Pyflink 的安装和 windows 开发环境配置

作者：Sakura6868
原文链接：https://ld246.com/article/1631936123228
来源网站：链滴

在分布式大数据计算引擎这个领域，现在最常用的Apache Spark 早已支持 python语言编写，而且对**L（机器学习）和DM（数据挖掘）也都有api的支持，而作为第三代计算引擎的flink， 从 1.9.0 版开始增加了对 Python 的支持** （PyFlink），在Flink 1.10 中，PyFlink 添加了对 **Python UDFs** 自定义函数） 的支持，可以在 **Table API/SQL** 中注册并使用自定义函数，而从Flink 1.11开始还支在 Windows 上本地运行 PyFlink 作业，所以可以在 Windows 上开发和调试 PyFlink 作业

Pyflink的安装

Pyflink的安装十分之简单

- 首先先看看自己系统的python版本（PyFlink 需要 Python 版本（3.6、3.7 或 3.8））

```
$ python3 --version
# the version printed here must be 3.6, 3.7 or 3.8
```

- 环境配置

由于系统可能包含多个 Python 版本，因此也包含多个 Python 二进制可执行文件。运行以下 ls命令找出系统中可用的 Python 二进制可执行文件：

```
$ ls /usr/bin/python*
```

- 选择软链接　python指向您的python3解释器

```
ln -s /usr/bin/python3 python
```

- 安装Pyflink

由于pyflink还在持续火热的更新之中每个版本的变化较大，所以无脑安装最新版就行（更新此文章时新版为apache-flink1.13.2）

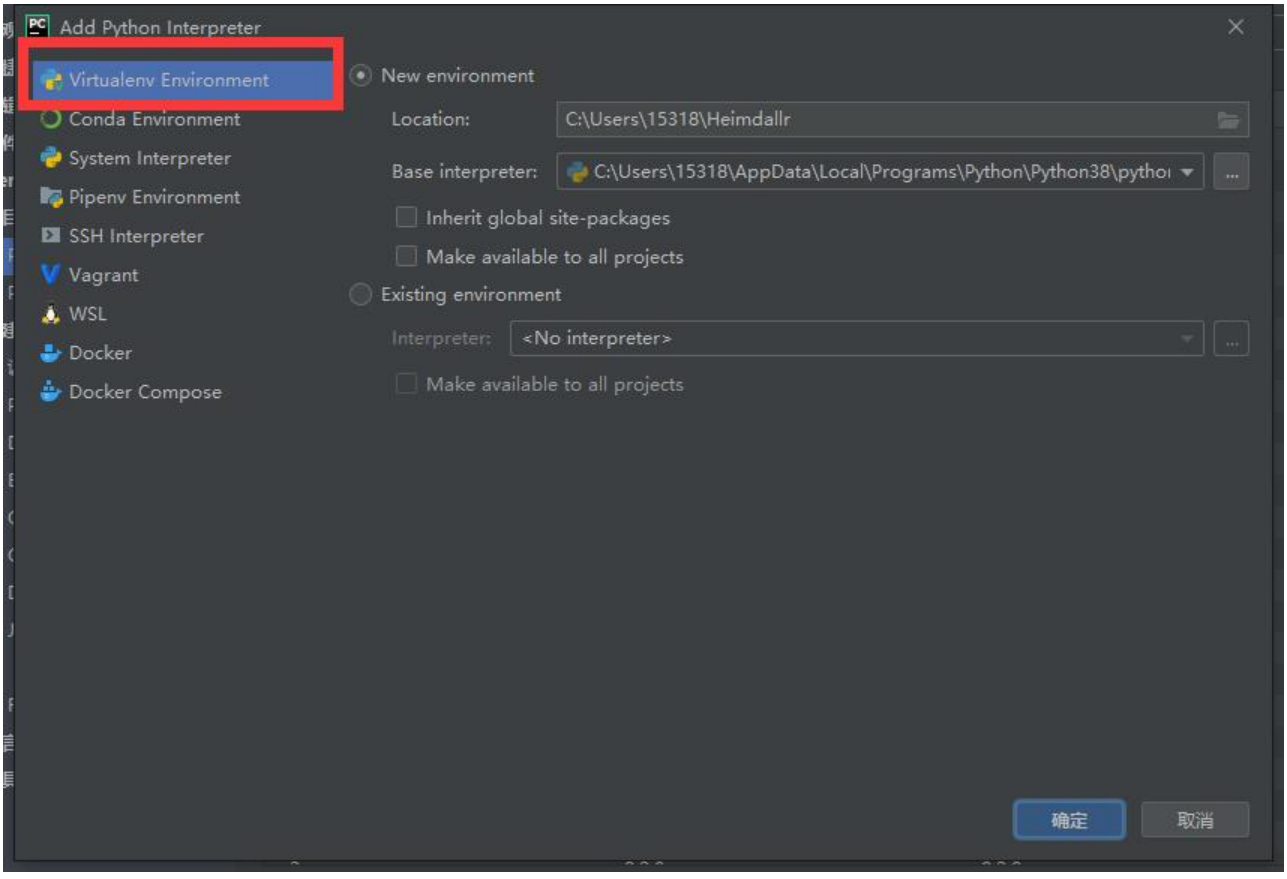
```
$ python3 -m pip install apache-flink
```

