



**École des Ponts**

ParisTech

# Matière

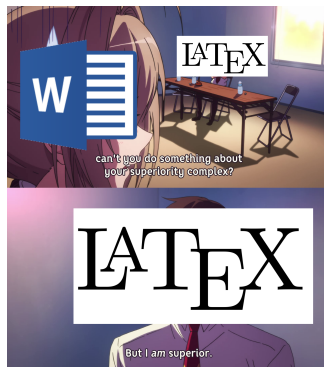
Sous-titre de la présentation

1 Introduction

2 Première partie

# Introduction

- ▶ Premier item
- ▶ On peut importer des images :



# Première partie

1. Item 1
2. Item 2
3. Item 3

## Titre du bloc 2

Ceci est un bloc block

## Titre du bloc 2

Ceci est un bloc exampleblock

## Tire du bloc 3

Ceci est un bloc alertblock

## 3 Appendix 1

## 4 Appendix 2

## Premier exemple de code

```
1     def f(x):  
2         return x**2
```

```
1 # This code is generated by copilot
2 # Create a function that print the sirpinski triangle
3 # The function takes 3 points and a level
4 # The level is the number of iteration
5 # The function returns nothing
6 import matplotlib.pyplot as plt
7
8 def sirpinski(p1, p2, p3, level):
9     # If the level is 0, we draw the triangle
10    if level == 0:
11        # Draw the triangle
12        draw_triangle(p1, p2, p3)
13    else:
14        # Compute the middle points
15        p12 = middle_point(p1, p2)
16        p23 = middle_point(p2, p3)
17        p31 = middle_point(p3, p1)
18        # Draw the 3 triangles
19        sirpinski(p1, p12, p31, level - 1)
20        sirpinski(p12, p2, p23, level - 1)
21        sirpinski(p31, p23, p3, level - 1)
22
23 # Create a function that draw a triangle
24 # The function takes 3 points
25 # The function returns nothing
26 def draw_triangle(p1, p2, p3):
27     # Draw the triangle
28     draw_line(p1, p2)
29     draw_line(p2, p3)
30     draw_line(p3, p1)
31
32 # Create a function that compute the middle point
33 # The function takes 2 points
```

```
34 # The function returns the middle point
35 def middle_point(p1, p2):
36     # Compute the middle point
37     x = (p1[0] + p2[0]) / 2
38     y = (p1[1] + p2[1]) / 2
39     return (x, y)
40
41 # Create a function that draw a line using matplotlib
42 # The function takes 2 points
43 # The function returns nothing
44 def draw_line(p1, p2):
45     # Draw the line
46     plt.plot([p1[0], p2[0]], [p1[1], p2[1]], 'k-')
47
48 # Create a function that draw the sirpinski triangle
49 # The function takes 3 points and a level
50 # The level is the number of iteration
51 # The function returns nothing
52 def draw_sirpinski(p1, p2, p3, level):
53     # Draw the triangle
54     sirpinski(p1, p2, p3, level)
55     # Show the result
56     # Remove axis and do equal aspect
57     plt.axis('equal')
58     plt.axis('off')
59     plt.show()
60
61 # Create a Sirpinski triangle of iteration 3
62 draw_sirpinski((0, 0), (1, 0), (0.5, 0.866), 3)
```