



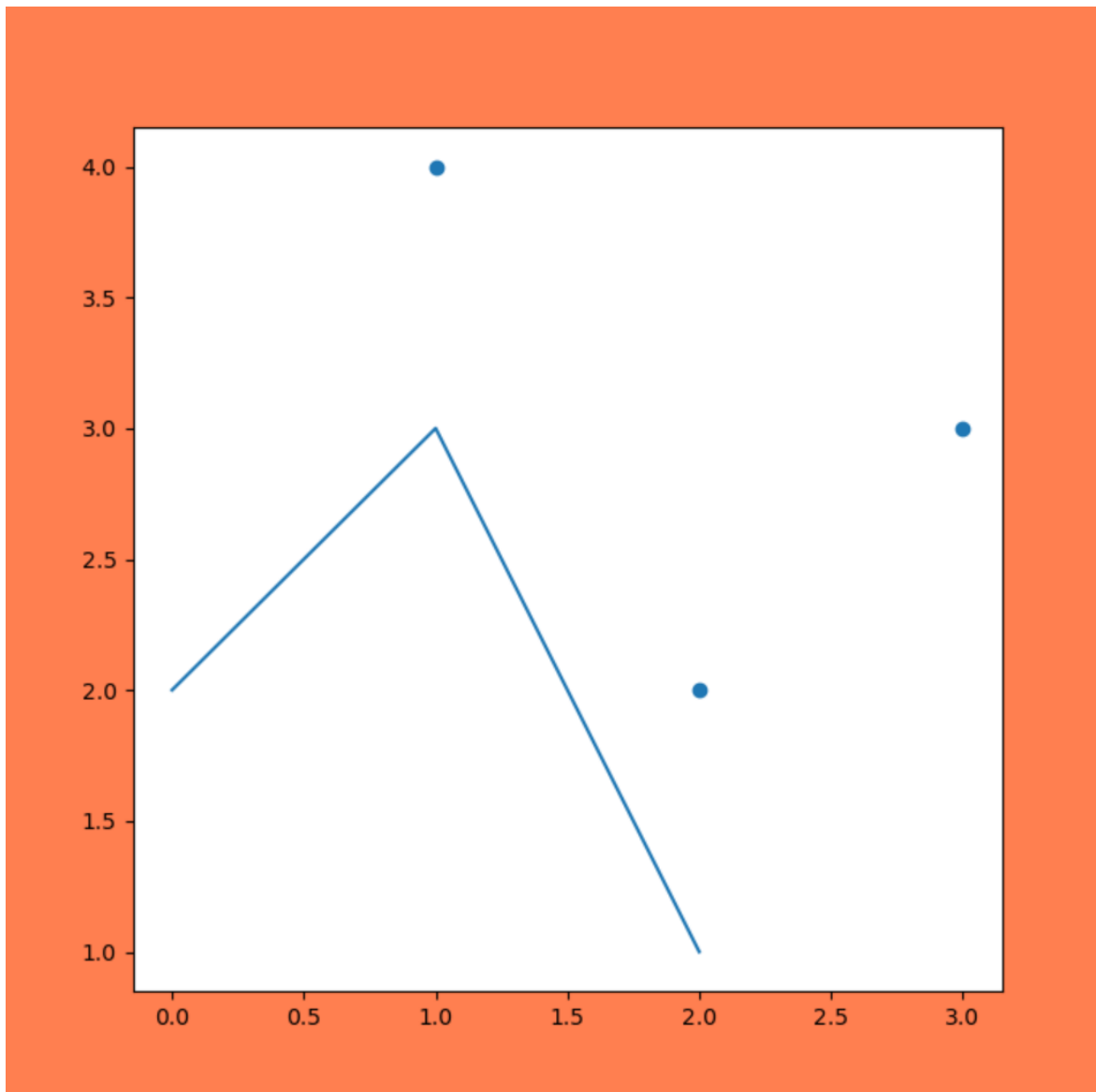
ML Day12 (Matplotlib)

[Matplotlib]

```
import matplotlib.pyplot as plt
import numpy as np
```

▼ 배경 색상 및 크기, 선그래프 or scatterplot

```
fig = plt.figure(figsize=(7, 7), facecolor='coral') # 배경 색상, figure 크기 지정
ax = fig.add_subplot() # 그래프를 만들 테두리 생성
ax.plot([2, 3, 1]) # 선 그래프, 1개의 list일 경우 자동으로 y축 좌표로 인식
ax.scatter([2, 3, 1], [2, 3, 4]) # 점, 각각 x, y 값으로 인식
plt.show()
```