windows开发环境配置

这里我们选择 Pycharm IDE 来进行windows Pyflink 开发

- 首先配置Pytthon 虚拟环境

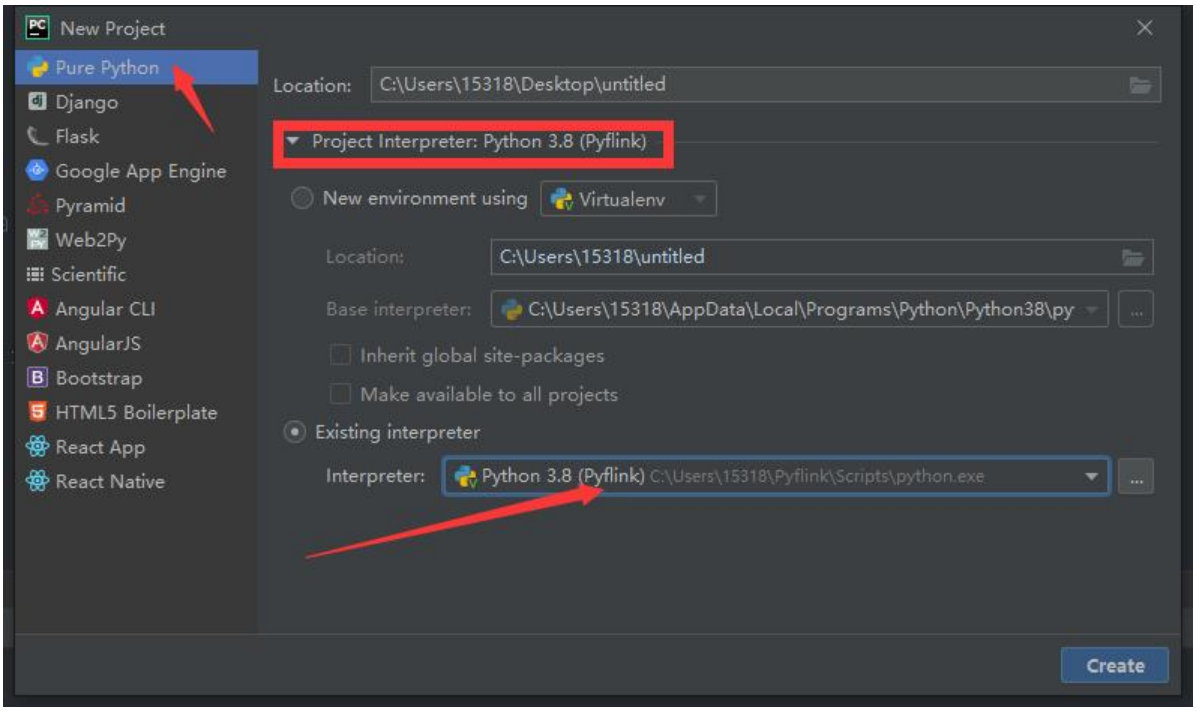
配置路径PyCharm -> Preferences -> Project Interpreter



