

Jupyter Notebook Intermedio

August 7, 2016

1 Jupyter Notebooks: Intermedio

Jupyter permite integrar sus capacidades de visualización html con el modulo `IPython.display`. De esta forma se puede generar la visualización de texto, imagen, video y ecuaciones de una forma sistematica al integrarlo con patrones generados mediante código.

```
In [1]: from IPython.display import display, Latex, Markdown
```

A continuación usaremos una celda de código en la cual guardaremos una expresión mark-down en una variable python para su posterior visualización.

```
In [2]: mark_text = "_Ejemplo_ de **markdown** \nHola mundo"
        display(Markdown(mark_text))
```

Ejemplo de markdown

Hola mundo

El método `display` cumple la función de renderización de código html en el navegador, donde dicho código es generado por otro método, en este caso por el método `markdown` que convierte markdown a html y si almacena en un objeto de `ipython` para dicho fin.

De esta misma forma se pueden generar elementos de documento más complejos de forma automatica, como tablas.

```
In [3]: fila1 = "|columna 1|columna 2|"
        filaalineacion = "|---:|:---:|"
        fila2 = "|der|cen|"
        display(Markdown(fila1+"\n"+filaalineacion+"\n"+fila2))
```

columna 1	columna 2
der	cen

En este ejemplo las variables fueron asignadas de manera estatica, pero usted puede generar un código que cree el contenido de cada celda, y uniendo su representación en *string* a las cadenas necesarias para formar el patron markdown de una tabla.

De igual manera es posible generar renderizado de ecuaciones de forma sistematica (elemento aprovechado por paquetes de algebra como Sympy).

```
In [4]: latexexp = "$$\frac{mc^2}{2}$$"
        display(Latex(latexexp))
```

$$\frac{mc^2}{2}$$

Otras opciones de visualización soportadas son:

```
In [5]: import IPython.display as Display
        dir(Display)
```

```
Out[5]: ['Audio',
         'DisplayObject',
         'FileLink',
         'FileLinks',
         'HTML',
         'IFrame',
         'Image',
         'JSON',
         'Javascript',
         'Latex',
         'Markdown',
         'Math',
         'Pretty',
         'SVG',
         'ScribdDocument',
         'TextDisplayObject',
         'VimeoVideo',
         'YouTubeVideo',
         '__builtins__',
         '__cached__',
         '__doc__',
         '__file__',
         '__loader__',
         '__name__',
         '__package__',
         '__spec__',
         'clear_output',
         'display',
         'display_html',
         'display_javascript',
         'display_jpeg',
         'display_json',
         'display_latex',
         'display_markdown',
         'display_pdf',
         'display_png',
         'display_pretty',
```

```
'display_svg',  
'publish_display_data',  
'set_matplotlib_close',  
'set_matplotlib_formats']
```

Para consultar sobre las distintas opciones posibles a través de la API de IPython, dirijase a su [documentación](#).