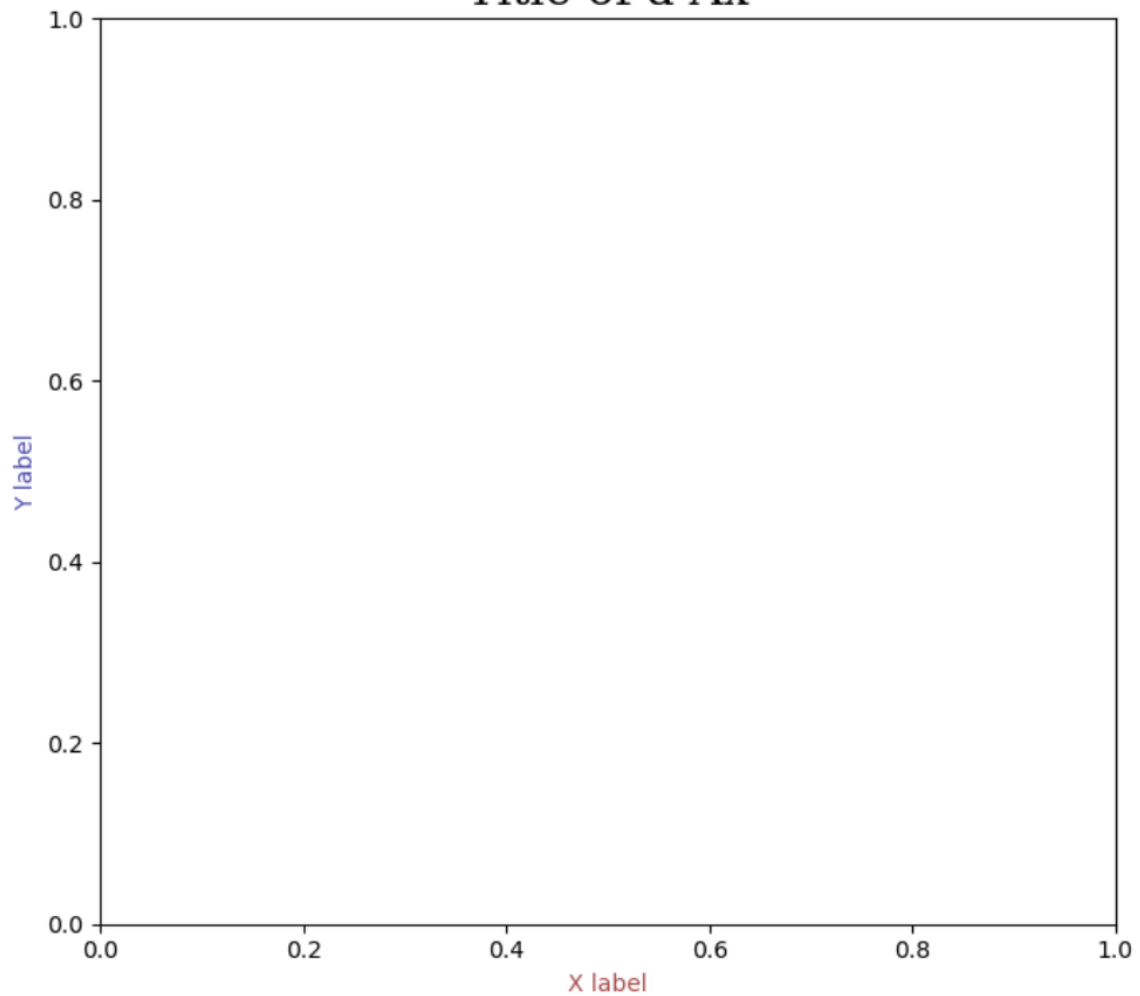


▼ 그래프의 제목 및 label 설정하기

```
fig, ax = plt.subplots(figsize=(7, 7))
fig.suptitle("Title of a Figure",      # figure(그래프)명이 아닌 전체 제목명 설정
             fontsize=30,              # font 크기
             fontfamily='monospace',   # font 설정
             color='gold',             # 제목명의 색상
             alpha=0.9)                # 투명도
ax.set_title("Title of a Ax",          # 그래프의 제목명
             fontsize=20,
             fontfamily='serif')
ax.set_xlabel("X label",               # x축의 label명
             fontsize=10,              # x축 label의 글자 크기
             color='maroon',          # x축 label의 색상 지정
             alpha=0.7)               # x축 label의 투명도 지정
ax.set_ylabel("Y label",
             fontsize=10,
             color='darkblue'
             alpha=0.7)
fig.tight_layout()                    # 그래프의 상하좌우 여백조정
plt.show()
```

