

Labs_p2

September 29, 2014



1 Sessió introductòria 2: Edició de documents amb IPython Notebook

1.1 Objectiu

En aquesta pràctica, aprendrem les eines que ofereix **IPython Notebook** per generar documents de text amb format. Aquestes eines us permetran integrar, dins del propi fitxer `.ipynb` totes les explicacions, fórmules, imatges i elements visuals que considereu necessaris per a la redacció d'un informe de pràctiques.

1.2 Estructura d'un fitxer notebook i el llenguatge Markdown

Els fitxers Notebook s'estructuren en blocs o cel·les (en anglès **cells**). Cada bloc s'edita i executa independentment de la resta i és convenient dividir els vostres informes en diversos blocs per facilitar-ne l'edició.

Un dels avantatges de **IPython Notebook** és que permet combinar blocs executables escrits en Python amb blocs de text amb format escrits en llenguatge Markdown. Combinant un i altre tipus de bloc podreu generar els vostres informes de pràctiques, que contindran tant els programes (codi) com les explicacions (text amb format). També hi ha un tercer tipus de bloc (format **RAW**) que permet insertar text sense cap format, per si us cal incloure un text pla. Es poden insertar blocs amb l'element de menú **Insert** i seleccionar el tipus amb el desplegable que es troba al menú d'eines inferior.

El primer tipus de bloc (**Code**) s'identifica amb el prefix `In[]`: a l'esquerra i permet introduir un seguit de sentències Python **que es poden executar**. En aquestes pràctiques, us serviran per introduir, editar, testejar i executar el vostre codi.

El segon tipus (**Markdown**) pot contenir text amb format, és a dir, amb diferents mides i tipus de lletra i material gràfic de suport, com imatges, fórmules, enllaços, etc. Per definir el format del text en aquests blocs s'utilitza el **llenguatge Markdown**.

Per aprendre les generalitats d'aquest llenguatge, seguiu el tutorial següent (en anglès):

<http://www.markdowntutorial.com/>

Llegiu les explicacions del tutorial amb atenció i aneu fent els exercicis que us proposa en el panell de l'esquerra. A la dreta, veureu el resultat del vostre codi Markdown.

Arribats a aquest punt, haurieu de ser capaços de:

- Utilitzar negreta i cursiva (****_**)
- Indicar diferents nivells de títol (**#**)
- Inserir enllaços a pàgines i imatges de la web (`[] () ![] ()`)

- Inserir cites a text (>)
- Definir llistats amb diferents jeraquies (* 1.)
- Separar paràgrafs amb sals de línies de diversos tipus

A més d'aquestes “*marques*” bàsiques, el llenguatge Markdown també permet altres millores de format que ens seràn d'utilitat i que estudiarem en les següents seccions.

1.3 Format per incloure fragments de codi en el text

Sovint, quan estem redactant una explicació ens cal citar una comanda Python i volem indicar que es tracta de codi escrit literalment.

El símbol ``` (accent obert) ens permet marcar una paraula o bloc de text com a `codi` de manera que es presenta amb un tipus de lletra típic de les terminals per indicar que es tracta de codi.

Exemple:

Per obtenir

La comanda Python per imprimir és `print`

Cal introduir

La comanda Python per imprimir és `'print'`

Obviament, es pot combinar amb els símbols anteriors, per utilitzar negretes, cursives, títols, llistes, enllaços, etc.

Si el que volem és inserir un fragment de codi més extens dins del nostre text Markdown, de forma que aparegui amb aspecte de `codi font`, només hem de precedir cadascuna de les línies de codi amb 4 espais en blanc.

Exemple:

Per obtenir

```
#Aquest codi calcula i imprimeix la suma
a = 5
b = 6
print(a + b)
```

Cal introduir (noteu que hi ha 4 espais en blanc davant de cada línia!)

```
#Aquest codi calcula i imprimeix la suma a = 5 b = 6 print(a + b)
```

Important: aquest codi no s'executarà, ja que forma part d'un bloc Markdown. Simplement, permet diferenciar dins de les vostres explicacions, què és `codi` i què no ho és.

1.4 Barres de separació

4 caràcters guió (-) consecutius, en una nova línia, permeten inserir una barra de separació per tal de distingir més clarament seccions del vostre document.

Exemple:

Per obtenir

Cal introduir (noteu que hi ha 4 espais en blanc davant de cada línia!)

