+ rekursi

Import modul built-in & eksternal

https://github.com/dudung/sk5003-02-2022-2

Sparisoma Viridi
Master Program in Computational Science, Nuclear Physics and Biophysics Research Division, Department of Physics, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Bandung 40132, Indonesia

20230506-v9| https://doi.org/10.5281/zenodo.7903448

Silakan berdiskusi untuk kuliah hari ini di
https://github.com/dudung/sk5003-02-2022-2/issues/9

Kerangka

SAP dan referensi

Minggu 7

Minggu	Topik	Subtopik	Capaian Belajar
7	Struktur data, orientasi objek, rekursi dalam Python	Rekursi	Kemampuan untuk memahami dan menguasai rekursi dalam Python

Referensi utama

- Jose M. Garrido, "Introduction to Computational Models with Python", Routledge, 1st edition, 2020,

  url https://isbnsearch.org/isbn/9780367575533.

R1

C11

- Introduction
- Recursive approach to problem solving
- Recursive definition of functions

Rekursi

Pendahuluan

- Teknik dan rancangan yang digunakan untuk implementasi tugas repetitif atau definisi sirkular suatu struktur data.
- Dalam suatu fungsi, rekursi melibatkan pendefinisian fungsi itu sendiri, secara teknis adalah memanggil fungsi itu sendiri.
- Pendekatan ini dapat mendeskripsikan permasalahan dengan cara yang lebih sederhana, lebih jelas ketimbang solusi iteratif.
- Dapat digunakan untuk mempartisi problem rumit (menyele-saikannya dan menggabungkannya kembali).

Pendekatan rekursi

- Suatu fungsi yang merupakan bagian dari pendekatan rekusi memiliki dua bagian

  Bagian 1: Satu atau beberapa kasus yang mendefinisikan kondisi terminasi (penghentian pemanggilan fungsi itu sendiri)

  Bagian 2: Satu atau beberapa kasus untuk melakukan pemanggilan fungsi itu sendiri.

Contoh permasalahan

- Menghitung faktorial
- Menghitung jumlah suku-suku deret
- Menghitung produk suku-suku deret
- Mengakses linked list

Menghitung faktorial

- Faktorial suatu bilangan bulat n dihitung melalui

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$$

- Perumusan rekursi

$$n! = \begin{cases} 1, & n = 0 \\ n \cdot (n-1), & n > 0 \end{cases}$$

Menghitung jumlah suku-suku deret

- Menghitung jumlah suku-suku suatu deret

$$\sum_{i=1}^{n} f(i) = f(n) + f(n-1) + \ldots + f(2) + f(1)$$

- Perumusan rekursi

$$\sum_{i=1}^{n} f(i) = \begin{cases} 0, & i = 0 \\ f(i) + f(i-1), & i > 0 \end{cases}$$

Menghitung produk suku-suku deret

- Menghitung produk suku-suku suatu deret

$$\prod_{i=1}^{n} f(i) = f(n) \cdot f(n-1) \cdot \ldots \cdot f(2) \cdot f(1)$$

- Perumusan rekursi

$$\prod_{i=1}^{n} f(i) = \begin{cases} 1, & i = 0 \\ f(i) \cdot f(i-1), & i > 0 \end{cases}$$

Pertanyaan

- Apa kaitan rumusan ini

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$$

dengan rumusan ini

$$\prod_{i=1}^{n} f(i) = f(n) \cdot f(n-1) \cdot \ldots \cdot f(2) \cdot f(1)$$

Memeriksa linked list

- Pada suatu linked list periksa node pertama dan sebelumnya.
- Bila node telah sampai ujung, misalnya tidak ada node lain yang terkait, pemeriksaan selesai.
- Selama pemeriksaan semua node, nilai parameter pada suatu node dapat dicatat dan diakumulasikan.
- Melaporkan hasil pemeriksaan semua node dalam suatu linked list.
- Terminasi, secara teknis, tercapai saat `next = None`.

Implementasi rekursi

Faktorial

$$n! = \begin{cases} 1, & n = 0 \\ n \cdot (n-1), & n > 0 \end{cases}$$

```python
def factorial(x):
  if x == 0:
    return 1
  else:
    return x * factorial(x - 1)


print("1! =", factorial(1))
print("4! =", factorial(4))
print("3! =", factorial(3))
print("0! =", factorial(0))
print("7! =", factorial(7))
print("6! =", factorial(6))

```

Output:

```
1! = 1
4! = 24
3! = 6
0! = 1
7! = 5040
6! = 720
```

https://onecompiler.com/python/3z7qhg6mp

Jumlah suku-suku suatu deret

$$\sum_{i=1}^{n} f(i) = \begin{cases} 0, & i = 0 \\ f(i) + f(i-1), & i > 0 \end{cases}$$

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

def sumf(n, f):
  if n == 0:
    return 0
  else:
    return f(n) + sumf(n-1, f)

n = 3
sn = sumf(n, f)
print("n =", n)
print("sn =", sn)
```

Output:

```
n = 3
sn = 14
```

https://onecompiler.com/python/3z7qhptkh

Produk suku-suku suatu deret

$$\prod_{i=1}^{n} f(i) = \begin{cases} 1, & i = 0 \\ f(i) \cdot f(i-1), & i > 0 \end{cases}$$

```python
def f(x):
  return (x + 1)

def prodf(n, f):
  if n == 0:
    return 1
  else:
    return f(n) * prodf(n-1, f)

n = 3
sn = prodf(n, f)
print("n =", n)
print("pn =", sn)
```

Output:

```
n = 3
pn = 24
```

https://onecompiler.com/python/3z7qk9w52

Linked list: Book

```
1  class Book:
2    def __init__(self, title):
3      self.title = title
4      self.next = None
5      self.prev = None
6
```

```
7    def append(self, book):
8      if self.next == None:
9        self.next = book
10       book.prev = self
11     else:
12       last = self.next
13       while last.next != None:
14         last = last.next
15       last.next = book
16       book.prev = last
17
```

```
18   def prepend(self, book):
19     if self.prev == None:
20       self.prev = book
21       book.next = self
22     else:
23       first = self.prev
24       while first.prev != None:
25         first = first.prev
26       first.prev = book
27       book.next = first
28
```

Counting books

```
29 def count_next(book):
30   if book.next == None:
31     return 0
32   else:
33     return 1 + count_next(book.next)
34
```

```
35 def count_prev(book):
36   if book.prev == None:
37     return 0
38   else:
39     return 1 + count_prev(book.prev)
40
```

Can you see the recursive functions?

