_test()	
😴 🍳 >>> checkcode	
· · · · · · · · · · · · · · · · · · ·	
555 C	

8、设置断点属性(过滤)

在断点的小红点上右键,我们会发现这样一个界面

11	print_test()
test.py:11		+
✓ Enable <u>d</u>		ndo
✓ Suspend: ○ Condition:	<u>A</u> ll	Make De <u>f</u> ault Ido
		<u>_</u> • do
More (Ctrl+Shift+	F8)	Done

通过配置 Condition 中的内容,当程序符合 Conditioin 中的条件时,才会在当前断点暂停(挂起)





DeBug执行代码,效果如下



如果 DeBug 执行多线程的时候,可以指定线程名,专门针对某个线程进行DeBug.

9、Debug窗口悬浮

如果有多太显示器,或者想要将 Debug 窗口独立放大,可以点击相应标签,左键长按进行拖动



如果找不到相应窗口,或者布局比较乱,可以重置布局

bu	g: 👌 test	×							:	¢ –		
	Debugger	Console	Frames	Variables	Variables	Frames	🗠 🛨 🖽	<u>+</u> <u>1</u>		<u> </u>		_
	Frames				ariables						Frames	
											ariables	虫立放
											′ Viables	python-tes
											Restore Default Layout	
											om.py ×	

PyCharm版本控制



PyCharm集成了大部分流行的版本控制系统,如Git、Subversion、Mercurial、Perforce,以最 常用的Git为例

Pycharm关联git

在 File - > settings 中, 找到 Version Control -> Git 如下图所示,配置本地 git.exe 的路径



1、导入GitHub项目及配置

1、方法一

点击菜单栏中 VCS - > Get from Version Control



2、方法二

欢迎界面点击 Get from VCS , 如果在项目中 , 可以点击 File-> close project 退出至欢迎界面



然后配置相关内容即可,如下图所示。



配置完毕后,点击 Clone



导入完毕,进入项目。我们会发现菜单栏的 csv 不见了,取而代之的是 Git



- Commit: 将代码提交到本地仓库
- Push: 将本地仓库代码提交到远程Git仓库
- Pull: 将远程Git代码仓库同步到本地仓库
- Branches: 查看项目的所有分支
- New Branch: 创建新的分支
- Show Git log: 查看Git的历史操作

2、示例

在项目中的某个文件进行修改,然后点击 Git - > Commit,快捷键 Ctrl + к, 会出现如下界面



如果点击 Commit 需要再进行一次 Push , 或者直接点击 Commit and Push 操作完成。

Push Commits to Python

 master → origin : master
 ● It ∠ □
 王 ★

 潮試建文
 ● Et\PyCharm_workspace\Git_Python\Python 4 files
 ● Et\PyCharm_workspace\Git_Python\Python 4 files

 ● Et\PyCharm_workspace\Git_Python\Python 4 files
 ● Et\PyCharm_workspace\Git_Python\Python 4 files
 ● Et\PyCharm_workspace\Git_Python\Python 4 files

 ● Et\PyCharm_workspace\Git_Python\Python Program to Remove Punctuations from a String
 ● area of triangle
 ● Python Program to Remove Punctuations from a String

 ● Tandom password generator
 ● Tandom password generator
 ● It Push
 ● It Push

 ● Push tags:
 All
 ● Levent or All
 ● Levent or All

Push 路径在 Git-> Push , 快捷键 Ctrl + Shift + K

点击 Push 之后, 会弹出一个验证窗口



点击 Log In via GitHub 会在浏览器中弹出一个验证窗口。点击 Authorize in GitHub 即可。



Please continue only if this page is opened from a JetBrains IDE.



当前修改记录,可以在 Show Git Log 中查看



3、差异比较

方法一:

如果要比较某个 py 文件的差异,或者和历史版本进行比较,可以在文件上右键选择 Show Diff、 Compare with ...



