

# Matplotlib 빠른 참조

Figure, 축, 플롯, 커스터마이징

## 기본 플롯

### 선 그래프

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 8, 3]
plt.plot(x, y)
plt.show()
```

### 빠른 플롯 단축키

```
plt.plot(y) # x auto 0..N-1
plt.plot(x, y, "ro--") # red circles, dashed
plt.plot(x, y, "bs-") # blue squares, solid
```

### 형식 문자열 코드

```
`r` `g` `b` `k` 빨강, 초록, 파랑, 검정
`o` `s` `^` `D` 원, 사각형, 삼각형, 다이아몬드 마커
`-` `--` `.` `:` 실선, 점선, 점-선, 점점선
```

## 서브플롯

### Figure 및 Axes

```
fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_title("Single Plot")
plt.show()
```

### 격자 서브플롯

```
fig, axes = plt.subplots(2, 2, figsize=(8, 6))
axes[0, 0].plot(x, y)
axes[0, 1].bar(x, y)
axes[1, 0].scatter(x, y)
fig.tight_layout()
```

### 공유 축

```
fig, (ax1, ax2) = plt.subplots(1, 2,
                              sharey=True, figsize=(10, 4))
ax1.plot(x, y)
ax2.plot(x, y2)
```

## 레이블 및 제목

### 축 레이블 및 제목

```
plt.plot(x, y)
plt.xlabel("Time (s)")
plt.ylabel("Value")
plt.title("Sensor Reading")
plt.show()
```

### 객체지향 스타일 레이블

```
fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_xlabel("X"); ax.set_ylabel("Y")
ax.set_title("My Plot")
```

### 주석

```
ax.annotate("Peak", xy=(4, 8),
            xytext=(3, 9),
            arrowprops=dict(arrowstyle="->"))
```

## 커스터마이징

### 색상 및 스타일

```
plt.plot(x, y, color="#FF5733",
         linewidth=2, linestyle="--")
plt.plot(x, y2, color="steelblue",
         marker="o", markersize=5)
```

### Figure 크기 및 DPI

```
fig, ax = plt.subplots(figsize=(10, 6), dpi=100)
plt.rcParams["figure.figsize"] = (8, 5)
```

### 스타일 시트

```
print(plt.style.available) # list all
plt.style.use("seaborn-v0_8")
plt.style.use("ggplot")
```

## 막대 차트 및 히스토그램

### 막대 차트

```
labels = ["A", "B", "C", "D"]
values = [23, 45, 12, 37]
plt.bar(labels, values, color="teal")
plt.show()
```

### 그룹형 / 누적 막대

```
import numpy as np
x = np.arange(4); w = 0.35
plt.bar(x - w/2, v1, w, label="2024")
plt.bar(x + w/2, v2, w, label="2025")
plt.xticks(x, labels)
```

### 히스토그램

```
data = np.random.randn(1000)
plt.hist(data, bins=30, edgecolor="black",
         alpha=0.7)
plt.show()
```

## 산점도 및 선

### 산점도

```
plt.scatter(x, y, c="red", s=50,
            alpha=0.6, edgecolors="black")
plt.show()
```

### 컬러맵 산점도

```
sc = plt.scatter(x, y, c=values,
                 cmap="viridis", s=sizes)
plt.colorbar(sc, label="Intensity")
```

### 여러 선

```
plt.plot(x, y1, label="Train")
plt.plot(x, y2, label="Validation")
plt.legend()
plt.show()
```

## 축 및 눈금

### 축 범위 및 스케일

```
ax.set_xlim(0, 10)
ax.set_ylim(-1, 1)
ax.set_xscale("log")
ax.set_yscale("log")
```

## 커스텀 눈금

```
ax.set_xticks([0, 1, 2, 3, 4])
ax.set_xticklabels(["Mon", "Tue", "Wed",
                  "Thu", "Fri"], rotation=45)
```

## 격자

```
ax.grid(True, linestyle="--", alpha=0.5)
ax.grid(axis="y") # horizontal only
```

## 범례

### 범례 추가

```
ax.plot(x, y, label="Series A")
ax.plot(x, y2, label="Series B")
ax.legend(loc="upper right")
```

### 범례 위치

```
`best` 자동 최적 위치 (기본값)
`upper left` 좌상단
`lower right` 우하단
`center` 축 중앙
`bbox_to_anchor=(1, 1)` 축 바깥에 배치
```

## 범례 커스터마이징

```
ax.legend(fontsize=8, frameon=False,
         ncol=2, title="Legend")
```

## 저장

### 파일로 저장

```
plt.savefig("plot.png", dpi=300,
           bbox_inches="tight")
plt.savefig("plot.pdf")
plt.savefig("plot.svg", transparent=True)
```

### 지원 형식

**PNG** 래스터, 웹/화면에 적합  
**PDF** 벡터, 인쇄/논문용에 적합  
**SVG** 벡터, 웹에서 확장 가능  
**EPS** 벡터, 레거시 학술 저널

### Figure 객체로 저장

```
fig, ax = plt.subplots()
ax.plot(x, y)
fig.savefig("output.png", dpi=150,
         facecolor="white")
```

## 주요 패턴

### 이중 축 (두 y 스케일)

```
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.plot(x, temp, "r-", label="Temp")
ax2.plot(x, pressure, "b-", label="Pressure")
```

### 영역 채우기

```
ax.fill_between(x, y_low, y_high,
               alpha=0.3, color="blue")
```

# Matplotlib 빠른 참조

---

## imshow 히트맵

```
data = np.random.rand(10, 10)
plt.imshow(data, cmap="hot",
            interpolation="nearest")
plt.colorbar()
```

## 파이 차트

```
plt.pie(sizes, labels=labels,
        autopct="%1.1f%%", startangle=90)
plt.axis("equal")
```