

Código a mano simple con `hola`.

También como un bloque de código con texto monoespaciado.

```
import math
import numpy as np
from lib.analytical import csa
```

Listings

1. Código en Python.	1
codigo/moment.m	1
Código en Python	

```
1 """
2 -----
3 sin2_theta = np.sin(theta)**2
4 """
5 import math
6 import numpy as np
7 from lib.analytical import csa
8
9 sin2_theta = np.sin(theta)**2
10 += -= *= /= + - * / ? < > & % == <=
11 # += -= *= /= + - * / ? < > & % == <=
12 def test(a=100, b=True):
13 <= >= == 2 + 3j * 7e-3
```

Listing 1: Código en Python.

Código en Octave

```
1 % Funcion que calcula los grados de libertad de un elemento
2 % en funcion de las conectividades
3
4 function gdl = conec2gdlframe (conectiv , elem )
5
6 nnod = size(conectiv,2);
7 gdl = zeros(1,3*nnod);
8 for i=1:nnod
9     gdl(3*i-2) = 3*conectiv(elem,i)-2;
10    gdl(3*i-1) = 3*conectiv(elem,i)-1;
11    gdl(3*i) = 3*conectiv(elem,i);
12 end
```

Cargo archivo de octave de la carpeta.

```

1 % Funcion que plotea los momentos
2
3 function h = moment( x_coord , y_coord , M , offset)
4
5 R = offset*1.5 ;
6
7 x_iqz = x_coord-offset*1.5 ;
8 x_der = x_coord+offset*1.5 ;
9
10 % Punto de aplicacion del momento
11 h = plot(x_coord, y_coord,'x', 'linewidth',1.6,'color'
    ,[0.7,0.07,0.07]);
12
13 y_coord = y_coord - offset*.5 ;
14 theta = 0:pi/50:pi;
15
16 xunit = R * cos(theta) + x_coord;
17 yunit = R * sin(theta) + y_coord;
18 h = plot(xunit, yunit, 'linewidth',1.6,'color'
    ,[0.7,0.07,0.07]);
19
20
21 % Coordenadas de puntas del vector
22 x_1_ini = x_coord-sign(M)*offset*1.5-offset*.2+sign(M)*offset
    *.1 ;
23 x_2_ini = x_coord-sign(M)*offset*1.5+offset*.2+sign(M)*offset
    *.1 ;
24 x_fin = x_coord-sign(M)*offset*1.5 ;
25
26 % Ploteo puntas del vector
27 p = plot([x_1_ini,x_fin],[y_coord+offset*.5,y_coord], '
    linewidth',1.6,'color',[0.7,0.07,0.07]);
28 p = plot([x_2_ini,x_fin],[y_coord+offset*.5,y_coord], '
    linewidth',1.6,'color',[0.7,0.07,0.07]);

```