Title of a Figure

Title of a Ax



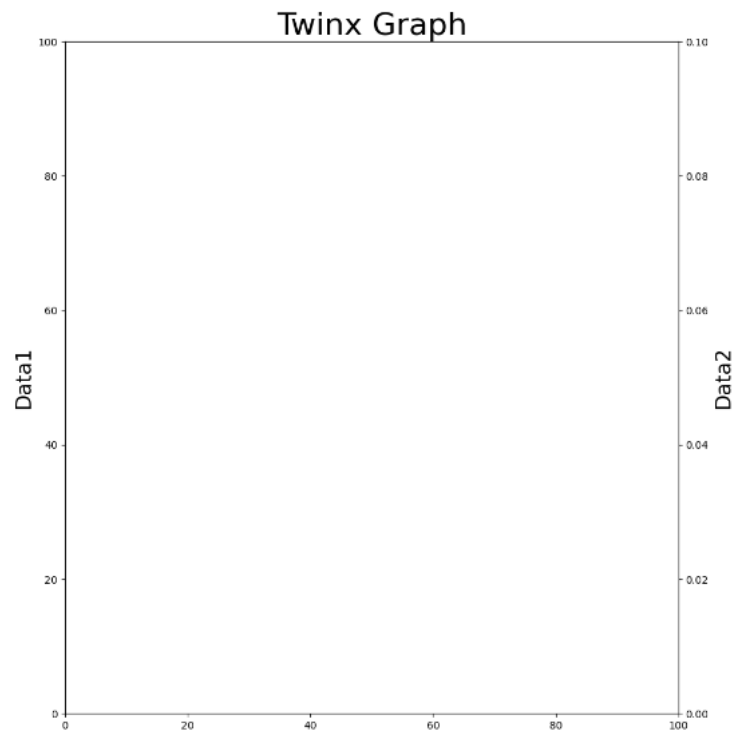
▼ 축 추가하기

```
fig = plt.figure(figsize=(10, 10))

ax1 = fig.add_subplot()
ax2 = ax1.twinx()          # ax2를 ax1 의 twinx(쌍둥이?)로 설정
ax1.set_xlim([0, 100])     # ax1의 x 범위 설정
ax1.set_ylim([0, 100])     # ax1의 y 범위 설정
ax2.set_ylim([0, 0.1])     # ax2의 y 범위 설정

ax1.set_title("Twinx Graph", fontsize=30) # 그래프명, 글자크기 설정
ax1.set_ylabel("Data1", fontsize=20)     # ax1의 y축 명, 글자크기 설정
ax2.set_ylabel("Data2", fontsize=20)     # ax2의 y축 명, 글자크기 설정

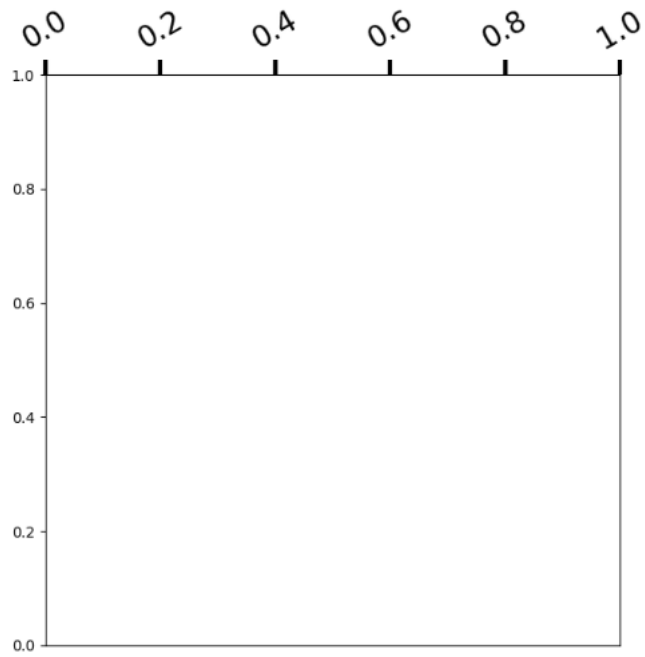
fig.tight_layout()
plt.show()
```



