

# VisualData--Exp1

班 级： 计算机23A4

专 业： 计算机科学与技术专业

学 号： 2023013090

姓 名： 蔚嘉琪

## 实验要求

编写程序，分别采用面向对象和面向函数两种方式绘制正弦曲线和余弦曲线。

提示：利用numpy的linspace()、sin()或cos()函数生成样本数据、正弦或余弦值。

## 实验步骤分析

### 1. FP

```
x = np.linspace(0, 2 * np.pi, 100)
```

生成区间为  $[0, 2\pi]$  的等间距数组，总计 100 个点。

```
y_sin = np.sin(x)
```

```
y_cos = np.cos(x)
```

分别计算  $x$  对应的正弦值和余弦值，生成两个数组 `y_sin` 和 `y_cos`。

```
plot()
```

绘图函数，其中给出了 4 个参数，分别是  $x$ ,  $y$ ,  $label$ ,  $color$ 。

其中  $x$  和  $y$  分别表示  $x$ 轴 和  $y$ 轴 的数据， $label$  为图例标签， $color$  为线条颜色。

```
title()
```

图表标题函数

```
grid()
```

为图表添加网格线，`True` 表示网格线是启用的状态。

```
show()
```

显示图表

## 2. OOP

大部分函数与 FP 方式类似，仅部分函数不同。

```
ax = plt.subplots(figsize(8, 4))
```

图像尺寸为 8 x 4，单位为 inch。

这里和 FP 里面的 `figure` 效果类似，至于使用 `subplots` 的原因是因为：对于本次实验，我们需要绘制两份图表，分别对应 sin 和 cos，如果是 FP 思想，就是将两条线画到一张图上面，也就是全局 plt 的方法；但是对于 OOP 思想，我们需要单独管理布局，也就是说要明确 fig 和 ax 对象，运用 OOP 的灵活性，就需要使用 `subplots` 来让我们的代码在后期维护过程中更加灵活与方便。

## 3. 优化函数（基于 ChatGPT）

```
legend()
```

显示图例，用于区分正弦和余弦两条线。

```
tight_layout()
```

自动调整图像元素的间距，避免文字被遮挡或重叠。

```
fig, ax = plt.subplots(figsize=(8, 4))
```

特别的，对于 OOP 的：

创建一个图像对象 `fig` 和一个坐标轴对象 `ax`。

`fig`：代表整张图；

`ax`：表示坐标系区域，真正用于绘图；

`figsize=(8, 4)` 设置图像尺寸。

它之所以能用一个逗号分隔赋值给两个对象，是因为 `plt.subplots()` 的返回值是一个元组，我们可以通过解包（unpacking）的方式将其拆成两个变量。

实际上上述代码等价于：

```
1 result = plt.subplots()
2 fig = result[0]
3 ax = result[1]
```

也就是说，`plt.subplots()` 返回的是一个 (Figure对象, Axes对象) 的元组。

关于元祖解包：

`a, b = (1, 2)` 等价于 `a = 1, b = 2`。

所以同理：`fig, ax = (Figure, Axes)`。

对于两个对象的解释：

`fig` : Figure 对象

- 表示整个**图像画布** (canvas)，可能包含多个坐标轴 (Axes)。
- 它是最高层的图对象，用于控制整个图的大小、标题、保存等。

`ax` : Axes 对象

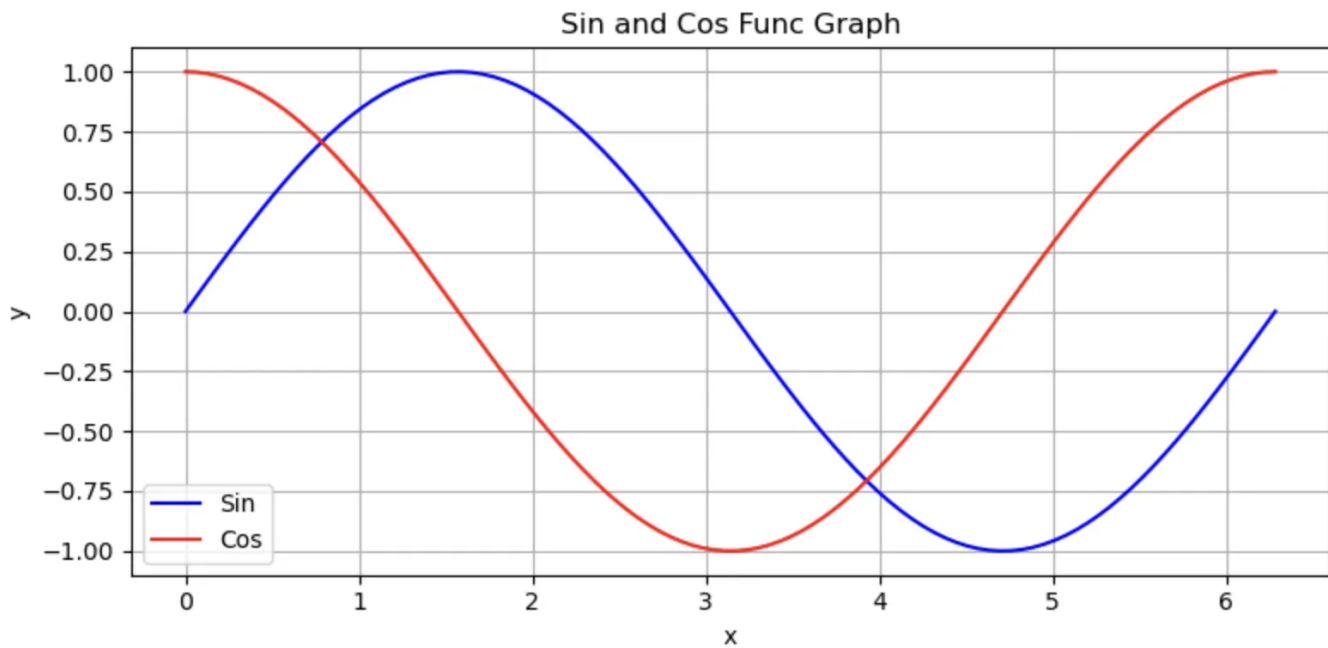
- 表示图中的一个**坐标轴区域**，你在上面画的线图、柱状图都属于 Axes。
- 它控制的是一幅子图 (subplot)，一个 `fig` 可以有多个 `ax`。

## 实验代码及结果

exp1-FP.ipynb

Python |

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  x = np.linspace(0, 2 * np.pi, 100)
5  y_sin = np.sin(x)
6  y_cos = np.cos(x)
7  plt.figure(figsize=(8, 4))
8  plt.plot(x, y_sin, label='Sin', color='blue')
9  plt.plot(x, y_cos, label='Cos', color='red')
10 plt.title('Sin and Cos Func Graph')
11 plt.xlabel('x')
12 plt.ylabel('y')
13 plt.legend()
14 plt.grid(True)
15 plt.tight_layout()
16 plt.show()
```



exp1-OOP.ipynb

Python |

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  x = np.linspace(0, 2 * np.pi, 100)
5  y_sin = np.sin(x)
6  y_cos = np.cos(x)
7  fig, ax = plt.subplots(figsize=(8, 4))
8
9  ax.plot(x, y_sin, label='Sin', color='blue')
10 ax.plot(x, y_cos, label='Cos', color='red')
11 ax.set_title('Sin and Cos Func Graph')
12 ax.set_xlabel('x')
13 ax.set_ylabel('y')
14 ax.legend()
15 ax.grid(True)
16
17 fig.tight_layout()
18 plt.show()
```

Sin and Cos Func Graph

