

INSTALLING PYSPARK

This presentation will explain how to install PySpark on MacOS
and automatically connect it to Jupyter Notebooks



```
dominiquepaul — -bash — 80x23
[dominiques-mbp:~ dominiquepaul$ python --version
Python 3.6.3 :: Anaconda custom (64-bit)
dominiques-mbp:~ dominiquepaul$
```

I. PYTHON

- We assume that you have python installed
- If you are not sure, open Terminal and enter:
python --version
- If nothing shows up, install python from [here](https://www.python.org/downloads/):

<https://www.python.org/downloads/>



```
dominiquepaul ~ -bash — 80x23
Last login: Sun Sep 23 23:40:24 on ttys003
dominiquepaul$ conda list
# packages in environment at /Applications/anaconda3:
#
# Name          Version          Build          Channel
#-----
_ipyw_jlab_nb_ext_conf 0.1.0            py36h2fc01ae_0
absl-py          0.1.10           <pip>
absl-py          0.3.0            py_0           conda-forge
alabaster        0.7.10           py36h174008c_0
anaconda         custom           py36ha4fed55_0
anaconda-client  1.6.5            py36h04cfe59_0
anaconda-navigator 1.6.11           py36_0
anaconda-project 0.8.0            py36h99320b2_0
appnope          0.1.0            py36hf537a9a_0
appscript        1.0.1            py36h9e71e49_1
asn1crypto       0.22.0           py36hb705621_1
astor            0.6.2            <pip>
astor            0.7.1            py_0           conda-forge
astroid          1.5.3            py36h133018_0
astropy          2.0.2            py36hf79c81d_4
babel            2.5.0            py36h9f161ff_0
backports        1.0              py36ha3c1827_1
backports.shutil_get_terminal_size 1.0.0            py36hd7a2ee4_2
```

```
dominiquepaul ~ -bash — 80x23
itsdangerous     0.24             py36h49fbb8d_1
jasper           1.900.1          h1f36771_4
jbig             2.1              h4d881f8_0
jdcad            1.3              py36h1986823_0
jedi             0.10.2           py36h6325097_0
jinjja2          2.9.6            py36hde4beb4_1
jpeg             9b               haccdd157_1
jsonschema       2.6.0            py36hb385e00_0
jupyter          1.0.0            py36h598a6cc_0
jupyter_client   5.1.0            py36hf6c435f_0
jupyter_console  5.2.0            py36hccf5b1c_1
jupyter_core     4.3.0            py36h93810fe_0
jupyterlab       0.27.0           py36hd3092eb_2
jupyterlab_launcher 0.4.0           py36h93e02e9_0
keras            2.1.3            <pip>
lazy-object-proxy 1.3.1            py36h2fbbe47_0
libcxx           4.0.1            h579ed51_0
libcxxabi        4.0.1            hebdc6815_0
libedit          3.1              hb4e282d_0
libffi           3.2.1            hd939716_3
libgfortran      3.0.1            h93005f0_2
libiconv         1.15             h99df5da_5
libopenblas      0.2.20           hdc02c5d_4
```

2. JUPYTER

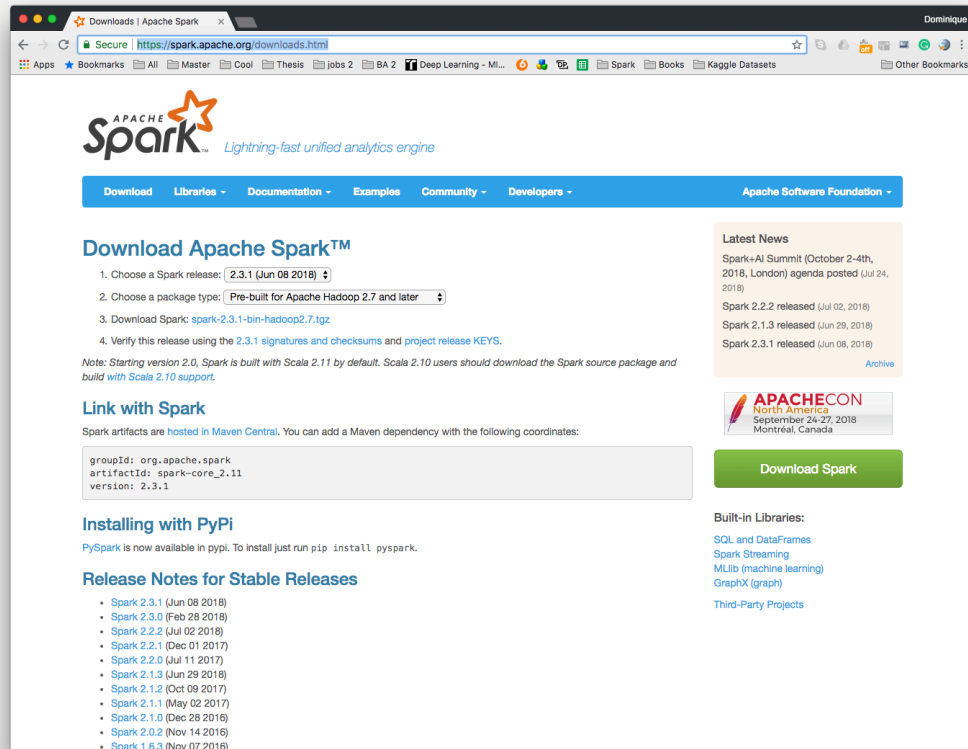
- We also assume that you have Jupyter Notebooks installed
- If you are not sure, open Terminal and enter one of the following:
conda list
pip list
- Jupyter should be one of the packages listed
- Anaconda comes with Jupyter Notebooks pre-installed
- If you are working with pip, you can install Jupyter Notebooks with
pip install jupyter