▼ Tick_params(tick 조정)

```
fig, ax = plt.subplots(figsize=(7, 7))
ax.tick_params(axis='x',          # x, y축을 지정하여 조정(두 축 모두 지정하려면 'both')
               labelsz=20,        # tick의 labelsz조정
               length=10,         # tick의 길이
               width=3,           # tick의 너비
               rotation=30,       # tick label의 기울기(양수는 왼쪽으로 회전, 음수는 오른쪽으로 회전)
               bottom=False, labelbottom=False, # bottom(x축 tick) -> False값을 줌으로써 제거, labelbottom(x축 tick label)
               left=False, labelleft=False,      # left(y축 tick) -> False값을 줌으로써 제거, labelleft(y축 tick label)
               top=True, labeltop=True,          # bottom(x축 tick)을 False로 주고 top에서 표현, labelbottom(x축 label)을 False로 주
               right=True, labelright=True)      # left를 False로 주고 right에서 표현, labelright를 False로 주고 labelright로 표현

plt.show()
```



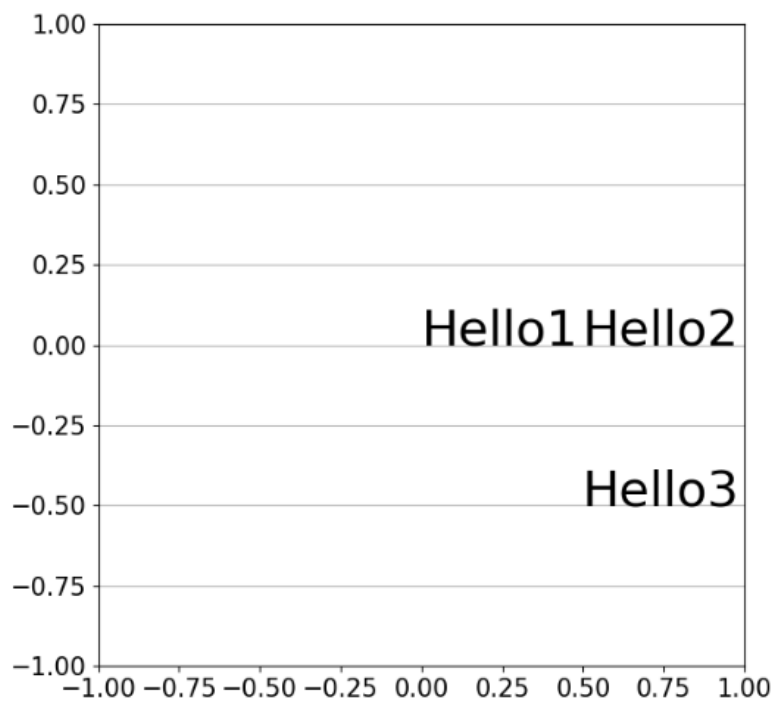
▼ Grid 생성 및 그래프 내 문자표현(text alignment-default)

```
fig, ax = plt.subplots(figsize=(7, 7))
ax.set_xlim([-1, 1])
ax.set_ylim([-1, 1])

ax.grid()                                # grid(격자)생성
ax.tick_params(axis='both',              # 'both'로 x, y축 모두 지정하여 설정(쓰지 않아도 무방)
                labelsize=15)

ax.text(x=0, y=0,                          # grid에 문자가 들어갈 위치 설정(default)
        s="Hello1",
        fontsize=30)
ax.text(x=0.5, y=0,
        s="Hello2",
        fontsize=30)
ax.text(x=0.5, y=-0.5,
        s="Hello3",
        fontsize=30)

plt.show()
```