方法二:

在修改的代码左侧有绿色的线条,表示这段代码被修改过



在绿色线条上右键,

				_
		return		
56				
		A <u>n</u> notate with Git Blame		
58		Set Bookmark	F11	
		Set Bookmark with Mnemonic	Ctrl+F11	
		Soft-Wrap		
		Configure Soft Wraps		
	\checkmark	Show Line Numbers		
	\checkmark	Show Indent Guides		
		Breadcrumbs	►	
		Configure Gutter Icons		

选择 Annotate with Git Blame , 可以看到所有历史记录。

MD	KEADIVIE.ma	осс.ру ∧			
ЧΨ	2011/10/2	nenca		יץ	THECH
41	2019/10/3	Mehta		se	ent_da [.]
42	2019/10/3	Mehta		pr	int("
43	2019/10/3	Mehta		cr	c = ci
44	2019/10/3	Mehta		re	emaind
45	2019/10/3	Mehta		pr	int("
46	2019/10/3	Mehta		pr	int("
47	2019/10/3	Mehta		ch	n = inp
48	2019/10/3	Mehta		if	ch =:
49	2019/10/3	Mehta			brea
50	2019/10/3	Mehta		el	se:
51	2019/10/3	Mehta			con [.]
52	Today		otations		1
53	Today	Show Diff	otations		
54	Today	View		►	st():
55	Today	Copy Revi	sion Numbe	er	turn
56		Annotate	Revision		
57		Annotate	Previous Rev	vision	110()
57		Options Chow in C		•	
28		Open on (it Log GitHub		155

然后选择想要查看的,右键 Show Diff 即可查看到差异变化。

PyCharm搜索 (文件、函数、内容)



Pycharm对搜索有很强大的支持,非常方便我们在项目中搜索某个关键词,或者函数等等

1、文件内检索

在文件内 Ctr1 + F , 如下图所示

📇 READN	1E.md × 🛃 crc.py ×							
~ 内容				0 results			≡ī ▼	
60			1		<u>↑</u> 1	1		
61		公十小军		lm====			配项 🔪	在范围内
62			开启正则	」 K G 閉C				
63					匹配项			
64								
65								
66								

2、文件内替换

快捷键 Ctr1 + R , 将搜索到的内容替换成目标内容。

📇 README.md 🛛	ᡖ crc.py 🗵			
Q- Enter		× ⊋ Cc_W *	2/2 🛧 🤟 🖬 🖾 🖬	≡ī ▼
Q- Hello 🛛 🎇			Re <u>p</u> lace Repl <u>a</u> ce All E	xc <u>l</u> ude
	<u>crc</u> = temp_a	lata		
	<pre>print("Quoti</pre>	.ent: ",/result	t, "remainder",	crc)
	return crc 🧃	呆留原有大小写	, 排除甘此匹而而	后扶協
26 # r e	eturning crc	value	用标来至他临现	四百姓

说明:	保留原有大小写,	比如原来的首字母是大写,	替换之后仍旧保留首字母大写。	如下所示
りしやし・				XH I //I

Q- Enter	× ⊋ Cc W .* 1/2 ↑ ↓ 🖬 🖾 🖬 ¥
Q- hello	X R AA Re <u>p</u> lace Repl <u>a</u> ce All Exc <u>l</u> ude
	IT (L +) < Len(data)):
22	temp_data.append(data[l + j])
	<u>crc</u> = temp_data
	print("Quotient: <mark>/</mark> , result, "remainder", crc)
	return crc
	# returning crc value
28	神林海球プリールの一本で見たりに
	while 1 >如: M首换成了Hello, 则不是hello
30	print(" <mark>Enter</mark> data: ")
	<pre>data = input() #can use it like int(input())</pre>
	print("Enter divisor")