3. JAVA INSTALLATION

- To use PySpark, we will need the Java Developer Kit
- **Make sure that you are using version 8.** PySpark may not work with other versions such as 10.0
- Go to <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> and download the version for your system
- Once the download has completed, open the installer and follow the instructions

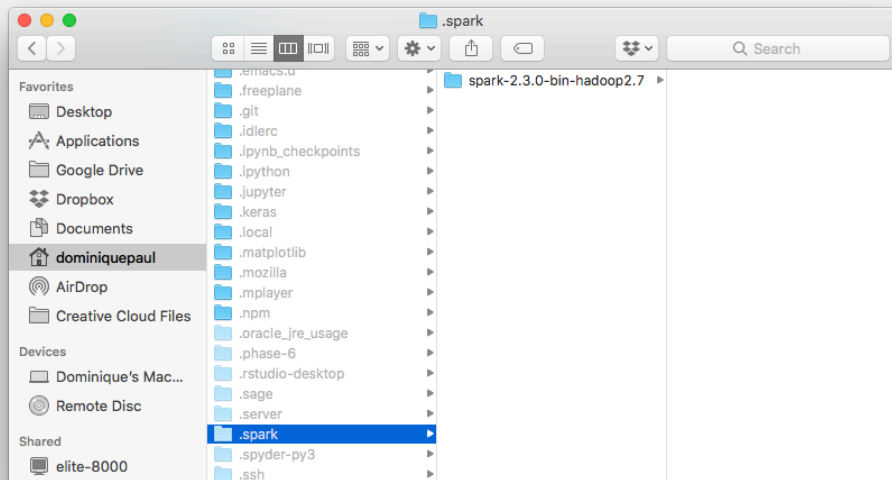
The screenshot shows the Oracle Java SE Development Kit 8 Downloads page. The page has a navigation bar with the Oracle logo, a menu, a search bar, and links for Sign In, Country/Region, and Call. Below the navigation bar, there's a breadcrumb trail: Oracle Technology Network / Java / Java SE / Downloads. The main content area is divided into three columns. The left column contains a list of links: Java SE, Java EE, Java ME, Java SE Subscription, Java Embedded, Java Card, Java TV, Community, and Java Magazine. The middle column is titled 'Java SE Development Kit 8 Downloads' and contains a paragraph about the JDK, a 'See also' section with links to the Java Developer Newsletter, Java Developer Day, and Java Magazine, and a link to the JDK 8u181 checksum. The right column is titled 'Java SDKs and Tools' and contains links to Java SE, Java EE and Glassfish, Java ME, Java Card, NetBeans IDE, and Java Mission Control. Below this is a 'Java Resources' section with links to Java APIs, Technical Articles, Demos and Videos, Forums, Java Magazine, Developer Training, Tutorials, and Java.com. At the bottom, there's a table titled 'Java SE Development Kit 8u181' with columns for Product / File Description, File Size, and Download. The table lists various operating systems and architectures with their corresponding download links and file sizes.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.95 MB	jdk-8u181-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.89 MB	jdk-8u181-linux-arm64-vfp-hflt.tar.gz
Linux x86	165.06 MB	jdk-8u181-linux-i586.rpm
Linux x86	179.87 MB	jdk-8u181-linux-i586.tar.gz
Linux x64	162.15 MB	jdk-8u181-linux-x64.rpm
Linux x64	177.05 MB	jdk-8u181-linux-x64.tar.gz
Mac OS X x64	242.83 MB	jdk-8u181-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.17 MB	jdk-8u181-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.34 MB	jdk-8u181-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.83 MB	jdk-8u181-solaris-x64.tar.Z
Solaris x64	92.11 MB	jdk-8u181-solaris-x64.tar.gz
Windows x86	194.41 MB	jdk-8u181-windows-i586.exe
Windows x64	202.73 MB	jdk-8u181-windows-x64.exe