```
41 def count(book):
42   prev = count_prev(book)
43   next = count_next(book)
44   total = 1 + prev + next
45   return total, prev, next
46
```

Use the class `Book` and `count*()` functions

```
47 b1 = Book("Physics")
48 b2 = Book("Chemistry")
49 b3 = Book("Biology")
50 b4 = Book("Medicine")
51 b5 = Book("Architecture")
52 b6 = Book("Sport")
53
54 b1.prepend(b2) # b2-b1
55 b1.append(b3)  # b2-b1-b3
56 b3.prepend(b4) # b4-b2-b1-b3
57 b1.prepend(b5) # b5-b4-b2-b1-b3
58 b1.append(b6)  # b5-b4-b2-b1-b3-b6
```

```
60 print("books: b5-b4-b2-b1-b3-b6")
61 print("(total, prev, next)")
62
63 n = count(b1)
64 print("using b1:", n)
65 n = count(b2)
66 print("using b2:", n)
67 n = count(b3)
68 print("using b3:", n)
69 n = count(b4)
70 print("using b4:", n)
71 n = count(b5)
72 print("using b5:", n)
73 n = count(b6)
74 print("using b6:", n)
75
```

Results

```
$ python linked_list.py
books: b5-b4-b2-b1-b3-b6
(total, prev, next)
using b1: (6, 3, 2)
using b2: (6, 2, 3)
using b3: (6, 4, 1)
using b4: (6, 1, 4)
using b5: (6, 0, 5)
using b6: (6, 5, 0)
```

```
$ python linked_list.py
books: b5-b4-b2-b1-b3-b6
(total, prev, next)
using b1: (6, 3, 2)
using b2: (6, 2, 3)
using b3: (6, 4, 1)
using b4: (6, 1, 4)
using b5: (6, 0, 5)
using b6: (6, 5, 0)
```

```
$ python linked_list.py
books: b5-b4-b2-b1-b3-b6
(total, prev, next)
using b1: (6, 3, 2)
using b2: (6, 2, 3)
using b3: (6, 4, 1)
using b4: (6, 1, 4)
using b5: (6, 0, 5)
using b6: (6, 5, 0)
```

Built-in module: math

pi

$$\pi = 3.141592653589793 \quad (1)$$

```
$ python pi.py
math.pi = 3.141592653589793
```

```
import math

print("math.pi =", math.pi)
```

e

$$e = 2.718281828459045$$

```
$ python e.py
e = 2.718281828459045
```

```
import math

print("e =", math.e)
```

sqrt

$$y = \sqrt{x}$$

```
$ ../../../../scripts/mdpy.sh sqrt.md
x       y
0       0.0
1       1.0
2       1.4142135623730951
3       1.7320508075688772
4       2.0
5       2.23606797749979
6       2.449489742783178
7       2.6457513110645907
8       2.8284271247461903
9       3.0
```

```python
import math

nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

print("x", "y", sep='\t')
for x in nums:
  y = math.sqrt(x)
  print(x, y, sep='\t')
```

pow

$$z = x^y$$

```
$ python pow.py
x       y       z=x^y
1       5       1.0
2       4       16.0
3       3       27.0
4       2       16.0
5       1       5.0
6       0       1.0
100     0.5     10.0
10      -1      0.1
4       2.5     32.0
```

```python
import math

x = [1, 2, 3, 4, 5, 6, 100, 10, 4]
y = [5, 4, 3, 2, 1, 0, 0.5, -1, 2.5]
n = min(len(x), len(y))

print("x\ty\tz=x^y")
for i in range(n):
  z = math.pow(x[i], y[i])
  print(x[i], y[i], z, sep='\t')
```

sin

$$y = \sin x \qquad (1)$$

x	y
0	0
$\frac{1}{6}\pi$	$\frac{1}{2}$
$\frac{1}{3}\pi$	$\frac{1}{2}\sqrt{3}$
$\frac{1}{2}\pi$	1

```
$ python sin.py
θ(°)    θ(rad)  sin θ
0       0.000π  0.000
30      0.167π  0.500
60      0.333π  0.866
90      0.500π  1.000
120     0.667π  0.866
150     0.833π  0.500
180     1.000π  0.000
210     1.167π  -0.500
240     1.333π  -0.866
270     1.500π  -1.000
300     1.667π  -0.866
330     1.833π  -0.500
```

```python
import math

angs = [(i / 12) * 2 * math.pi for i in range(12)]

print("θ(°)", "θ(rad)", "sin θ", sep='\t')
for x in angs:
  c1 = f"{(x / math.pi * 180):.0f}"
  c2 = f"{(x / math.pi):.3f}π"
  c3 = f"{math.sin(x):.3f}"
  print(c1, c2, c3, sep='\t')
```

cos

$$y = \cos x \tag{1}$$

x	y
0	1
$\frac{1}{6}\pi$	$\frac{1}{2}\sqrt{3}$
$\frac{1}{3}\pi$	$\frac{1}{2}$
$\frac{1}{2}\pi$	0

```
$ python cos.py
θ(°)    θ(rad)  cos θ
0       0.000π  1.000
30      0.167π  0.866
60      0.333π  0.500
90      0.500π  0.000
120     0.667π  -0.500
150     0.833π  -0.866
180     1.000π  -1.000
210     1.167π  -0.866
240     1.333π  -0.500
270     1.500π  -0.000
300     1.667π  0.500
330     1.833π  0.866
```

```python
import math

angs = [(i / 12) * 2 * math.pi for i in range(12)]

print("θ(°)", "θ(rad)", "cos θ", sep='\t')
for x in angs:
  c1 = f"{(x / math.pi * 180):.0f}"
  c2 = f"{(x / math.pi):.3f}π"
  c3 = f"{math.cos(x):.3f}"
  print(c1, c2, c3, sep='\t')
```

tan

$$y = \tan x \qquad (1)$$

x	y
0	0
$\frac{1}{6}\pi$	$\frac{1}{3}\sqrt{3}$
$\frac{1}{3}\pi$	$\sqrt{3}$
$\frac{1}{2}\pi$	$\infty, -\infty$

```
$ python tan.py
θ(°)    θ(rad)  tan θ
0       0.000π  0.000
30      0.167π  0.577
60      0.333π  1.732
90      0.500π  16331239353195370.000
120     0.667π  -1.732
150     0.833π  -0.577
180     1.000π  -0.000
210     1.167π  0.577
240     1.333π  1.732
270     1.500π  5443746451065123.000
300     1.667π  -1.732
330     1.833π  -0.577
```

```python
import math

angs = [(i / 12) * 2 * math.pi for i in range(12)]

print("θ(°)", "θ(rad)", "tan θ", sep='\t')
for x in angs:
    c1 = f"{(x / math.pi * 180):.0f}"
    c2 = f"{(x / math.pi):.3f}π"
    c3 = f"{math.tan(x):.3f}"
    print(c1, c2, c3, sep='\t')
```

asin

$$y = \sin x \tag{0}$$

$$x = \sin^{-1} y \equiv \operatorname{asin} y \tag{1}$$

```
$ python asin.py
sin x   x(rad)  x(°)
0       0.000   0.0
0.25    0.253   14.5
0.5     0.524   30.0
0.75    0.848   48.6
1       1.571   90.0
```

