



## 2.4 Jupyter Notebook

### ■ Jupyter Notebook

- 通过**网页**访问的**交互式**程序编辑工具
- 便于程序的远程访问，屏蔽了不同系统间的显示差异，便于代码共享
- 可以代码、图像、注释、公式、图形、运行结果**整合在同一个文档中**，编写出漂亮的交互式文档



### □ 修改Jupyter Notebook默认目录

- 在Window命令窗口中，输入以下命令：

```
jupyter notebook --generate-config
```

得到配置文件的地址：



```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>jupyter notebook --generate-config
Writing default config to: C:\Users\Administrator\.jupyter\jupyter_notebook_conf
ig.py

C:\Users\Administrator>
```



## 2.4 Jupyter Notebook

- 打开配置文件jupyter\_notebook\_config.py
- 搜索\_dir，定位到配置文件的键值c.NotebookApp.notebook\_dir
- 取消前面的注释符号#，将其值更改为希望的工作目录，如D:\jupyter  
(首先需要在D盘根目录下创建“jupyter”文件夹)

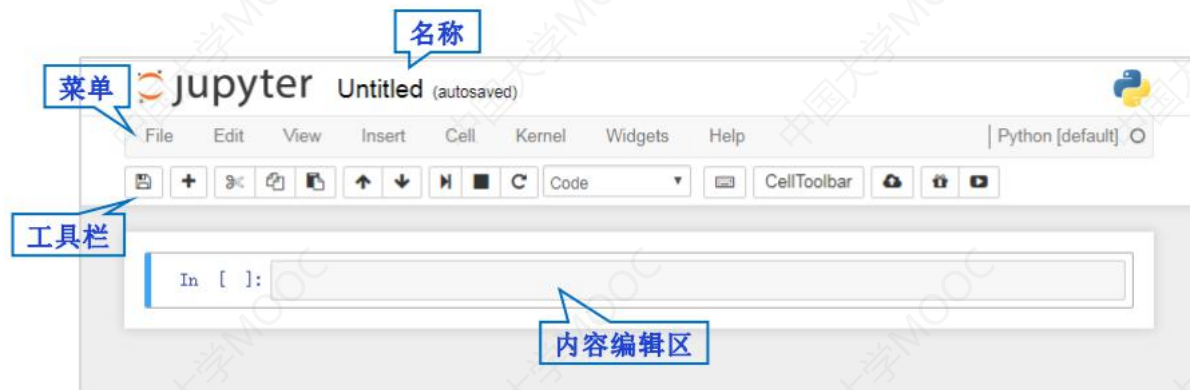
```
## The directory to use for notebooks and kernels.  
c.NotebookApp.notebook_dir = 'D:\jupyter'
```