3、项目中查找

1、Ctrl + Shift + F

该快捷键容易冲突,比如本地如果安装了搜狗输入法,可以先将对应的快捷键关闭再使用。

或者通过菜单栏进入,如下图

<u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode <u>R</u> efactor	R <u>u</u> n <u>T</u> ools Git	<u>W</u> indow <u>H</u> elp Python - crc.py - Administrator
r Undo Duplicate Line or Selection	Ctrl+Z	× × > + © 5 × Q
	Ctrl+Shift+Z	
X Cu	Ctrl+X	README.md × 🛛 🚜 crc.py × 🛔 Ohms_law.py × 🚜 c
Copy Path	¢ Culte	def crc_check(data, div):
Paste		l = len(div)
Path From Repository Root		
Delete 🔫	Delete	
<u>F</u> ind	•	Q Find Ctrl+F
Find Usages	►	R <u>R</u> eplace Ctrl+R
Column Selection Mode	Alt+Shift+Insert	Find <u>N</u> ext Occurrence F3
Select <u>A</u> ll	Ctrl+A	Find Pre <u>v</u> ious Occurrence Shift+F3
Add Carets to Ends of Selected Lines	Alt+Shift+G	Select All Occurrences Ctrl+Alt+Shift+J
Extend Selection	Ctrl+W	Add Selection for Next Occurrence Alt+J
Shrink Selection	Strl+Shift+W	Unselect Occurrence Alt+Shift+J
Toggle Case	Ctrl+ \\ift+U	In Selection Ctrl+Alt+E
Join Lines	Ctrl+Shitt	Next Occurrence of the Word at Caret Ctrl+F3
<u>D</u> uplicate Line	Ctrl+D	Find in Files Ctrl+Shift+F
Fill Paragraph		Repl <u>a</u> ce in Files Ctrl+Shift+R
Sort Lines		Search S <u>t</u> ructurally
Reverse Lines		Repla <u>c</u> e Structurally
Transpose		TOP 1 1N range
		🗘 🛛 temp_data[
Unindent Line or Selection	Shift+Tab	else:
Convert Indents	►	pecult append(
Macro <u>s</u>	•	resocc.appenu(
Encode XML/HTML Special Characters		for i in range
🆽 Edit as Table		🚊 temp_data[

界面如下,可以检索出项目中的所有复合条件的结果。



2、Shift + Shift

快捷键双击 shift,可以更精确的查找到类名/函数名/文件名

×	🖧 assembler.py 🗡		
	else:		
	print("Error: Not found ro return	egister")	
	# increment pointer for fetching pointer += 1	next token.	
		All Classes Files Symbols Actions Git	🗌 Include <u>n</u> on-project items 🏾 🍸 🔲
		Q. assembl	
def	<pre>registerLabels():</pre>	📸 <mark>assembl</mark> er Assembler	
		🖿 Assembler Assembler	
	This function search for labels /		

勾选 Include non-project items,可以搜索项目代码之外的内容,比如引入的库

		CLASS EFFORTUSER		
龄 pythonw.pdb				Git ✓ Include <u>n</u> on-project items ▼ 🗔
by thonw_d.exe				Tuna (to see command
pythonw_d.pdb	10 1	library: str = .		Type / to see command
vcruntime140.dll			💑 dbm E:\Python\Lib\dbm	
site performe		# Actual typenam	😢 dbm datetime	
I Tupeshed Stubs			W DBL_MAX _testcapi	
2and3 library root		class _dbm:	😢 DBL_MIN _testcapi	
2and3 library root	14	<pre>def close(se</pre>	🚦 de_be.msg E:\Python\tcl\tcl8.6\msgs	
a library root		def getite		微信兰注八个县·Duthon 据安去栏
> 🖿 asyncio		doftto		
> 🖿 collections		uersetite	指 test_bdist_msi distutils.tests	
> 🖿 concurrent		def delite	💑 bdist msi distutils.command	获取最新PyCharm操作手册
Y 🖿 dbm		def len (Logist msi distutils.command	_ ,
💑 _initpyi		dof dol (Addist msi setuntools, distutils command	
🖧 dumb.pyi			IDPOpenDPPequetEventMap built in dom	
🛃 gnu.pyi		defenter_		
ndbm.pyi		defexit	TDBOpenDBRequestEventMap built-in, webwoi	rker
> Constitutions		colf ov	misc pipvendor.distlibbackport	
			(f) _days_before_month(year, month) datetime	
> Dimportlib) \rightarrow None: .	m test_dotted_but_module_not_loaded(self) Patch	Test in unittest.test.testmock.testpatch
> ison		@overload	① _DAYS_BEFORE_MONTH _CalendarOffset in zo	neinfozoneinfo
> 🖿 multiprocessing		def det(self	SALERT_DESCRIPTION_BAD_RECORD_MAC _ssi	
> 🗖 os			ALERT_DESCRIPTION_BAD_RECORD_MAC ssl	
> 🖿 unittest		loverload	mtest_discovery_from_dotted_path_builtin_modu	les(self) TestDiscovery in unittest.test.test_discovery
> 🖿 urllib		def get(self	C _gdbm dbm.gnu	
> 🖿 venv		def kevs(sel	指 ndbm dbm	
> 🖿 xmlrpc		dof ootdofou	💑 ndbm dbm	
💑 _bootlocale.pyi			http://www.com/arm.and/arm	rthon\helpers\typeshed\stdlih\3\dhm
🍖 _compat_pickle.pyi		# Don't exis	dbm E\Python\Lib\dbm	and the state of the second
compression.pyi		new: Non	models EV. Dy Charmy Dy Charm 2020 2 2) aluging	\nuthen\halpers\nuthen_skalatens\diange\dh\medals
[о́_ітр.руі		init No	models chypenarm(Pycharm 2020.3.3)plugins	Apychon (neipers/pychon-skeletons/ujango/ub/models
importlib_modulespec.pyi		: NO		