```python
import math

x = [0, 0.25,  0.5, 0.75, 1]

print("sin x\tx(rad)\tx(°)")
for s in x:
  rad = math.asin(s)
  deg = math.degrees(rad)
  print(f"{s}\t{rad:.3f}\t{deg:.1f}")
```

acos

$$y = \cos x \tag{0}$$

$$x = \cos^{-1} y \equiv \operatorname{acos} y \tag{1}$$

```
$ python acos.py
cos x   x(rad)  x(°)
0       1.571   90.0
0.25    1.318   75.5
0.5     1.047   60.0
0.75    0.723   41.4
1       0.000   0.0
```

```python
import math

x = [0, 0.25,  0.5, 0.75, 1]

print("cos x\tx(rad)\tx(°)")
for s in x:
  rad = math.acos(s)
  deg = math.degrees(rad)
  print(f"{s}\t{rad:.3f}\t{deg:.1f}")
```

atan

$$y = \tan x \tag{0}$$

$$x = \tan^{-1} y \equiv \operatorname{atan} y \tag{1}$$

```
$ python atan.py
tan x   x(rad)  x(°)
0       0.000   0.0
0.25    0.245   14.0
0.5     0.464   26.6
0.75    0.644   36.9
1       0.785   45.0
1.333   0.927   53.1
```

```python
import math

x = [0, 0.25,  0.5, 0.75, 1, 1.333]

print("tan x\tx(rad)\tx(°)")
for s in x:
  rad = math.atan(s)
  deg = math.degrees(rad)
  print(f"{s}\t{rad:.3f}\t{deg:.1f}")
```

exp

$$y = e^x \tag{1}$$

```
$ python exp.py
x       y=e^x
0       1.0
1       2.718281828459045
2       7.38905609893065
3       20.085536923187668
4       54.598150033144236
```

```python
import math

nums = [0, 1, 2, 3, 4]

print("x", "y=e^x", sep='\t')
for x in nums:
  y = math.exp(x)
  print(x, y, sep='\t')
```

log

$$y = {}^{\mathrm{e}}\log x \equiv \ln x \qquad (1)$$

```
$ python log.py
x       y=log(x)
1.000   0.0
2.718   1.0
7.389   2.0
20.086  3.0
54.598  4.0
```

```python
import math

e = math.exp(1)
nums = [e**0, e**1, e**2, e**3, e**4]

print("x", "y=log(x)", sep='\t')
for x in nums:
  y = math.log(x)
  print(f"{x:.3f}", y, sep='\t')
```

log10

$$y = {}^{10}\log x \equiv \log x \tag{1}$$

```
$ python log10.py
x       y=log10(x)
10000   4.0
1000    3.0
100     2.0
10      1.0
1       0.0
```

```python
import math

nums = [10000, 1000, 100, 10, 1]

print("x", "y=log10(x)", sep='\t')
for x in nums:
  y = math.log10(x)
  print(x, y, sep='\t')
```

fabs

$$y = |x| \tag{1}$$

```
python fabs.py
x       y=|x|
-10     10.0
10.3    10.3
-2.5    2.5
4       4.0
0.12    0.12
-0.85   0.85
```

```python
import math

nums = [-10, 10.3, -2.5, 4, 0.12, -0.85]

print("x\ty=|x|")
for x in nums:
  y = math.fabs(x)
  print(x, y, sep='\t')
```

ceil

$$y = \lceil x \rceil \tag{1}$$

```
$ python ceil.py
x       y
-3.2    -3
-2.7    -2
-1.5    -1
-0.5    0
0       0
0.7     1
1.2     2
2.5     3
```

```python
import math

nums = [-3.2, -2.7, -1.5, -0.5, 0, 0.7, 1.2, 2.5]

print("x", "y", sep='\t')
for x in nums:
  y = math.ceil(x)
  print(x, y, sep='\t')
```

floor

$$y = \lfloor x \rfloor \quad (1)$$

```
$ python floor.py
x       y
-3.2    -4
-2.7    -3
-1.5    -2
-0.5    -1
0       0
0.7     0
1.2     1
2.5     2
```

```python
import math

nums = [-3.2, -2.7, -1.5, -0.5, 0, 0.7, 1.2, 2.5]

print("x", "y", sep='\t')
for x in nums:
  y = math.floor(x)
  print(x, y, sep='\t')
```

trunc

$$\begin{cases} \lceil x \rceil, & x < 0 \\ 0, & x = 0 \\ \lfloor x \rfloor, & x > 0 \end{cases}$$

$$y = [x] \tag{2}$$

```
$ python trunc.py
x       y
-3.2    -3
-2.7    -2
-1.5    -1
-0.5    0
0       0
0.7     0
1.2     1
2.5     2
```

```python
import math

nums = [-3.2, -2.7, -1.5, -0.5, 0, 0.7, 1.2, 2.5]

print("x", "y", sep='\t')
for x in nums:
  y = math.trunc(x)
  print(x, y, sep='\t')
```

radians

$$y\,(\text{rad}) = \left(\frac{\pi}{180}\right) x\,(^\circ) \qquad (1)$$

```
$ python radians.py
x(°)    y(rad)
0       0.0
30      0.5235987755982988
45      0.7853981633974483
60      1.0471975511965976
90      1.5707963267948966
120     2.0943951023931953
135     2.356194490192345
150     2.6179938779914944
180     3.141592653589793
```

```python
import math

x = [0, 30, 45, 60, 90, 120, 135, 150, 180]

print("x(°)\ty(rad)")
for degs in x:
  rads = math.radians(degs)
  print(degs, rads, sep='\t')
```

degrees

$$y\,(^\circ) = \left(\frac{180}{\pi}\right) x\,(\text{rad}) \qquad (1)$$

```
$ python
degrees.py
x(rad)  y(°)
0.0000  0.0
0.5236  30.0
1.0472  60.0
1.5708  90.0
2.0944  120.0
2.6180  150.0
3.1416  180.0
```

```python
import math

pi = math.pi

nums = [0, pi/6, pi/3, pi/2, 2*pi/3, 5*pi/6, pi]

print("x(rad)\ty(°)")
for x in nums:
  y = math.degrees(x)
  print(f"{x:.4f}", f"{y:.1f}", sep='\t')
```

copysign

```
1   import math
2
3   nums = [-1.2, 2.3, -3.4, 4.5]
4   sign = [20.2, 100.9, -30.7, -59.2]
5
6   print("n\ts\tcopysign(n,s)")
7   for n, s in zip(nums, sign):
8     cps = math.copysign(n, s)
9     print(n, s, cps, sep='\t')
10
```

```
11  """
12  $ python copysign.py
13  n       s       copysign(n,s)
14  -1.2    20.2    1.2
15  2.3     100.9   2.3
16  -3.4    -30.7   -3.4
17  4.5     -59.2   -4.5
18  """
```

Built-in module: random

```python
import random as r

for i in range(10):
  x = r.random()
  print(x)


"""
$ python random_float.py
0.42689830565807474
0.05603614535818047
0.09834931748713582
0.29663851125029317
0.9426737300225886
0.6874368918582094
0.4968107188175743
0.6234078659578437
0.5405964599822057
0.785189895099332
"""
```