1.5 Inserció d'imatges locals

Sovint és necessari inserir al text una imatge pròpia que no es troba al web. En aquest cas, les marques a utilitzar són:

```
![titol](nom_de_la_imatge)
```

on

- **titol** és un text alternatiu que voleu que es mostri, si IPython no troba la imatge
- **nom_de_la_imatge** és la ruta completa al lloc on es troba guardada la imatge dins del vostre disc. Sovint voldreu inserir imatges que es troben en la mateixa carpeta que el vostre notebook; en aquest cas és suficient indicar el nom del fitxer d'imatge. Noteu que cal sempre mantenir el prefix **files/**.

Exemple:

Si tenim una imatge anomenada `python_logo.png` en la nostra carpeta de treball, la comanda

```
![Logotip de Python](python_logo.jpg)
```

Hauria de mostrar la imatge



Figure 1: Logotip de Python

1.6 Taules

Markdown suporta [taules](#). Aquestes es poden inserir fent servir el format següent:

El codi

```
| Fila 1 Columna 1 | Fila 1 Columna 2 | Fila 1 Columna 3 |
|-----|:-----:|-----:|
| Fila 2 Columna 1 | Fila 2 Columna 2 | Fila 2 Columna 3 |
| Fila 3 Columna 1 | Fila 3 Columna 2 | Fila 3 Columna 3 |
```

que genera la taula

Fila 1 Columna 1	Fila 1 Columna 2	Fila 1 Columna 3
Fila 2 Columna 1	Fila 2 Columna 2	Fila 2 Columna 3
Fila 3 Columna 1	Fila 3 Columna 2	Fila 3 Columna 3

Analitzant el codi anterior i el seu resultat, veiem que les marques markdown:

- Les columnes d'una taula es posen entre |
- S'ha de posar guions per indicar la capçalera ----
- Es pot indicar l'alineació fent servir dos punts :
 - Esquerra: |----|
 - Centrat: |:--:|
 - Dreta: |---:|

Obviament, dins del contingut de cada cel·la es pot utilitzar notació **Markdown** per canviar el format del text, inserir elements gràfics, etc.

Variable	Valor
Hola	23.45234


$$\theta$$

1.7 Fórmules matemàtiques amb Latex

De forma semblant al cas anterior, Markdown suporta codi en format **Latex** per inserir fórmules matemàtiques.

Exemple:

El codi

```
$$ z = \frac{x}{y} $$
```

genera la fórmula

$$z = \frac{x}{y}$$

En l'exemple anterior fixeu-vos que les marques **\$\$** **\$\$** encapsulen (contenen) una expressió escrita en codi **Latex** (en l'exemple, el codi **LaTeX** correspon únicament a `z = \frac{x}{y}`). Per tant, totes les fórmules **Latex** que inseriu en Markdown han d'anar precedides i seguides dels símbols **\$\$**.

La sintaxi de formulació amb **Latex** és força complexa i potent. Per aprendre els fonaments seguiu el següent Tutorial de Fórmules amb Latex.

[Tutorial de fórmules amb LaTeX](#)

Podeu trobar una referència completa a les comandes d'edició de fórmules amb **Latex** en el següent enllaç:
http://en.wikibooks.org/wiki/LaTeX/Mathematics#List_of_Mathematical_Symbols

1.8 Conversió d'un fitxer notebook a PDF

A les aules de pràctiques generarem un PDF a partir del Notebook amb els següents passos:

- Obriu una terminal
- Aneu al directori de treball on teniu els fitxers Notebook (extensió `.ipynb`)
- Executeu la següent comanda (on `Nom.Fitxer.Practica.ipynb` és el vostre fitxer Notebook)

```
ipython nbconvert --to latex --post pdf Nom.Fitxer.Practica.ipynb
```

El resultat serà un fitxer PDF amb el contingut de l'informe en el mateix directori.

Nota: en altres entorns com [Windows](#) i [Mac](#) també es pot usar aquesta comanda, però cal tenir instal·lat LaTeX al sistema

1.9 Exercici a lliurar

Construiu un Notebook que tingui l'aspecte del següent document: <http://nbviewer.ipython.org/6768072>

En acabar, heu de lliurar dos fitxers com a resultat de la pràctica:

- El fitxer Notebook
- El fitxer PDF generat a partir del Notebook anterior

Empaqueteu els dos fitxers en **un únic fitxer zip** (els professors de pràctiques us indicaran com fer-ho) i lliureu-lo a través de Campus Virtual. No oblideu indicar els noms de tots els membres del grup de treball al principi del fitxer.