Classes 搜索并跳转特定的类,快捷键 Ctrl + N。

All	Classes	Files	Symbols	Actions	Git	<u>P</u> roject Files ~	T	
	୶							
- ito o	neeven Pyth	on						
🔥 o:	sinfo Python							
💑 O	hms_law Ele							
💑 O	rganise Pytł							

Files 可以快速跳转到文件,比如我输入 c ,就会检索出所有与C相关的文件,快捷键 Ctrl + Shift + N



另外 Symbols 的模糊查询也非常实用。当记不清完整的关键词时,可以进行模糊搜索。快捷键 Ctrl + Alt + Shift + N 如下图所示



4、当前类、方法、属性列表

快捷键 Ctr1 + F12 , 可以把当前文件中的所有属性、类、方法都显示出来

	assembler.py	
Inherited members (Ctrl+F12)		\$
😢 eax		
€ (m		
🕐 ecx		
😢 edx		
 GuvalidSyntax(Exception) 		
🛅init('0		
🕑 jumps		
🥑 lines		
(f) loadFile(fileName)		
(f) main()		
(f) parser()		
(f) resetInterpreter()		
v returnStack		
(f) scan()		_
f scanner(string)		
V stack		
V Co Token		
t of t		
fj token		
W tokens		

直接输入关键字,就可以检索出符合条件的属性/类/方法,并且可以定位到相关位置



同样的功能,也可以通过 Alt + 7 来实现,如下图所示。同样也是直接直接输入关键字进行搜索。



5、查看最近修改的文件

快捷键 ctr1 + e 。可以查看最近修改的文件



6、查看函数的调用关系

当一个函数不知道被哪些地方调用的时候,可以通过快捷键 Alt + F7 进行查看,效果如下图



常用小技巧



1、快速导入依赖

平时我们导入第三方库时, 会回到文件的开头写上 import xxx 。

其实可以通过快捷键 Alt + Enter 进行快速导入,但是前提是你本地已经安装过这个库。

键盘敲入 Alt + Enter 或者点击下图中的红色小灯泡,选择 import this name 然后选择对应的包即可





2、快速删除冗余依赖

有时候项目中有些无用的依赖,会呈灰色现实。

pimport requests
└import math 多余的依赖
<pre>requests.get('https://www.baidu.com')</pre>

这个时候不需要一个个手动删除了,只需要点击 Code - > Optuimize Imports 即可,当然也可以通过 快捷键 Ctrl + Alt + 0 来快速清除

<u>N</u> avigate	<u>C</u> ode <u>R</u> efactor R <u>u</u> n <u>T</u> ools (Git <u>W</u> indow <u>H</u> elp Pyth	ion - Test.p
🍦 Pyth 🛃 Test.p	<u>O</u> verride Methods Implement Methods Generate	Ctrl+O Ctrl+I Alt+Insert	5 ¥
Charm_wo	Code <u>C</u> ompletion	► (+-!+1	est.py× auests
r	Sur ound With	Ctrl+Alt+T	ith
les pler py	Un <u>w</u> rap/Remove Folding	Ctrl+Shift+Delete	
txt	Comr ent with <u>L</u> ine Comment	t Ctrl+/	get('ŀ
1E.md	Comment with <u>B</u> lock Comme <u>R</u> eforn at Code	nt Ctrl+Shift+/ Ctrl+Alt+L	
vnloader d Schedule	Reform t File	Ctrl+Alt+Shift+L	
y '	<u>Auto-Inc</u> ant Lines Optimi <u>z</u> e Imports	Ctrl+Alt+O	
lar.py agment_s	Rearrange Code Move Statement Down	Ctrl+Shift+向下箭头	