```python
import random as r

for i in range(10):
  r.seed(0)
  x = r.random()
  print(x)


"""
$ python random_seed.py
0.8444218515250481
0.8444218515250481
0.8444218515250481
0.8444218515250481
0.8444218515250481
0.8444218515250481
0.8444218515250481
0.8444218515250481
0.8444218515250481
0.8444218515250481
"""
```

```
import random as r

for i in range(20):
  x = r.randrange(0, 8, 2)
  print(x, end=' ')
print()

for i in range(20):
  x = r.randrange(7, 10)
  print(x, end=' ')
print()


"""
$ python randrange.py
2 0 2 2 6 0 0 0 4 0 4 0 0 2 6 2 2 6 4 0
9 9 9 8 8 9 7 7 8 8 9 7 8 7 8 8 8 7 7 9
"""
```

```
import random as r

for i in range(20):
  x = r.randint(3, 5)
  print(x, end=' ')
print()


"""
$ python randint.py
5 4 3 4 5 3 3 5 3 5 4 4 5 4 4 3 4 3 3 5
"""
```

```python
import random as r

x = ["A", "B", "C", "D", "E"]

for i in range(20):
  y = r.choice(x)
  print(y, end='')
print()


"""
$ python choice.py
CEACCBCBCCECBCACEDAB
"""
```

```python
import random as r

x = ["A", "B", "C", "D", "E"]

for i in range(1, 10):
  y = r.choices(x, weights=[1, 10, 1, 1, 1], k=i)
  print('-'.join(y))
print()
```

```
"""
$ python choices.py
B
B-E
B-B-B
B-B-A-B
A-B-C-C-E
B-B-E-B-A-B
B-E-B-B-B-B-B
B-B-C-B-B-B-C-B
B-B-B-B-B-B-B-C-B
"""
```

```
1   import random as r
2
3   for i in range(10):
4     x = r.uniform(10, 15)
5     print(f"{x:.3f}", end=' ')
6   print()
7
8
9   for i in range(10):
10    x = r.uniform(20, 23)
11    print(f"{x:.3f}", end=' ')
12  print()
13
14  """
15  $ python uniform.py
16  13.673 13.185 14.458 10.358 13.173 11.644 10.203 12.709 13.410 13.778
17  21.047 22.892 20.234 22.613 21.671 20.894 20.637 21.438 20.921 22.389
18  """
```

Built-in module: statistics

```
1   import statistics as s
2
3   x = [1, 2, 3, 4, 5, 6, 7]
4   mean = s.mean(x)
5   print(x)
6   print("mean =", mean)
7
8   print()
9
10  y = [1, 2, 3, 4, 5, 6]
11  mean = s.mean(y)
12  print(y)
13  print("mean =", mean)
14
```

```
15
16  """
17  $ python mean.py
18  [1, 2, 3, 4, 5, 6, 7]
19  mean = 4
20
21  [1, 2, 3, 4, 5, 6]
22  mean = 3.5
23  """
```

```
1   import statistics as s
2
3   x = [1, 2, 3, 4]
4   fmean = s.fmean(x)
5   print(x)
6   print("fmean =", fmean)
7
8   print()
9
10  y = [1, 2, 3, 4, 5]
11  fmean = s.fmean(y)
12  print(y)
13  print("fmean =", fmean)
14
15
16  """
17  $ python fmean.py
18  [1, 2, 3, 4]
19  fmean = 2.5
20
21  [1, 2, 3, 4, 5]
22  fmean = 3.0
23  """
```

```
import statistics as s

x = [1, 2, 3, 4, 5, 6, 7]
median = s.median(x)
print(x)
print("median =", median)

print()

y = [1, 1, 3, 3, 4, 8, 10]
median = s.median(y)
print(y)
print("median =", median)

print()

z = [ 1, 2, 3, 3, 80, 100]
median = s.median(z)
print(z)
print("median =", median)


"""
$ python median.py
[1, 2, 3, 4, 5, 6, 7]
median = 4

[1, 1, 3, 3, 4, 8, 10]
median = 3

[1, 2, 3, 3, 80, 100]
median = 3.0
"""
```

```python
import statistics as s

x = [3, 1, 1, 3, 2, 2, 2]
mode = s.mode(x)
print(x)
print("median =", mode)


"""
$ python mode.py
[3, 1, 1, 3, 2, 2, 2]
median = 2
"""
```

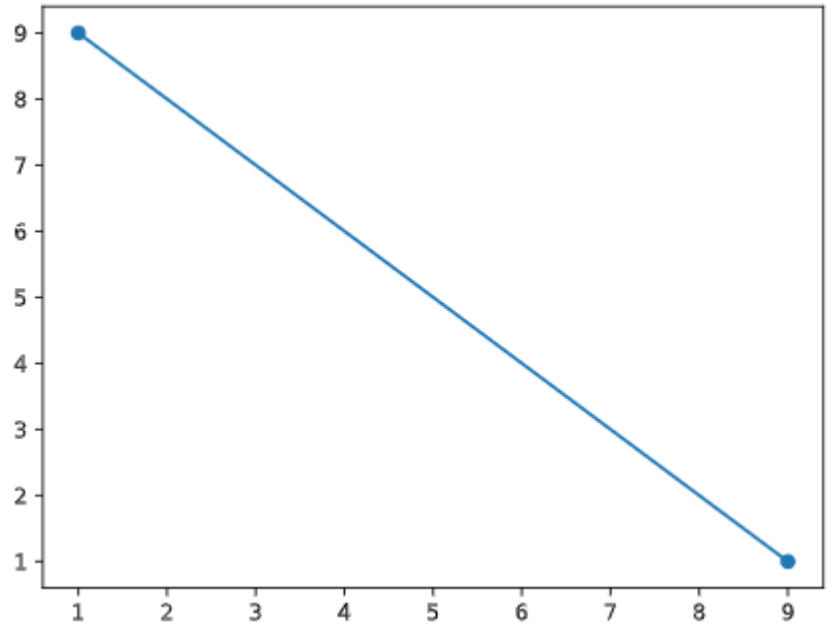
External package: matplotlib basics

min_loc

minimum lines of code

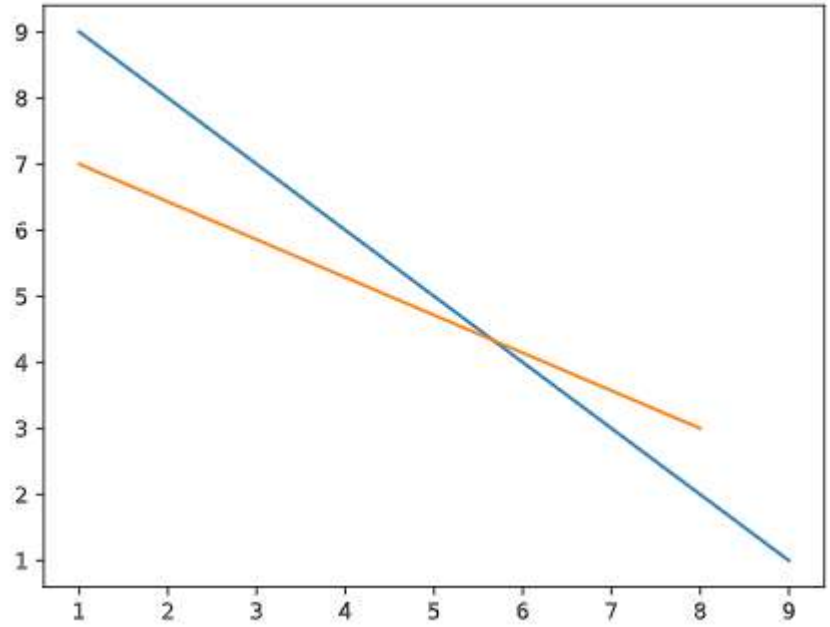
```
import matplotlib.pyplot as plt
x = [1, 9]
y = [9, 1]
plt.plot(x, y, 'o-')
plt.show()
```