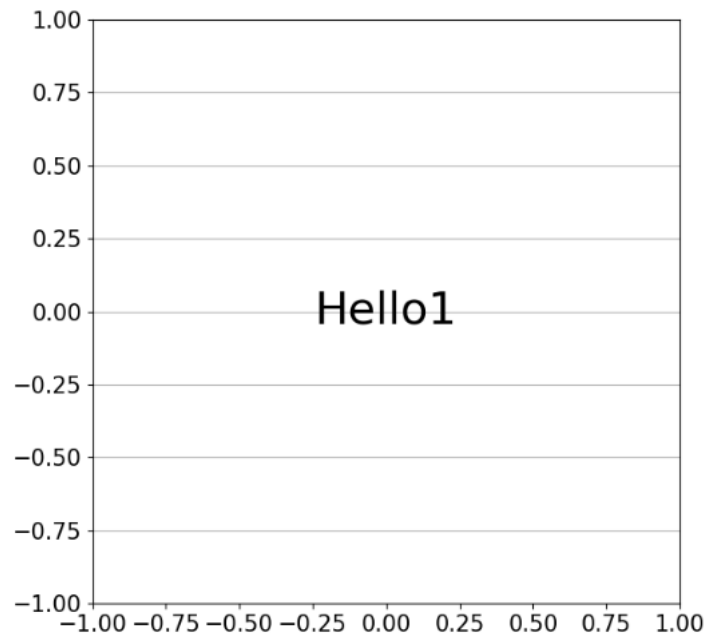
▼ Text alignment(위치 세부조정-vertical alignment, horizontal alignment)

```
fig, ax = plt.subplots(figsize=(7, 7))
ax.set_xlim([-1, 1])
ax.set_ylim([-1, 1])

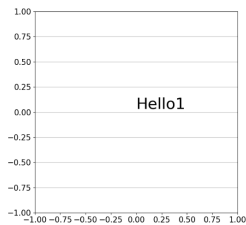
ax.grid(True, axis='y')
ax.tick_params(axis='both',
               labels=15)

ax.text(x=0, y=0,
        va='center',      # 문자의 세로 축 어떤 부분이 (0,0)에 위치할 것인가(top,center,bottom)
        ha='center',      # 문자의 가로 축의 어떤 부분이 (0,0)에 위치할 것인가(left,center,right)
        s="Hello1",
        fontsize=30)

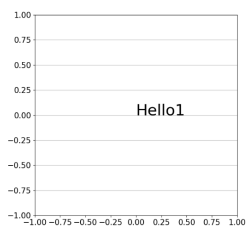
plt.show()
```



(*) `va='bottom', ha='left'`인 경우 `va, ha`값을 주지 않은 default상태와 큰 차이는 없어 보이지만, default인 경우 문자의 왼쪽 하단부가 위 코드에서 주는 좌표인 (0, 0)과 맞닿아 있는 반면, `va='bottom', ha='left'` 을 주게되면 (0, 0)기준으로 미세한 차이로 떨어져 있는 것을 볼 수 있다.



▼ `va='bottom', ha='left'`



▼ `va, ha`를 주지 않은 경우(default)

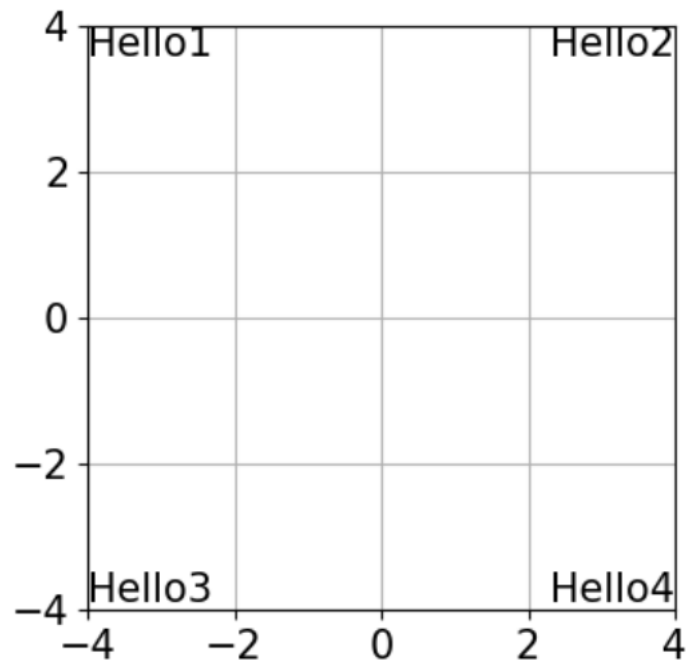
▼ Text alignment(위치 세부조정 - vertical alignment, horizontal alignment) 2

```
fig, ax = plt.subplots(figsize=(4, 4))
ax.set_xlim([-4, 4])
ax.set_ylim([-4, 4])
ax.grid()
ax.tick_params(axis='both',
               labelsz=15)
```

```

ax.text(x=-4, y=4,
        va='top',
        ha='left',
        s='Hello1',
        fontsize=15)
ax.text(x=4, y=4,
        va='top',
        ha='right',
        s='Hello2',
        fontsize=15)
ax.text(x=-4, y=-4,
        va='bottom',
        ha='left',
        s='Hello3',
        fontsize=15)
ax.text(x=4, y=-4,
        va='bottom',
        ha='right',
        s='Hello4',
        fontsize=15)
plt.show()