3、编辑器窗口管理

前面提到在菜单栏中可以操作窗口

it	<u>W</u> ir	ndow	<u>H</u> elp	Python -	· Test.	py - Admin	stra	tor		
Git		Store Rest	Current L	.a <u>y</u> out as	Defau	lt shift+E12				
12		Active	1 ol Wir	ndow)				
1		Editor Notific Backo	<u>l</u> abs ations round Ta	neke		1		Select Ne <u>x</u> t Tab Select Previous Tab Pin Active Tab	Alt+向左箭头 Alt+向左箭头	
2 3		Next P Previo	Project W	'indow -t Windov		· Ctrl+Alt+] Ctrl+Alt+[- <u>C</u> lose Close Other Tabs	Ctrl+F4	
4	✓	Pythor	n 1			CUITAT		Close <u>A</u> ll Tabs		
								Close Tabs to the Right		
								Close All Read-Only Close <u>U</u> nmodified Tabs		
								Reopen Closed Tab Split Right		
							₿	Split and Move Right Split Down Split and Move Down		

但是这样操作很麻烦,直接在代码的选项卡上右键就可以操作。



重点介绍几个

- Close:关闭当前窗口,
- Close Other Tabs: 关闭当前窗口之外的所有窗口
- Close Tabs to the Lef: 关闭当前窗口左侧的所有窗口
- Close Tabs to the Right: 关闭当前窗口右侧的所有窗口
- close unmodified tabs: 关闭没有修改过的窗口
- Copy Path...:复制文件路径,可以选择文件的绝对路径/文件名 复制
- Split Right: j将当前选项卡分割到右侧(并不会从当前窗口移除),可以分割成两个或多个窗口



• Split and Move Right: 将当前选项卡窗口移除,并分割到右侧,也可以鼠标长按选项卡进行拖动实现

	▲1 ▲ 258 <i>☆</i> 7 ^ ~
	=
	_
	=
	5
	E
	=
variables = {}	-
	=
returnstack = []	=
	=
A Sample Exception class	
	=
	Ē
	_
	± -
	÷

4、任意代码块折叠

通常情况下,遇到代码块,PyCharm会有折叠的标识



如果对于任意代码块改如何折叠呢?比如我想要折叠12~18行的内容。

只需要选中相关代码在菜单栏 Code - > Folding - > Fold Selection/Remove region 中操作即可。

5、设置显示行数和分隔符

显示行数,可以直接在编辑侧边栏右键,选择 Show Line Numbers



或者在菜单栏中进行设置, File -> settings 中选择 Editor-> General-> Appearance, 如下图标 识, 勾选 Show line numbers。



上图如果勾选了 Show method separators 则每个方法会有分割线提示,效果如下:

9 🖓	def	test(): pass	
	def	test1(): pass	
	def	test2(): pass	

6、多行标签显示

默认情况,打开多个文件时,标签是一行显示的,如下图



当打开的选项卡过多时,多余的会自动被折叠,不方便查看。

解决:在File->settings中,选择Editor->Editor Tabs 取消勾选Show tabs in one row即可

PC Settings							
	Editor > General > Editor Tabs						
> Appearance & Behavior	Appearance						
Keymap	Tab placement: Top 取消勾选						
✓ Editor	Show tabs in one row						
✓ General							
Auto Import	✓ Hide tabs if there is no space						
Appearance	Show pinned tabs in a separate row						
Breadcrumbs	Use small font for labels						
Code Completion	✓ Show file icon						
Code Folding	✓ Show file extension						
Console	✓ Show directory for non-unique file names						
	─ Mark modified (*)						
Gutter Icons							
Postfix Completion	Show full path on mouse hover						
> Smart Keys	Close button position: Right 🔻						
Code Editing							
Ft	T-h O-d-r						

设置完毕后,效果如下:

, Te								
×	💑 Flappy Bird.py 🛛 🛛	👸 main.py 🛛 🗙	🛃 Test.py 🛛 🛛	ᡖ schedular.py 🛛 🛛	🛔 mbox-short.txt 🛛	🐇 dict1.py 🛛 🕹	README.md	
an	🖧 eight_p			🛃 ntpath.pyi		🖧 Merge_linked_	list.py	Datab

7、快速补全:,切换下一行

通过使用快捷键 Ctrl + Shift + Enter 快速补全:,并切换下一行。不光是补全:,也可以进行收 尾



8、粘贴板历史

CV大法好, Ctr1 + C/V 大家也都会用。但是如果想看历史复制的记录该怎么办呢? 很简单,使用快捷键 Ctr1 + Shift +V,然后在弹窗中,就可以看到历史复制记录啦