```
$ python min_loc.py
```



series

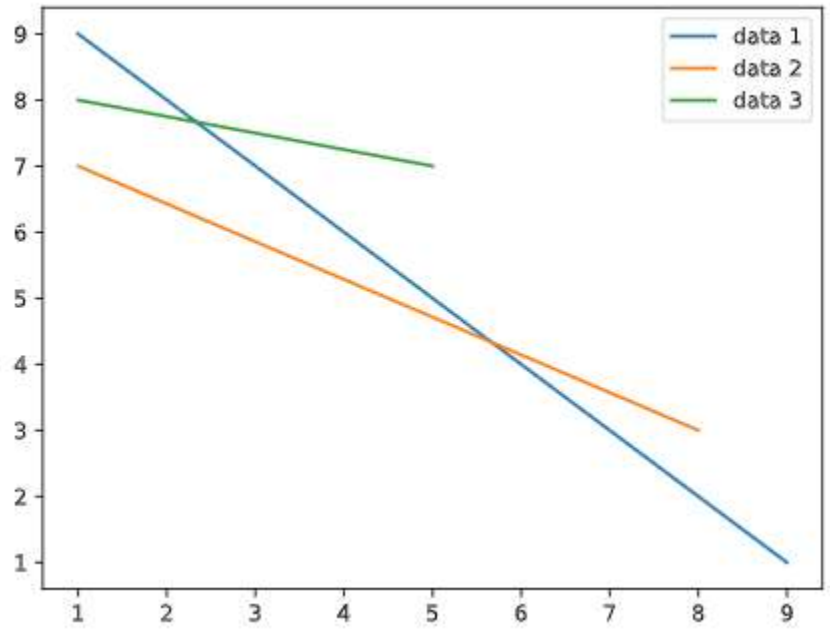
```
import matplotlib.pyplot as plt

x1 = [1, 9]
y1 = [9, 1]

x2 = [1, 8]
y2 = [7, 3]

plt.plot(x1, y1, x2, y2)
plt.show()
```




```
$ python series.py
```

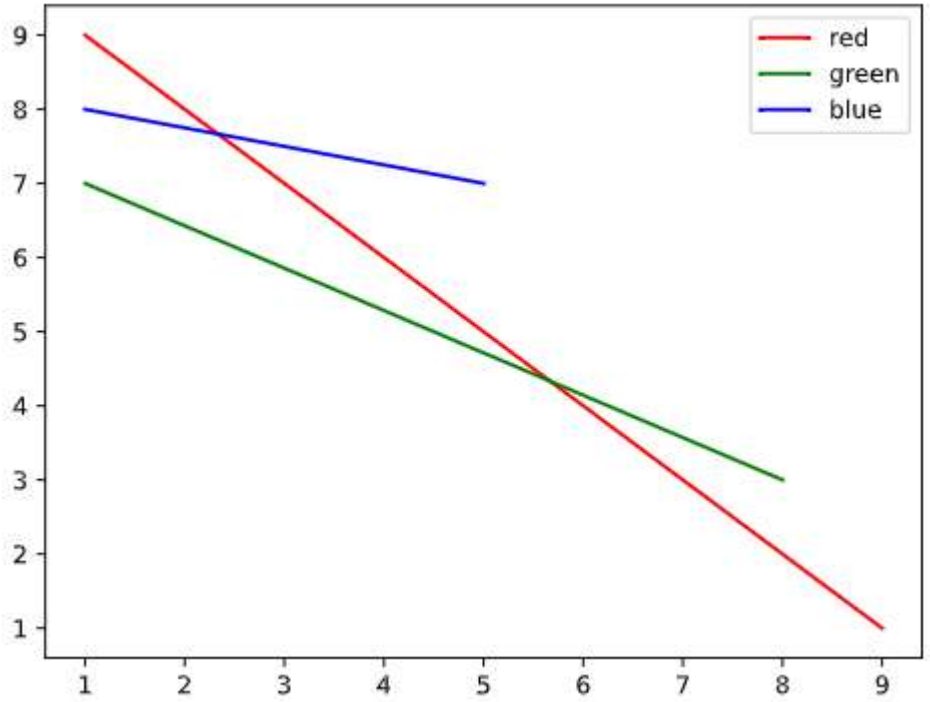
legend

```
import matplotlib.pyplot as

x1 = [1, 9]
y1 = [9, 1]

x2 = [1, 8]
y2 = [7, 3]

x3 = [1, 5]
y3 = [8, 7]

legends = ['data 1', 'data
```



line_color

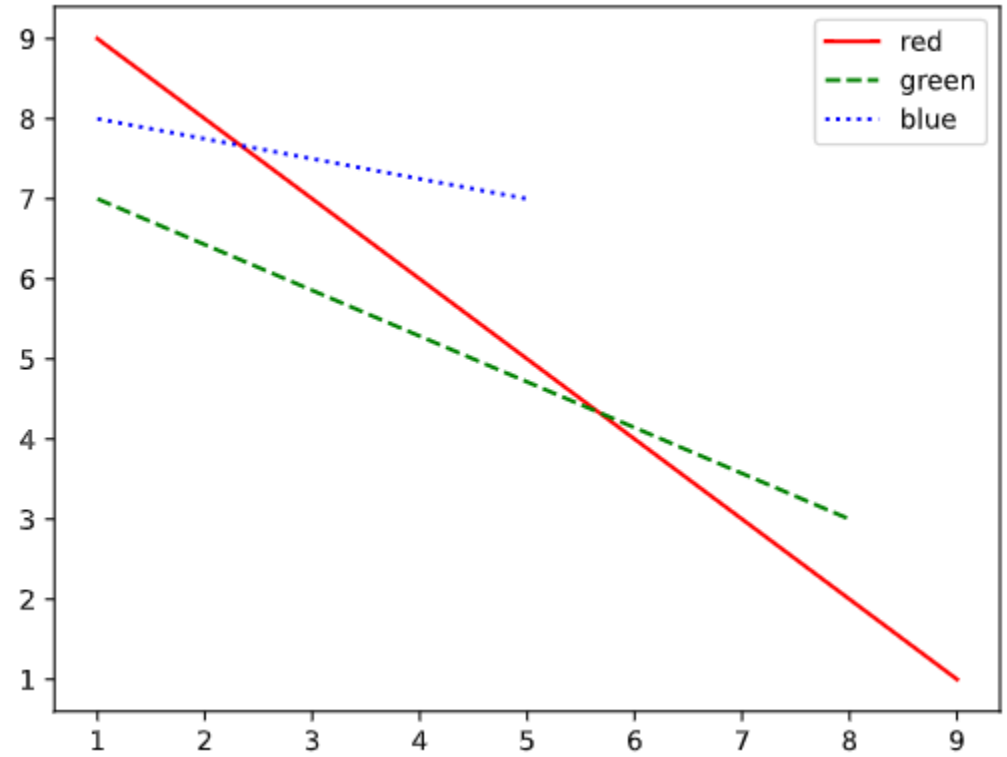
```
import matplotlib.pyplot as plt