以后在Jupyter Notebook中创建的文件都会存在这个目录中



## 2.4 Jupyter Notebook

### □ Jupyter Notebook界面

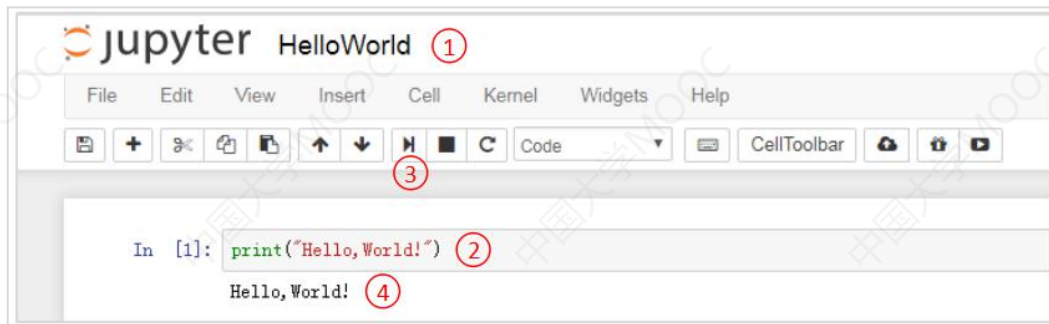


在Jupyter Notebook中，是以**Cell**为基本单位的，即图中的**内容编辑区**



## 2.4 Jupyter Notebook

### 编辑代码



- ① 在**名称**区域修改文件名称
- ② 在**内容编辑区**输入代码
- ③ 点击**运行**按钮
- ④ 运行结果

保存后会在工作目录下生成一个  
**HelloWorld.ipynb**文件，并显示  
在**home**页面的文件列表中



## 2.4 Jupyter Notebook

### 编辑代码

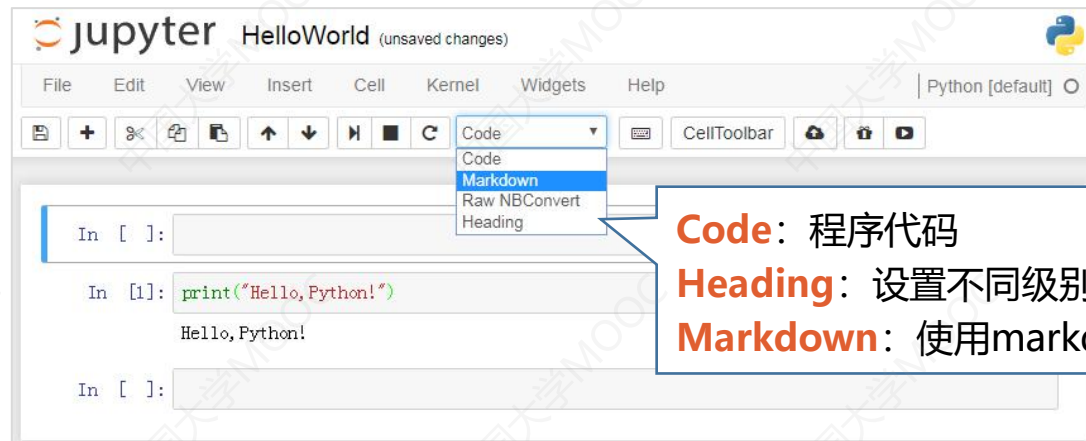


### 保存文件



## 2.4 Jupyter Notebook

### □ 添加标题和注释文字



**Code:** 程序代码

**Heading:** 设置不同级别的标题

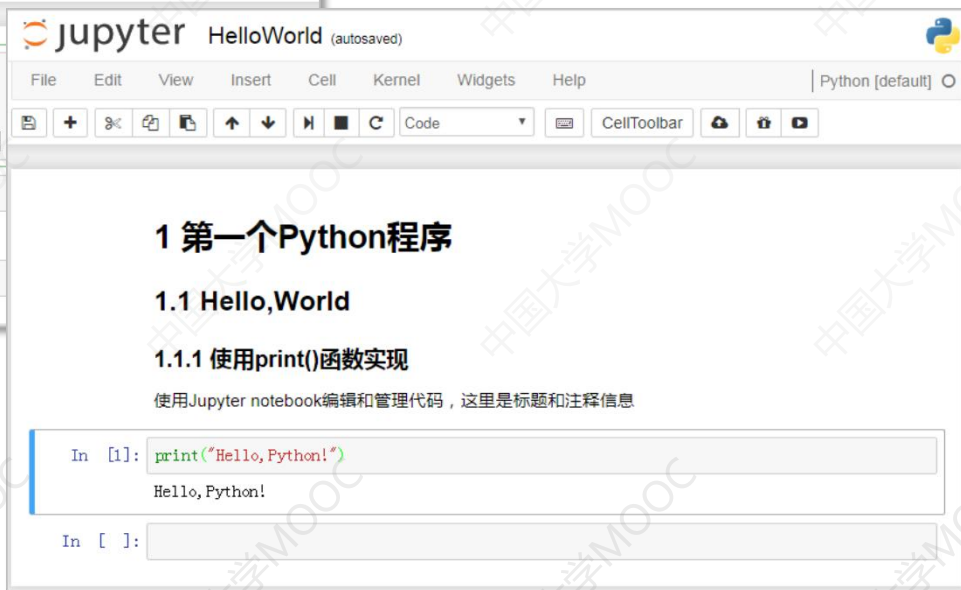
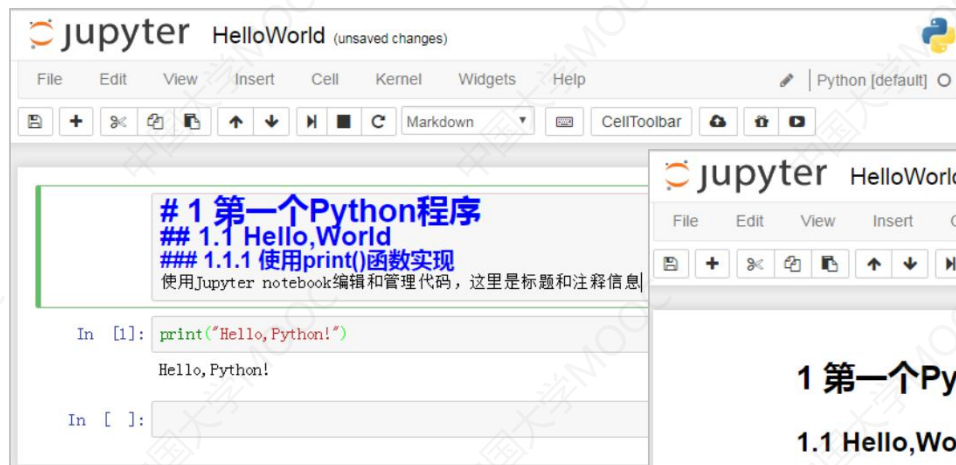
**Markdown:** 使用markdown[ ]的语法来编辑注释文本