4. DOWNLOAD SPARK

- Go to <https://spark.apache.org/downloads.html>
- Download the latest stable version of Spark based on a pre-built version for Apache Hadoop



```
.bash_profile — Edited
# Setting PATH for Python 3.6
# The original version is saved in .bash_profile.pysave
PATH="/Library/Frameworks/Python.framework/Versions/3.6/bin:${PATH}"
export PATH

# added by Anaconda3 4.4.0 installer
export PATH="/anaconda/bin:${PATH}"

# added by Anaconda3 5.0.1 installer
export PATH="/Applications/anaconda3/bin:${PATH}"

export SPARK_HOME="/Users/dominiquepaul/.spark"
export PATH=${PATH}:${SPARK_HOME}/bin
```

4. INSTALLING SPARK

1. In your finder home directory, press (command + shift + .) This will reveal all hidden folders and files. You can reverse this by using the same keyboard shortcut again.
2. create a folder called ".spark" and move the unzipped Spark folder into it.
3. Right click (ctrl + click) onto the spark folder and click on "copy spark as pathname"
4. Go back to your homedirectory and open the file called ".bash_profile". This will open the file in textedit.
5. In the file, add the following lines, thereby replacing "YOUR_FILEPATH" with the filepath of the spark folder you just copied:

```
export SPARK_HOME=YOUR_FILEPATH
export PATH=${PATH}:${SPARK_HOME}/bin
```

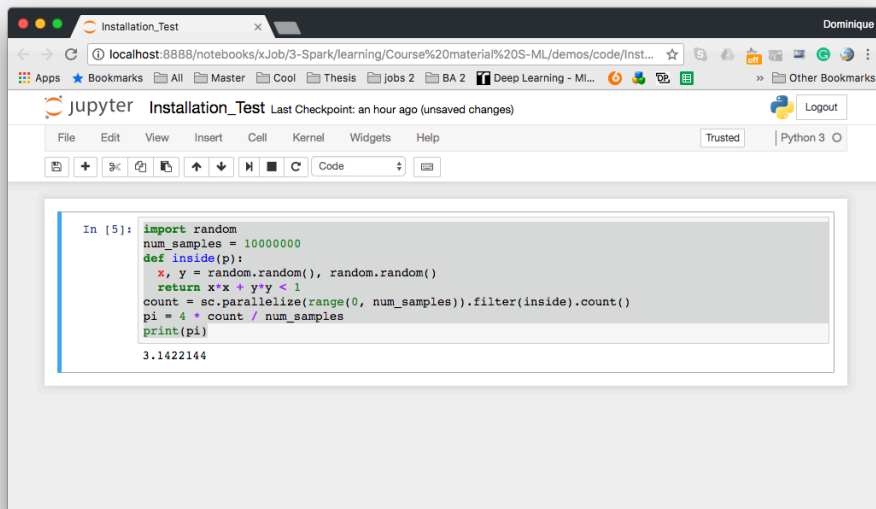
```
.bash_profile — Edited
# The original version is saved in .bash_profile.orig
PATH="/Library/Frameworks/Python.framework/Versions/3.6/bin:${PATH}"
export PATH

# added by Anaconda3 4.4.0 installer
export PATH="/anaconda/bin:${PATH}"

# added by Anaconda3 5.0.1 installer
export PATH="/Applications/anaconda3/bin:${PATH}"

export SPARK_HOME=/Users/dominiquepaul/.spark
export PATH=$PATH:$SPARK_HOME/bin

export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON_OPTS='notebook'
```



4. MAKING THE JUPYTER CONNECTION

- In the same `.bash_profile` file, add the following two lines:
`export PYSPARK_DRIVER_PYTHON=jupyter`
`export PYSPARK_DRIVER_PYTHON_OPTS='notebook'`
- Save the `.bash_profile` file and open terminal. If you had it open already, quit it (command + Q) and re-open it so it can re-load the `.bash_profile` file
- Enter `pyspark` in the terminal, this opens a new Jupyter Notebook with a spark connection.
- Test whether the installation is successful by running the code on the next slide

CODE SNIPPET

```
import random
num_samples = 10000000
def inside(p):
    x, y = random.random(), random.random()
    return x*x + y*y < 1
count = sc.parallelize(range(0, num_samples)).filter(inside).count()
pi = 4 * count / num_samples
print(pi)
```