x1 = [1, 9]
y1 = [9, 1]

x2 = [1, 8]
y2 = [7, 3]

x3 = [1, 5]
y3 = [8, 7]

legends = ['red', 'green', 'blue']

plt.plot(x1, y1, 'r', x2, y2, 'g', x3, y3, 'b')
plt.legend(legends)
plt.show()
```




```
$ python line_color.py
```

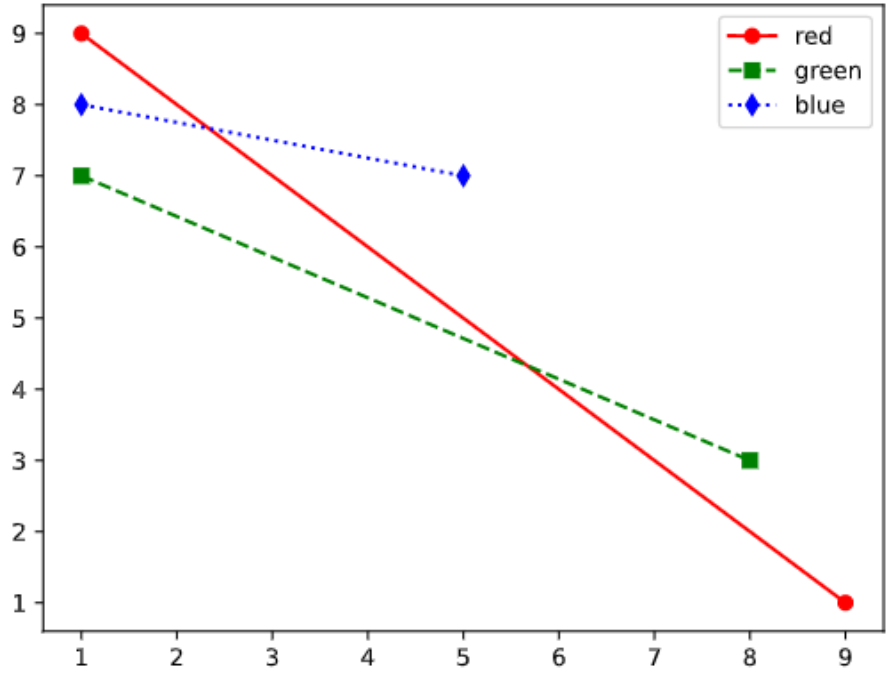
line_style

```
import matplotlib.pyplot as plt

x1 = [1, 9]
y1 = [9, 1]

x2 = [1, 8]
y2 = [7, 3]

x3 = [1, 5]
y3 = [8, 7]

legends = ['red', 'green', 'blue']

plt.plot(x1, y1, 'r-', x2, y2, 'g--', x3, y3, 'b:')
plt.legend(legends)
plt.show()
```



marker

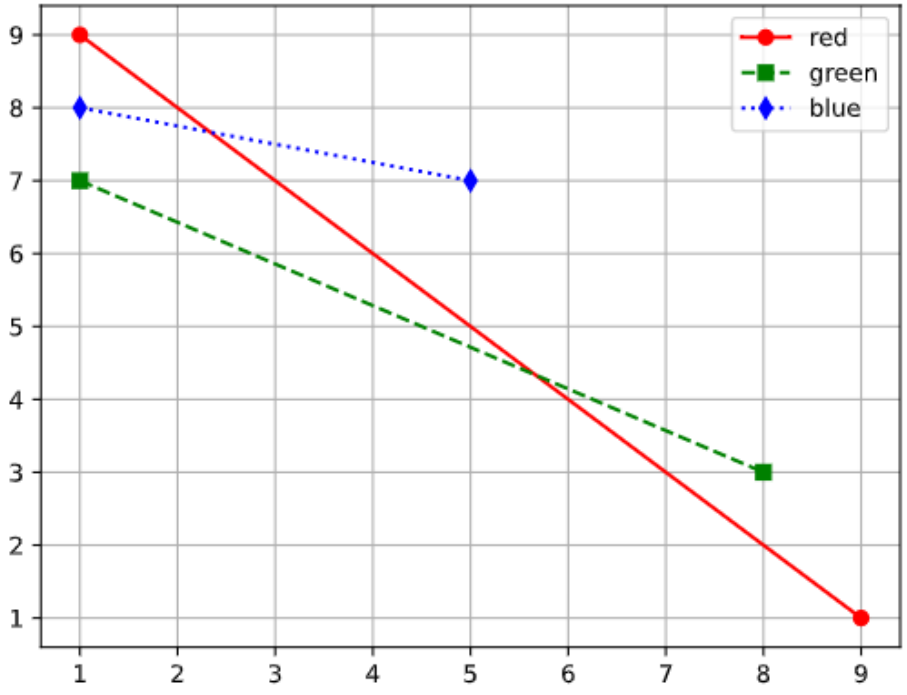
```
import matplotlib.pyplot as plt

x1 = [1, 9]
y1 = [9, 1]

x2 = [1, 8]
y2 = [7, 3]

x3 = [1, 5]
y3 = [8, 7]

legends = ['red', 'green', 'blue']

plt.plot(x1, y1, 'ro-', x2, y2, 'gs--', x3, y3, 'bd:')
plt.legend(legends)
plt.show()
```