## 2.4 Jupyter Notebook

### 添加标题和注释文字



### □ 删除、复制和重命名

选中HelloWorld.ipynb前面的复选框，出现Duplicate（复制）、Rename（重命名）和 （删除）按钮。



### 删除快速查看和编辑快捷键

通过点击菜单栏的**Help**选项，在下拉菜单栏中选中**Keyboard Shortcuts**即可查看命令模式和编辑模式的快捷键。

快捷键

进入keyboard Shortcuts后

Jupyter笔记本有两种不同的键盘输入模式。编辑模式允许您将代码或文本输入到一个单元格中，并通过一个绿色的单元格来表示。命令模式将键盘与笔记本级命令绑定在一起，并通过一个灰色的单元格边界显示。该边框为蓝色的左边框。

方法二：编辑键盘快捷方式

编辑快捷键

命令行模式(按 **Esc** 生效)

- F**: 查找并且替换
- Ctrl-Shift-F**: 打开命令配置
- Ctrl-Shift-P**: 打开命令配置
- Enter**: 进入编辑模式
- P**: 打开命令配置
- Shift-Enter**: 运行代码块, 选择下面的代码块
- Ctrl-Enter**: 运行选中的代码块
- Alt-Enter**: 运行代码块并且插入下面
- Y**: 把代码块变成代码
- M**: 把代码块变成标签
- R**: 清除代码块格式
- 1**: 把代码块变成heading 1
- 2**: 把代码块变成heading 2
- 3**: 把代码块变成heading 3
- 4**: 把代码块变成heading 4
- 5**: 把代码块变成heading 5
- 6**: 把代码块变成heading 6
- G**: 选择上面的代码块
- U**: 选择上面的代码块
- D**: 选择下面的代码块

- Shift-T**: 扩展下面选择的代码块
- Shift-J**: 扩展下面选择的代码块
- A**: 在上面插入代码块
- B**: 在下面插入代码块
- C**: 剪切选择的代码块
- V**: 复制选择的代码块
- Shift-V**: 粘贴到上面
- V**: 粘贴到下面
- Z**: 撤销删除
- D, D**: 删除选中单元格
- Shift-W**: 合并选中单元格, 如果只有一个单元格被选中
- Ctrl-S**: 保存并检查
- S**: 保存并检查
- I**: 切换行号
- O**: 选择单元格的输出
- Shift-O**: 切换选定单元的输出滚动
- H**: 显示快捷键
- I, I**: 中断服务
- O, O**: 重启服务(带窗口)



### □ 关联环境和包—nb\_conda

安装**nb\_conda**包后能够将conda创建的环境与Jupyter Notebook相关联，便于在使用Jupyter Notebook时，在不同的环境下创建笔记本进行工作。

安装：

打开终端，执行conda install nb\_conda命令

卸载：

执行conda remove nb\_conda命令即可卸载nb\_conda包。



### 关联环境和包—nb\_conda

可以在Conda菜单选项下对conda环境和包进行以下一系列操作。

The screenshot displays the JupyterLab interface with the 'Conda' tab selected. It shows a list of 4 Conda environments and a list of 1431 available packages for the 'root' environment.