```



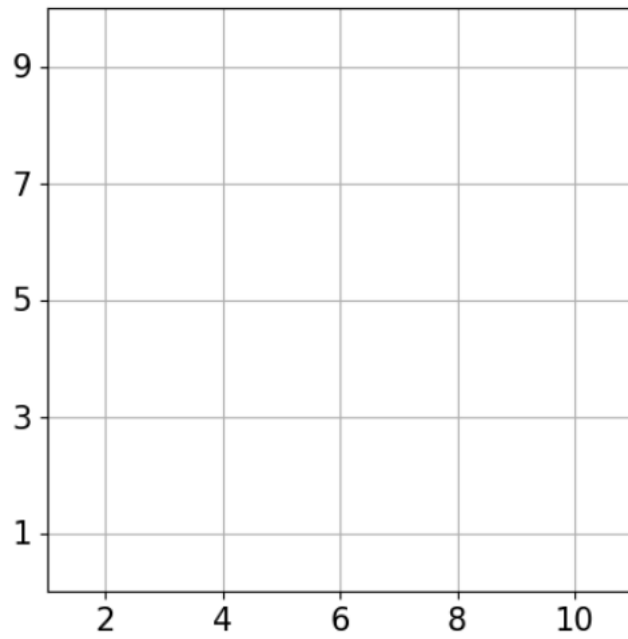
▼ 축의 tick 범위 및 나타낼 특정 tick 지정 - `ax.set_xticks(Arbitrary Location)`

```

fig, ax = plt.subplots(figsize=(5, 5))
ax.set_xlim([1, 11])
ax.set_xticks([2, 4, 6, 8, 10])
ax.set_ylim([0, 10])
ax.set_yticks([1, 3, 5, 7, 9])
ax.grid()
ax.tick_params(axis='both',
               labelsize=15)

plt.show()

```

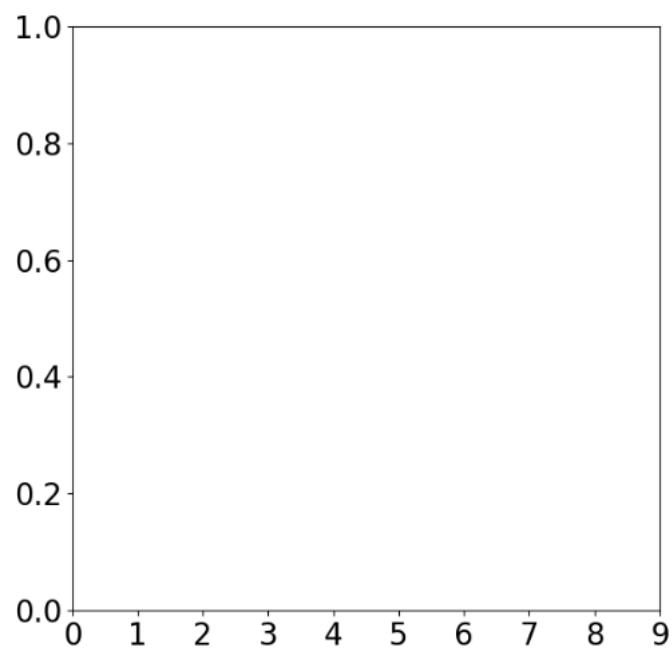



```
fig, ax = plt.subplots(figsize=(7, 7))
ax.tick_params(labelsize=20)

xticks = [_ for _ in range(10)]      # for문을 이용하여 xticks의 범위 지정

ax.set_xticks(xticks)
ax.tick_params(labelsize=20)

plt.show()
```