```
$ python marker.py
```

grid

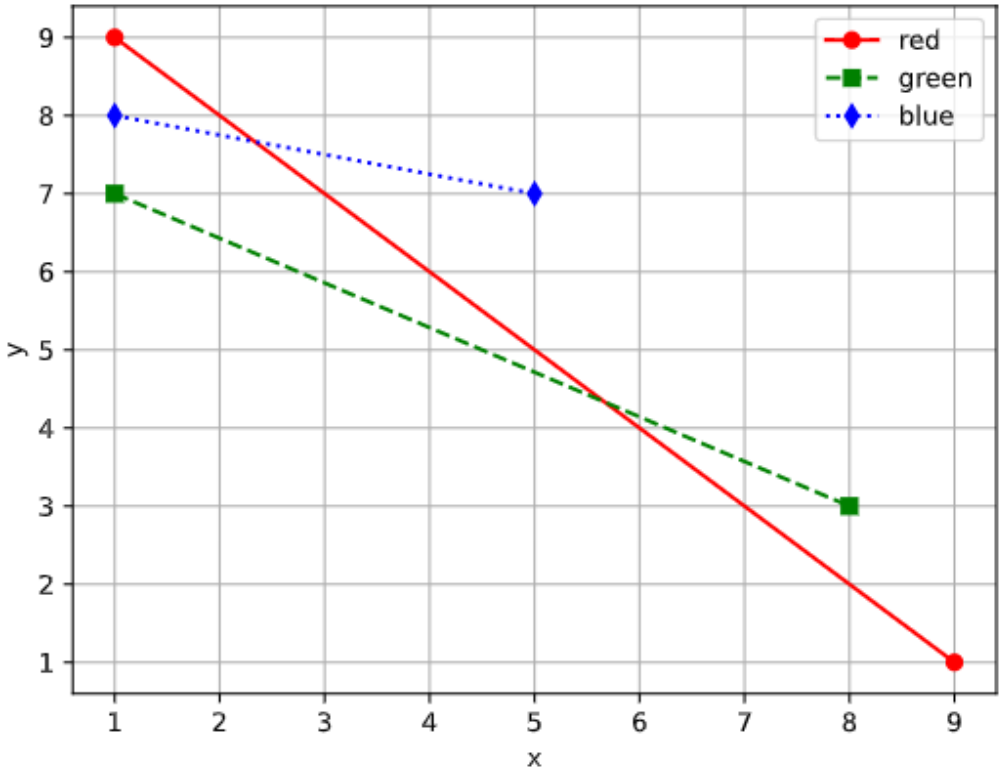
```
import matplotlib.pyplot as plt

x1 = [1, 9]
y1 = [9, 1]

x2 = [1, 8]
y2 = [7, 3]

x3 = [1, 5]
y3 = [8, 7]

legends = ['red', 'green', 'blue']

plt.plot(x1, y1, 'ro-', x2, y2, 'gs--', x3, y3, 'bd:')
plt.legend(legends)
plt.grid()
plt.show()
```




```
$ python grid.py
```

axis_label

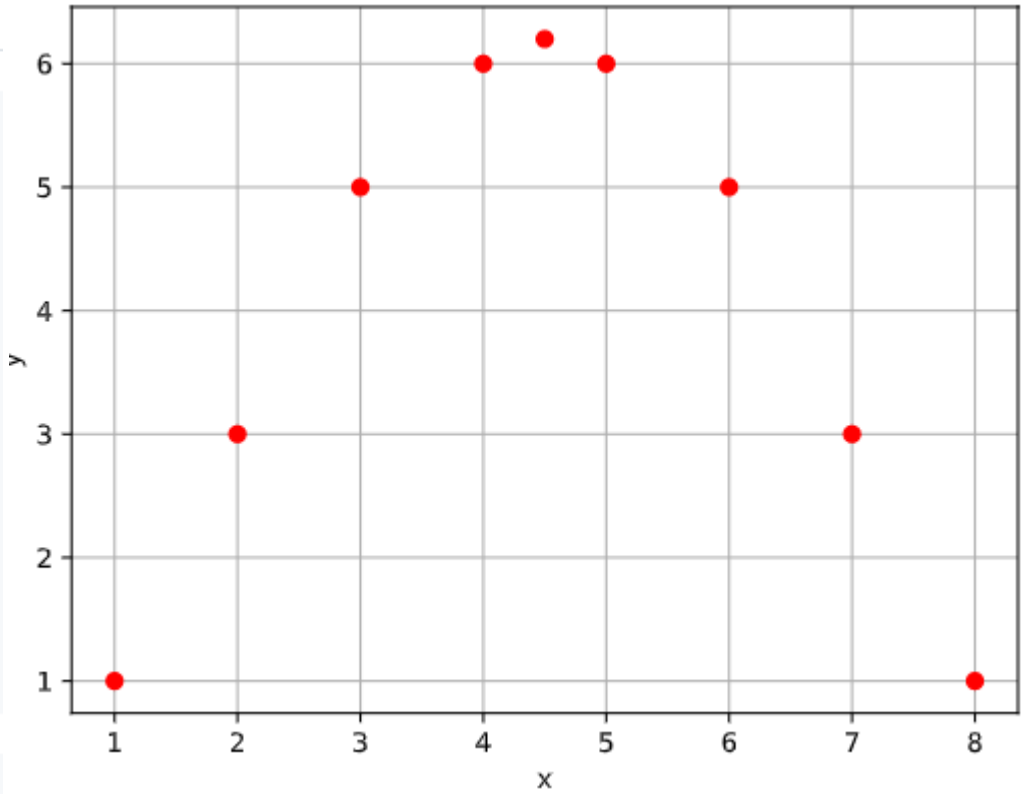
```
import matplotlib.pyplot as plt

x1 = [1, 9]
y1 = [9, 1]

x2 = [1, 8]
y2 = [7, 3]

x3 = [1, 5]
y3 = [8, 7]

legends = ['red', 'green', 'blue']

plt.plot(x1, y1, 'ro-', x2, y2, 'gs--', x3, y3, 'bd:')
plt.legend(legends)
plt.grid()
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```




```
$ python axis_label.py
```