**Conda Environments Table:**

Action	Name	Default?	Directory
	root	✓	/Users/raxxie/anaconda3
	anaconda3	✓	/Users/raxxie/anaconda3
	py2		/Users/raxxie/anaconda3/envs/py2
	py3		/Users/raxxie/anaconda3/envs/py3

**Available Packages Table (for 'root' environment):**

Name	Version	Channel
<input type="checkbox"/> _ipyw_jlab_nb_ext_conf	0.1.0	https://repo.continuum.io/pkgs/main
<input type="checkbox"/> _license	1.1	https://repo.continuum.io/pkgs/free
<input type="checkbox"/> _nb_ext_conf	0.4.0	https://repo.continuum.io/pkgs/free
<input type="checkbox"/> abstract-rendering	0.5.1	https://repo.continuum.io/pkgs/free
<input type="checkbox"/> accelerate	2.3.1	https://repo.continuum.io/pkgs/pro
<input type="checkbox"/> accelerate_cudalib	2.0	https://repo.continuum.io/pkgs/pro

**Installed Packages Table (for 'root' environment):**

Name	Version	Build	Available
<input type="checkbox"/> _ipyw_jlab_nb_ext_conf	0.1.0	py36h2fc01ae_0	
<input type="checkbox"/> _nb_ext_conf	0.4.0	py36_1	
<input type="checkbox"/> alabaster	0.7.10	py36h174008c_0	
<input type="checkbox"/> anaconda	custom	py36ha4fed55_0	
<input type="checkbox"/> anaconda-client	1.6.6	py36hd86ed11_0	
<input type="checkbox"/> anaconda-navigator	1.6.12	py36_0	

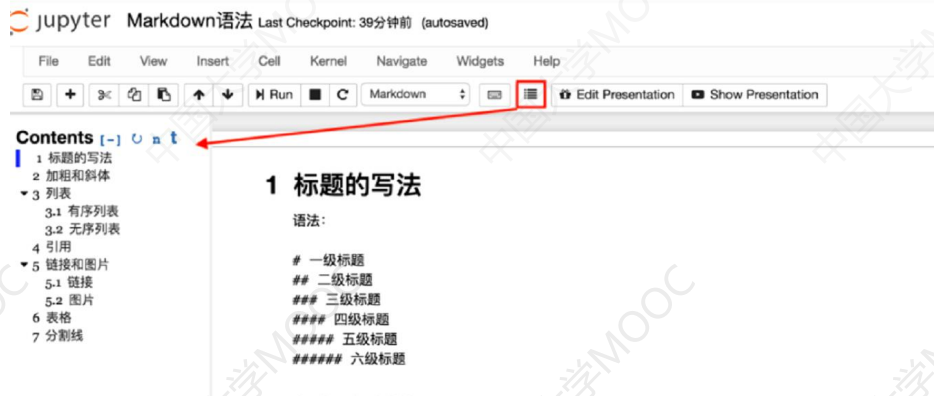
Annotations in the image:

- 克隆环境 (Clone Environment) - points to the clone icon in the environments table.
- 导出环境的txt文档 内含: 平台信息、安装包信息 (Export environment txt document containing platform information and package information) - points to the export icon in the environments table.
- 删除环境 (Delete Environment) - points to the delete icon in the environments table.
- 刷新包列表 (Refresh Package List) - points to the refresh icon in the installed packages table.
- 检查更新 (Check for updates) - points to the check for updates icon in the installed packages table.
- 更新选中包 (Update selected packages) - points to the update selected packages icon in the installed packages table.
- 删除选中包 (Delete selected packages) - points to the delete selected packages icon in the installed packages table.



### □ Markdown生成目录

- ① 打开终端，执行**conda install -c conda-forge jupyter\_contrib\_nbext**命令
- ② 启动Jupyter Notebook，导航栏增加了**Nbextensions**选项，点击Nbextensions后，勾选**Table of Contents (2)**
- ③ 随后在Jupyter Notebook中使用Markdown编写文档，点击下图的图标即可使用



### □ 常用命令

#### ① 获取当前位置：

执行命令：%pwd

作用：在Jupyter Notebook中获取当前文件所在位置的绝对路径

#### ② 加载本地Python文件

执行命令：%load Python文件的绝对路径

作用：在Jupyter Notebook中加载本地的Python文件并执行文件代码

#### ③ 直接运行本地Python文件

执行命令：%run Python文件的绝对路径

作用：不在Jupyter Notebook的单元格中加载本地Python文件，而直接运行。