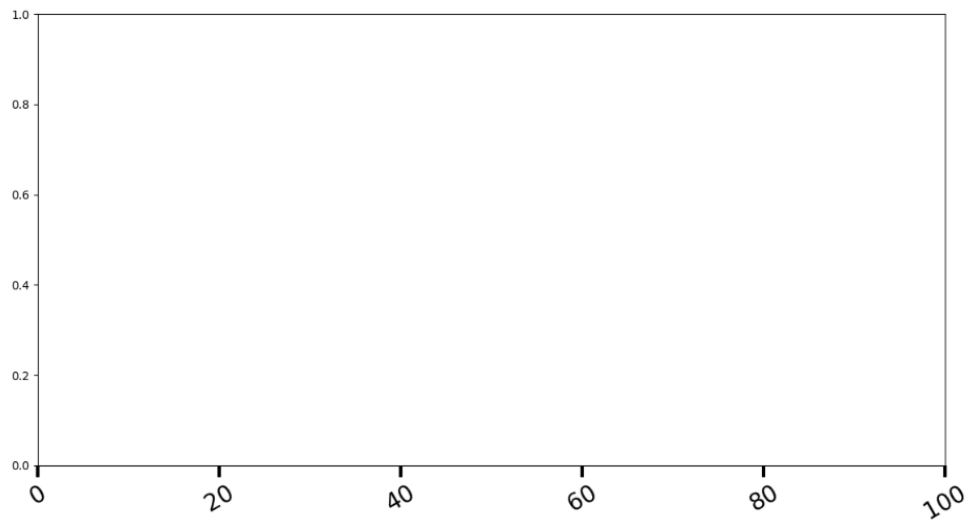


```
fig, ax = plt.subplots(figsize=(14, 7))

xticks = [_ for _ in range(0, 101, 20)] # 0 ~ 100까지 step을 20으로 범위 지정
ax.set_xticks(xticks)

ax.tick_params(axis='x',
               labelsiz=20,
               length=10,
               width=3,
               rotation=30) # 기울기

plt.show()
```



▼ Major, Minor tick 설정하기

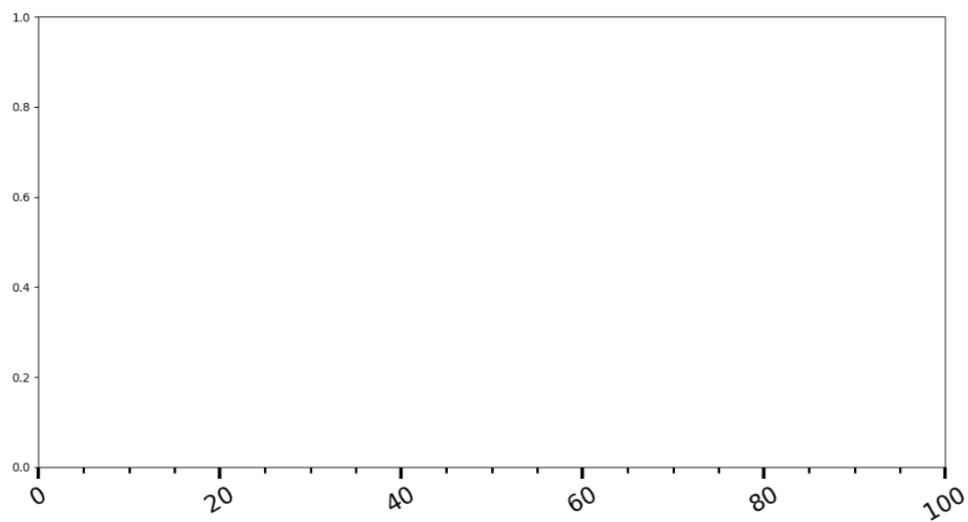
```
fig, ax = plt.subplots(figsize=(14, 7))

major_xticks = [_ for _ in range(0, 101, 20)] # major xticks는 step을 20
minor_xticks = [_ for _ in range(0, 101, 5)] # minor xticks는 step을 5

ax.set_xticks(major_xticks)
ax.set_xticks(minor_xticks, # minor=True라는 값을 줌
              minor=True)

ax.tick_params(axis='x', # major_xticks 설정
               labelsiz=20,
               length=10,
               width=3,
               rotation=30)
ax.tick_params(axis='x', # minor_xticks 설정(which='minor')
               which='minor',
               length=5,
               width=2)

plt.show()
```



```
fig, ax = plt.subplots(figsize=(14, 7))

major_xticks = [_ for _ in range(0, 101, 20)] # major_xticks 범위설정(for문이용)
minor_xticks = [_ for _ in range(0, 101, 5)]   # minor_xticks 범위설정

major_yticks = [_ for _ in range(0, 11, 2)]    # major_yticks 범위설정
minor_yticks = [_ for _ in range(0, 11)]       # minor_yticks 범위설정

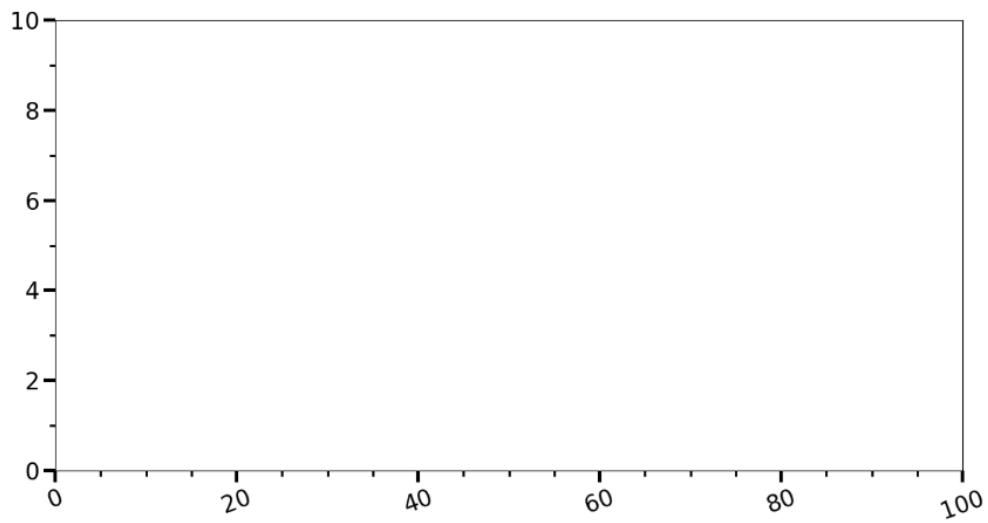
ax.set_xticks(major_xticks)
ax.set_xticks(minor_xticks,                    # minor_xticks - minor=True
               minor=True)

ax.tick_params(axis='x',                      # major_xticks 설정
               labelsz=20,
               length=10,
               width=3,
               rotation=20)
ax.tick_params(axis='x',                      # minor_xticks 설정
               which='minor',
               length=5,
               width=2)

ax.set_yticks(major_yticks)
ax.set_yticks(minor_yticks,                  # minor_yticks - minor=True
               minor=True)

ax.tick_params(axis='y',                      # major_yticks 설정
               labelsz=20,
               length=10,
               width=3)
ax.tick_params(axis='y',                      # minor_yticks 설정
               which='minor',
               length=5,
               width=2)

plt.show()
```