记住选择的python版本一定要是3.6、3.7 或 3.8

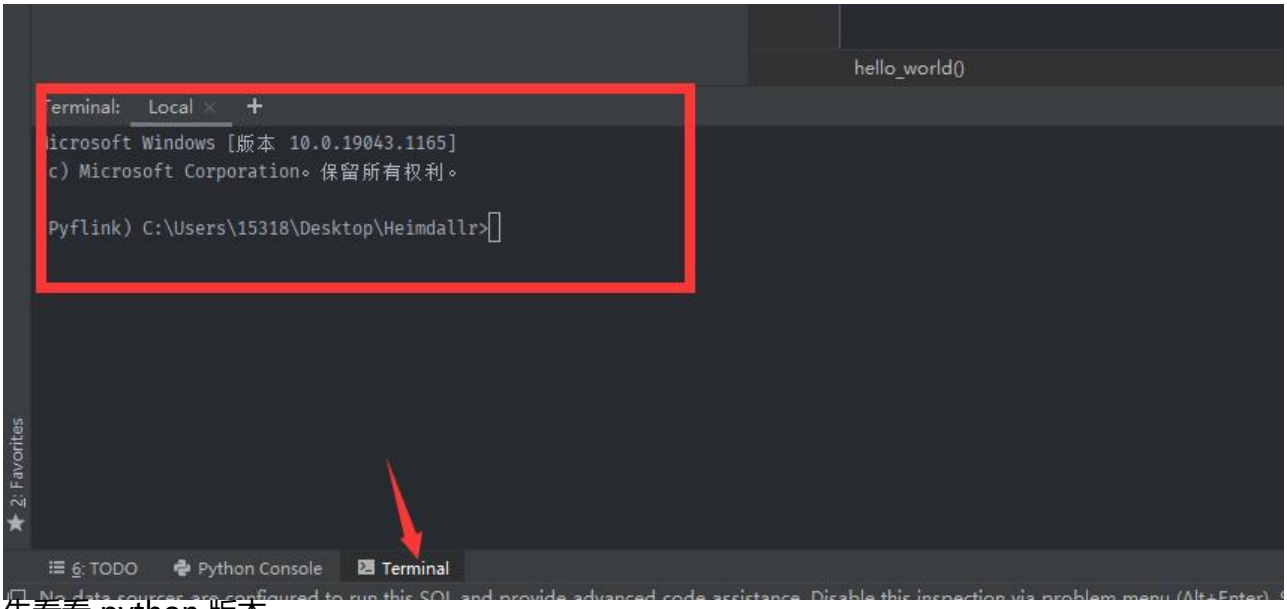
- 新建一个项目，环境选择我们刚刚配置的python 虚拟环境

原文链接：Pyflink 的安装和 windows 开发环境配置



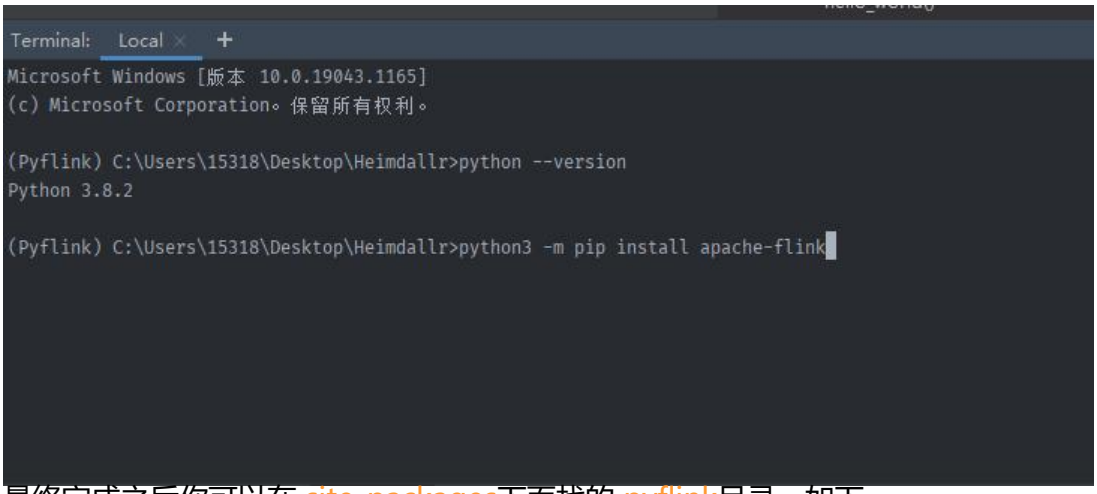
● 安装Pyflink

进入到终端界面

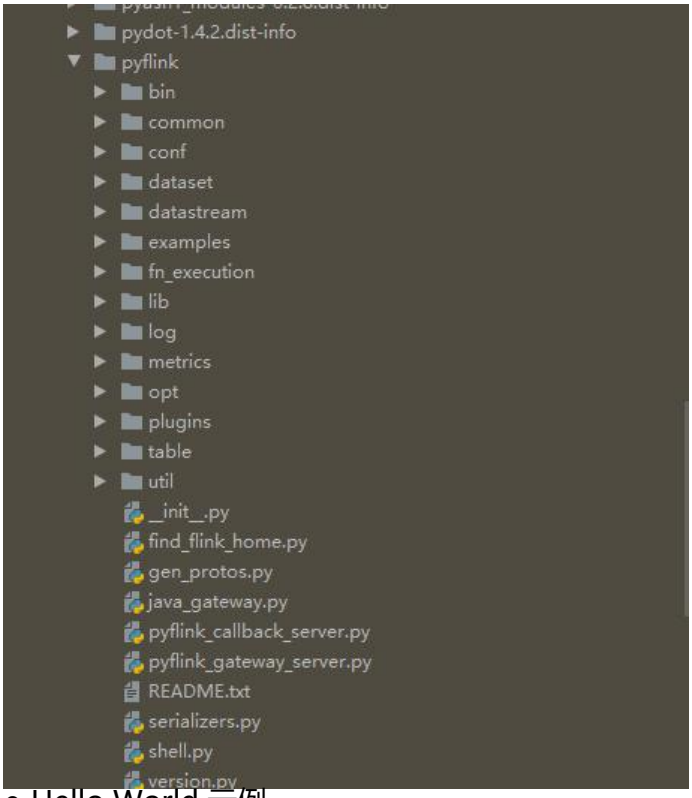


先看看 python 版本

然后安装 Pyflink



最终完成之后你可以在 site-packages下面找的 pyflink目录，如下



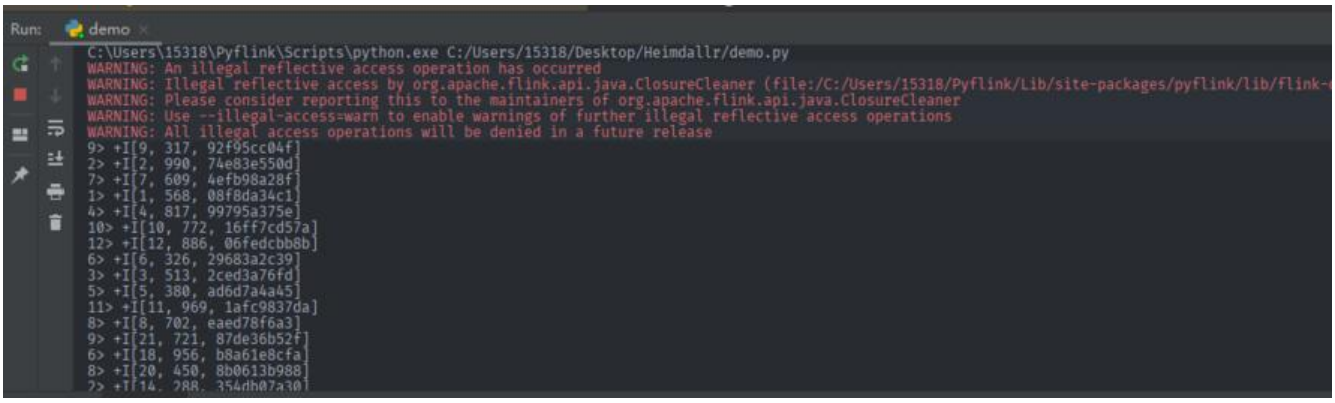
- Hello World 示例

新建一个.py文件输入以下代码

```
#!/usr/bin/env python38
#-*- coding:utf-8 -*-
from pyflink.datastream import StreamExecutionEnvironment
from pyflink.table import EnvironmentSettings, StreamTableEnvironment

def hello_world():
    """
    从随机Source读取数据，然后直接利用PrintSink输出。
    """
    settings = EnvironmentSettings.new_instance().in_streaming_mode().use_blink_planner().buid()
    env = StreamExecutionEnvironment.get_execution_environment()
```

```
    t_env = StreamTableEnvironment.create(stream_execution_environment=env, environment
settings=settings)
    source_ddl = """
                CREATE TABLE random_source (
                    f_sequence INT,
                    f_random INT,
                    f_random_str STRING
                ) WITH (
                    'connector' = 'datagen',
                    'rows-per-second'='5',
                    'fields.f_sequence.kind'='sequence',
                    'fields.f_sequence.start'='1',
                    'fields.f_sequence.end'='1000',
                    'fields.f_random.min'='1',
                    'fields.f_random.max'='1000',
                    'fields.f_random_str.length'='10'
                )
                """

    sink_ddl = """
                CREATE TABLE print_sink (
                  f_sequence INT,
                  f_random INT,
                  f_random_str STRING
                ) WITH (
                  'connector' = 'print'
                )
        """

    # 注册source和sink
    t_env.execute_sql(source_ddl)
    t_env.execute_sql(sink_ddl)

    # 数据提取
    tab = t_env.from_path("random_source")
    # 这里我们暂时先使用 标注了 deprecated 的API, 因为新的异步提交测试有待改进...
    tab.execute_insert("print_sink").wait()
    # 执行作业
    t_env.execute_sql("Flink Hello World")



if __name__ == '__main__':
    hello_world()
```

结果如下



原文链接：Pyflink 的安装和 windows 开发环境配置