9、批量重命名

有时候粗心大意,发现一个变量名写错了,等到发现的时候已经被引用了一大堆。一个个的去修改容易 出错不说还很麻烦。

这个时候只需要将光标放到变量上,按下快捷键 Ctr1 + F6, 就可以批量重命名了。



10、PyCharm的书签功能

PyCharm也可以像浏览器那样,对关键代码打上标签。对某段代码进行标记,后面需要查阅的时候就会方便很多。

打书签,快捷键 Ctrl +F11,选择对应的数字键,可以通过 Ctrl + 数字键 快速定位到书签处



书签列表,快捷键 Shift + F11,展示所有书签

	/	- †		•							Bookmark				
	ť	for l	ine in	n fh:											#!/usr/bin/env python
	ť	for i	, row	in e	umera	te(tal	ole.rov	vs): ma							
3.	i A	table	e = do	ocun	ent.tab	les[0] main	.py:84							counts=dict()
	R R	df_te	ext_ch	nild3	main.py	r:104	(Extrac	t-Tabl							mails=list()
1.	ť	, prin	t(a) ⊺												<pre>fname=input('Enter file r</pre>
															fh_open(fname)
												7	1	 Image: A set of the set of the	for line in fh:
															if not line.startswit
															continue
															if line.startswith('F
															continue
															id=line.split()
												13			mail-id[1]

11、快速定位位置

如何快速定位代码在 Project 中的位置,有两种方法。

直接点击项目结构区的小圆圈,如下图

		py on a	27						
Pr	roject 🔻	€	ŝ	Ŧ	\$ —	樻 Flapp	y Bird.py $ imes$	🛃 main.py 🛛	i 🍖
× 🖿	Python E:\PyCharm_workspace\Git_Python\							s App(Tk	. s
>	🖿 .github								/ _
~	🖿 Assembler								
	> 🖿 examples							Classe p	rin
	🛃 assembler.py								
	🗧 GUIDE.txt								
	🚜 README.md								
	🖧 Test.py							# Variáv	eis
>	🖿 async_downloader								~~~~~
~	Automated Scheduled Call Reminders							раскдг	oun
	🖧 caller.py							bestSc	ore
	💑 schedular.py							hind d	<u></u>
>	🖿 bank_managment_system								630
>	BlackJack_game							button	s =
>	🖿 brickout-game					25		nlavin	a =

第二种方法,通过快捷键 Alt + F1 , 选择 Project New



12、自动生成常用语法

假设有个变量 a。

生成if判断可以直接 a.if

类似的还有 a.print 、 a.while



必备插件



以下所有插件在公众号: Python极客专栏, 后台回复【插件】

1、Key Promoter X (快捷键)

用来提示快捷键的插件,帮助我们尽可能的摆脱鼠标操作

在用鼠标进行操作是,插件会自动提示相应功能对应的快捷键。



2、Translation (翻译)

一款翻译插件,支持谷歌/有道/百度翻译。支持中英文互译。再也不用担心看不懂英文,和命名变量/函数了。



3、CodeGlance (缩略图)

对于较长的代码文件,有这个缩略图插件会非常方便。插件生效后效果如下



4、Json Parser (Json格式化)

一个支持在PyCharm内部进行Json验证和格式化的轻量级插件,省去了来回切换浏览器格式化校验的麻烦。



如果找不到入口可以在左下角如下图所示

	> 🖿 wsairef	7						
-0-	Commit Alt+0							
8	Database							
1	Event Log							
*	Favorites Alt+2	.ну .foo.py						
벽	Git Alt+9	ort.py						
Js	on Parser	ale.py						
*	Ke Promoter X	process.py						
θ	Problems Alt+6	pickle.py						
	Poject Alt+1	ssion.py						
÷	F /thon Console	pase.py						
	: ciView	port.py						
.:	Structure Alt+7	al.pv						
Z	Terminal Alt+F12							
≔	торо	ns.py						
Ш		.py		whwswswws				
	In Like it: Like this plugin? Donate or Give it a star Json Parser and spread the word (a minute ago)							
5、activate-power-mode (花里胡哨)

安装完毕之后,每次敲击字符都有动画效果。可以把combo/shake 取消勾选,这样效果会好一些



6、ideaVim (vim编辑器)