▼ 내 인생그래프 그려보기

```
import numpy as np
import matplotlib.pyplot as plt

age = [0, 5, 10, 15, 20, 21, 25, 30, 35]
level = [0, 1, 10, 15, 20, 0, 90, 2, 85]
fig, ax = plt.subplots(figsize=(10, 10))

ax.plot(age, level)
fig.suptitle("A life graph",
             fontsize=30,
             fontfamily='fantasy',
             color='darkblue')

ax.grid()
major_xticks = [_ for _ in range(0, 39, 5)]
minor_xticks = [_ for _ in range(0, 40)]

major_yticks = [_ for _ in range(0, 101, 20)]
minor_yticks = [_ for _ in range(0, 101)]

ax.set_xticks(major_xticks)
ax.set_xticks(minor_xticks,
              minor=True)

ax.tick_params(axis='x',
              labelsize=15,
              length=10,
              width=3)
ax.tick_params(axis='x',
              which='minor',
              length=5,
              width=2)

ax.set_yticks(major_yticks)
ax.set_yticks(minor_yticks,
              minor=True)

ax.tick_params(axis='y',
              labelsize=15,
              length=10,
              width=3)
ax.tick_params(axis='y',
              which='minor',
              length=5)

ax.set_title("Happiness level",
            fontsize=20,
            fontfamily='serif')
ax.set_xlabel("Age",
            fontsize=10,
            color='maroon',
            alpha=0.7)
ax.set_ylabel("Level",
```

```

        fontsize=10,
        color='maroon',
        alpha=0.7)

ax.text(x=21, y=0,
        va='top',
        ha='center',
        s='Military service shit!!!',
        fontsize=10,
        color='red')
ax.text(x=25, y=90,
        va='bottom',
        ha='center',
        s='Come true!',
        fontsize=10,
        color='red')
ax.text(x=30, y=2,
        va='top',
        ha='center',
        s='Hair dresser....',
        fontsize=10,
        color='red')
ax.text(x=35, y=85,
        va='bottom',
        ha='center',
        s='Now!!',
        fontsize=10,
        color='red')
plt.show()

```

A life graph

