

Plantilla para celdas de código

Luis Fernando Apáez Álvarez

19 de febrero de 2022

Ejemplo de celda de código del lenguaje **plpgsql** :

```
1 CREATE FUNCTION fnc_subir_sueldo(  
2 pid_empleados INT  
3 )  
4 RETURNS VARCHAR  
5 AS  
6 $$  
7 DECLARE vpuesto VARCHAR;  
8 BEGIN  
9     vpuesto = (SELECT puesto FROM empleados WHERE id_empleados = pid_empleados);  
10  
11     IF(vpuesto = 'Vigilante')  
12     THEN  
13         UPDATE empleados SET salario = salario + salario * 0.08  
14         WHERE id_empleados = pid_empleados;  
15         RETURN 'Hecho';  
16     ELSE  
17         IF(vpuesto = 'Cajero')  
18         THEN  
19             UPDATE empleados SET salario = salario + salario * 0.1  
20             WHERE id_empleados = pid_empleados;  
21             RETURN 'Hecho';  
22         ELSE  
23             IF(vpuesto = 'Vendedor')  
24             THEN  
25                 UPDATE empleados SET salario = salario + salario * 0.05  
26                 WHERE id_empleados = pid_empleados;  
27                 RETURN 'Hecho';  
28             ELSE  
29                 RETURN 'Error';  
30             END IF;  
31         END IF;  
32     END IF;  
33  
34 END;  
35 $$  
36 LANGUAGE 'plpgsql' VOLATILE;
```

O podemos escribir algo más sencillo como:

```
1 SELECT * FROM prueba;
```

Ejemplo de celda de código del lenguaje **python** :

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import numba
4
5 # Graficando el fractal de Julia
6
7 def py_julia_fractal(z_re, z_im, j):
8     '''Crea el grafico del fractal de Julia.'''
9     for m in range(len(z_re)):
10         for n in range(len(z_im)):
11             z = z_re[m] + 1j * z_im[n]
12             for t in range(256):
13                 z = z ** 2 - 0.05 + 0.68j
14                 if np.abs(z) > 2.0:
15                     j[m, n] = t
16                     break
17 jit_julia_fractal = numba.jit(nopython=True)(py_julia_fractal)
18 N = 1024
19 j = np.zeros((N, N), np.int64)
20 z_real = np.linspace(-1.5, 1.5, N)
21 z_imag = np.linspace(-1.5, 1.5, N)
22 jit_julia_fractal(z_real, z_imag, j)
23 fig, ax = plt.subplots(figsize=(8, 8))
24 ax.imshow(j, cmap=plt.cm.RdBu_r, extent=[-1.5, 1.5, -1.5, 1.5])
25 ax.set_xlabel(" $\mathrm{Re}(z)$ ", fontsize=18)
26 ax.set_ylabel(" $\mathrm{Im}(z)$ ", fontsize=18)
27 plt.show()
```

Ejemplo de celda de código del lenguaje **latex** :

```
1  $$  
2  k\approx\sqrt{\frac{\lambda_0}{n}}\Phi^{-1}\left(1-\frac{0.05}{2}\right)  
3  =\sqrt{\frac{\lambda_0}{n}}\Phi^{-1}\left(0.975\right)  
4  $$
```

$$k \approx \sqrt{\frac{\lambda_0}{n}} \Phi^{-1} \left(1 - \frac{0,05}{2} \right) = \sqrt{\frac{\lambda_0}{n}} \Phi^{-1} (0,975)$$

Ejemplo de celda de código del lenguaje **r**:

```
1 library(forecast)
2 # Cargamos los datos
3 data(wineind)
4
5 # Vemos los primeros 20 datos
6 head(wineind,20)
7
8 # Vemos un resumen de los datos
9 summary(wineind)
10
11 # Cargamos la libreria
12 library(ggfortify)
13
14 # Graficamos la serie de tiempo utilizando autoplot()
15 autoplot(wineind)
```

Así, para crear una celda de código, por ejemplo de lenguaje *plpgsql*, escribimos en *L^AT_EX* el siguiente código

```
\documentclass{article}

%—— Paquetes necesarios ——
\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}
\usepackage{minted}
\usepackage{vmargin}

%—— Margen ——
\setpapersize{A4}
\setmargins{2cm}% margen izquierdo
{1.5cm}% margen superior
{16.5cm}% anchura del texto
{23.42cm}% altura del texto
{10pt}% altura de los encabezados
{1cm}% espacio entre el texto y los encabezados
{0pt}% altura del pie de pagina
{2cm}% espacio entre el texto y el pie de pagina

\begin{document}
Lo siguiente es una celda de código:

\begin{minted}[frame=lines,linenos=true,breaklines]{plpgsql}
— Esto es un comentario en SQL

SELECT * FROM prueba;
\end{minted}

\end{document}
```

con lo cual conseguiremos algo como

```
1  -- Esto es un comentario en SQL
2
3  SELECT * FROM prueba;
```

Para mayor facilidad, puedes hallar el código anterior de *L^AT_EX* en el siguiente link de github:

[Código Látex](#)