安装完毕之后,在 Tools -> Vim Emulate 中进行开启和关闭



这款插件如果对Vim不熟悉的刚开始用回很不习惯,但是一旦熟悉之后,可以完全脱离鼠标进行编码了。不光效率看起来也很装X。

- ESC: 切换Vim模式
- i: 切换到编辑模式

方向键: k 上 / j 下 / h 左 / 1 右

词组正向跳转: W

词组反向跳转: B

正向跳转到指定字符: f+字符,例如 fh 正向跳转到h字符

反向跳转到指定字符: F+字符,例如 Fh 反向跳转到h字符

跳转文件头部: gg

跳转至文件结尾: G

跳转到指定行的开头: 行号+G, 例如 10G

复制当前行: yy

粘贴: p

删除当前行: dd

删除选中区域: d

撤销: u

正向按字符删除: X

从光标处删除至指定字符: df指定字符

7、Regex Tester in PyCharm (正则)

用来测试正则表达式的插件,安装完毕之后在左下角的小矩形框中可以找到



界面如下,上面输入的是正则表达式,下面输入需要查找的字符串。

Regular expression												
\d+												
Find	Match		Replace									
hello123	hello1234											
🗗 Git	≡ todo	\rm 🛛 Prob	blems 🏼 🌆	Regex Tester	🗷 Terminal	🖨 Python Console						

8、Rainbow Brackets (彩虹色)

可以让代码块之间很清晰的显示出各种颜色。匹配的括号是相同的颜色,并且实现选中代码高亮显示。可以对代码的排错有帮助。



9、Material Theme (炫酷主题)

如果大家不喜欢扁平化的主题风格,可以尝试一下这种主题,简洁大方。安装之后PyCharm的界面会变成这样



安装之前是这样的,大家可以比较一下。

PC	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	R <u>u</u> n			<u>W</u> in	dow <u>H</u> e	elp							
⊨	⊟	G	← →	🍦 Test	t 🔻 🕨	► ĕ G							05/1/	Q 🛪					
Ру	thon) Bla	ckJack_	game 👌 🛃	blackjad	ck.py													
t	📄 Pi	roject					⊕ Ξ	Ŧ	\$		ᡖ blacl		Class		Carly	E4			
roje	~	Pvth	on E:\F							17	1	-	<u>Close</u> Close Other T		Ctrl+	F 4			
		.g	jithub										Close <u>O</u> llier i						
		A	ssemble	er									Close Unmod	, ified Tabe			GAME	OF FO	RTU
		🖿 a	sync_da	ownloader									Come Dath	incu rabs					
<u>8</u>		A	utomate	ed Schedul	ed Call I	Reminders													
•		b b	ank_ma	inagment_s	ystem								Split Right					~ ~	10
		B	lackJacl	k_game									Split Down				0, /,	o, y,	ΤC
		b	rickout-	-game									P <u>i</u> n Tab		-1.16				
			hecker_	_game_by_c	lz J								Move Tab to	New Windo	ow Shift+	F4			
			olors	beDownioa	der								Configure Edi	tor labs					
			ommuni	ication									Add to F <u>a</u> vori	tes					
			ompres	ssion Analy									Add All To Fa	vorites		Þ			
			ountMil	llionCharac	ters-Var	iations							Kename File				*****	*****	***
		E C	RC										R <u>u</u> n 'blackjacl	k'	Ctrl+Shift+F	10			
		E E	ight_Puz	zzle_Solver									<u>D</u> ebug 'blackj	jack'					100
		E E	ectroni	cs_Algorith									More Run/De	bug		•	ге нег	e io	ACC
		🖿 e	mail id (dictionary									Open In			•			
		E E	xtract-T	able-from-	pdf-txt-	docx							Local <u>H</u> istory				une')		
		E Fa	Face and eye Recognition										<u>G</u> it						
			race_Mask_detection (haarcascade)						Ç,) Create Gist					Ano				
		fl.	flappy Bird - created with tkinter										princt Lets check how Lucky you are						
		fle	oodfill	a_p/gamo									time.sle	eep(2)					
		🖿 g	∎ noodiini ■ game_of_life ■ Google_Image_Downloader										<pre>print('Loading') time sleen(2)</pre>						
		🖿 G																	
		🖿 in	nage2p	odf									CTINE. 3 CC	seb(z)					
		🖿 In	nageDo	ownloader															
		in in											print('S	Still	Loading-		')		
			ista_ima	age_saving									time.sle	eep(2)					
				onitering									npint('		Ane Sti	111	Here	Not G	one
		J.	istDialS	crapperGU									print(Une
		ыĸ	Koch Curve										time.sle						
		🔲 libs											d_cards = [] # Initialising dealer's						
		🖿 lir	near-alg	gebra-pyth	on								p_cards	= []	# Initi	ial	.isina	plave	r's
		🖿 na	asa_apo	od_with_req	quests								time cla	(2)					
		n 🖿	odepad										LINE.SL	seh(z)					
		n	Inotepad other_pepole Decommend Concentral										<mark>⊽while len(d_cards) != 2:</mark>						
		0											random.shuffle <mark>(</mark> deck)						
		PDF											<pre>d_cards.append(deck.pop()) if lon(d_cards) == 2;</pre>						
			primelib																
		P P	vthon c	hatting ap	olication								IT	cen(a_	carus) =		2:		
			R_code	generator										print	('The ca	ard	ls deal		S a
		🖿 re	earrang	je-files															
		R	ecursio	n Visulaizei									# Displa	aving	the Play			she	
•		🖿 SI	MS sen	der									- DISPLO				s cai	45	
	>	S S	nake wa	ater gun							42	5	While le	en(p_c	ards) !=	= 2			

10、Statistic (代码统计)

可以统计当前项目中代码的行数和大小等信息

插件安装完毕, 在 View -> Tool Windows -> Statistic 中开启

t	⊻iew	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	R <u>u</u> n	I	ools	Git	<u>W</u> ind	ow	<u>H</u> elp	
	Īo	ol Window			•	÷	Com	mit		Alt	+0	
	Ap	pearance			►		Proje	ect		Alt	+1 -	
.е	Qı	uic <u>k</u> Definitio	on	Ctrl+Shi	ft+I	*	Favo	rites		Alt	+2	
,t	Qı	uick Type 🖸	efinitior			۹	Find			Alt	+3 ^a	
h	Re	ce <u>n</u> t Files		Cti	rl+E	Þ	Run			Alt-	+4	
g	Re	cently Chan	iged Fil	es		*	Debu	ıg		Alt-	+5	
AS ac	Re	cent Locatio	ons	Ctrl+Shi	ft+E	θ	Prob	lems		Alt	+6	
as Δi	Re	cent Chang	es	Alt+Shif	t+C		Struc	ture		Alt-	+7	
bε	🔂 Co	mpare with	Clip <u>b</u> o	ard		0	Servi	ces		Alt	+8	
BI	Qı	uick Switch S	Scheme	Ct	rl+`	ч 8	Git	L		Alt-	19	
br	Bio	di Text Base	Direct	ion	►		Data	base				
Cł	Sc	ientific Mod					Hiers	i Log archv				
Cli	тоціці		10 101				Inspe	ection	n Resu	ltc		
Ca	lors					Key Promoter X						
co	mmun	ication				s Learn						
Ca	mpres	ssion_Analys	515 \/			🍨 Python Console						
	untiviii	llionCharact	ers-vai	lations		(.*)	Rege	x Tes	ster			
Fie	uht Puz	zzle Solver					SciVi	ew				
Ele	ctroni	cs Algorithr	ns			þ	Statis	stic				
en	nail id	dictionary				Ł	Term	ninal	A	lt+F	12	
Ex	tract-T	able-from-	pdf-txt-	docx		≔	OD	0				
Fa	ce and	l eye Recog	nition			I	Word	d Boo	ok			

统计效果图如下

C Refresh C Refresh on selection @ Settings												
Source File 🔺												
🗅 _initpy												
🗅 _initpy												
05_mixed_sorting.py												
🗅 addition.py												
🗅 aj.py												
alexa_news_headlines.py												
🗅 armstrongnumber.py												
A solution to project euler problem 3.py												
🗅 assembler.py												
🗅 async_downloader.py												
🗅 audiobook_gen.py												
avg_xdspam_confidence.py												
🗅 backend.py												
Background.py												
backup_automater_services.py												
🗅 balance_parenthesis.py												
🗅 basic.py												
D Total:		8 16592	8 61%	ି ⊶	₿ 16%	8 6411	₿ 23%					

结尾

虽说学Python没有规定一定要使用PyCharm,但是不得不说PyCharm确实好用。希望能够帮助到那些刚开始使用PyCharm又束手无策的同学。

本手册有很多不足之处,欢迎斧正。

后续也会不断的进行完善和迭代,将第一时间发布在公众号上。欢迎关注公众号: Python极客专栏

如果本手册能够帮助你快速入手PyCharm,请作者喝杯咖啡吧~后续会继续完善更新! 一起加油!