marker_only

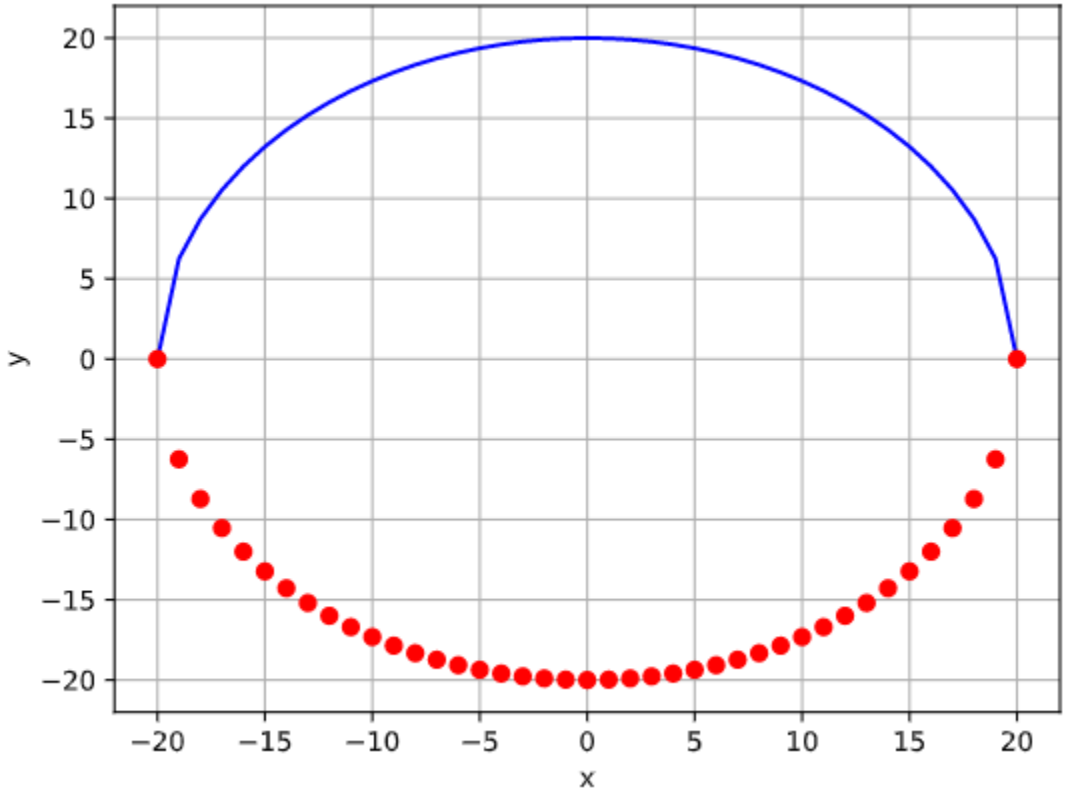
```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 4.5, 5, 6, 7, 8]
y = [1, 3, 5, 6, 6.2, 6, 5, 3, 1]


plt.plot(x, y, 'ro')
plt.grid()
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```




```
$ python marker_only.py
```

line_marker_only

```
import matplotlib.pyplot as plt

r = 20
x = [*range(-r, r+1)]
y = [(r*r - i*i)**0.5 for i in x]
z = [-i for i in y]

plt.plot(x, y, 'b-', x, z, 'ro')
plt.grid()
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

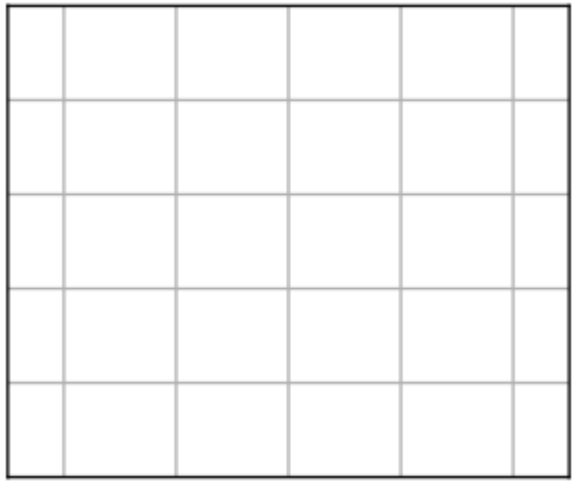



```
$ python line_marker_only.py
```

External package: matplotlib plot types

style_subplots

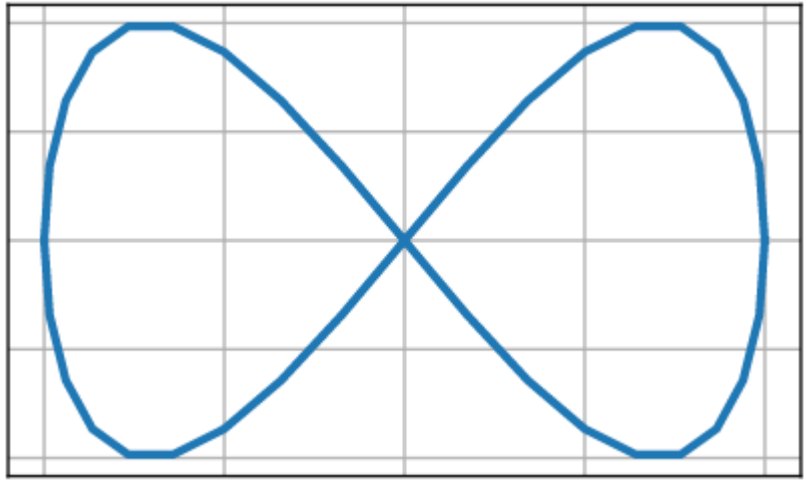
```
import matplotlib.pyplot as plt

plt.style.use("_mpl-gallery")

x = []
y = []

fig, ax = plt. subplots()

plt.show()
```



```
$ python style_subplots.py
```

subplots_plot

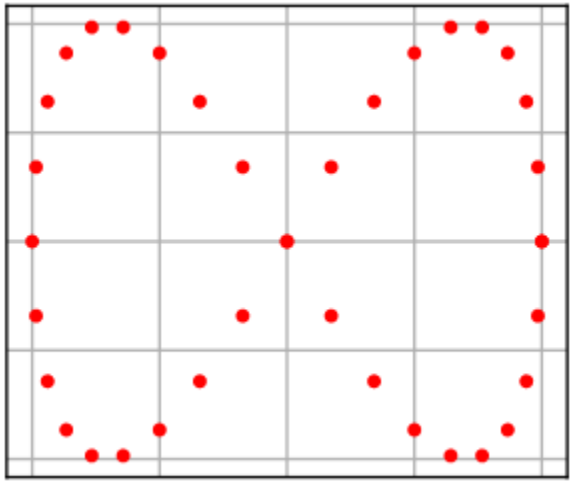
```
import matplotlib.pyplot as plt
import math as m

plt.style.use("_mpl-gallery")

x = [m.cos(m.radians(i*10)) for i in range(37)]
y = [m.sin(2 * m.radians(i*10)) for i in range(37)]

fig, ax = plt. subplots()

ax.plot(x, y, linewidth=2.5)

plt.show()
```



```
$ python subplots_plot.py
```

subplots_scatter

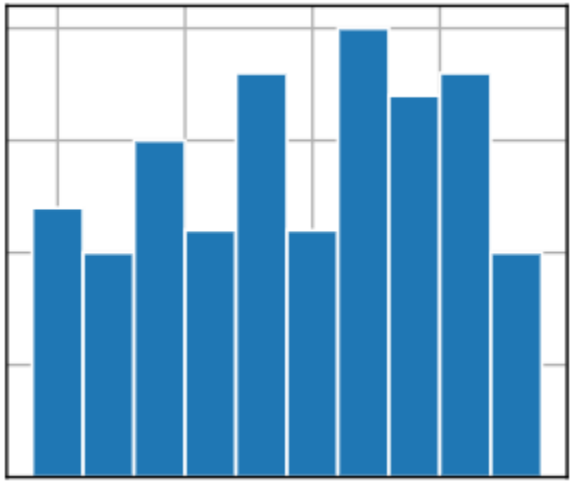
```
import matplotlib.pyplot as plt
import math as m

plt.style.use("_mpl-gallery")

x = [m.cos(m.radians(i*10)) for i in range(37)]
y = [m.sin(2 * m.radians(i*10)) for i in range(37)]

fig, ax = plt. subplots()

ax.scatter(x, y, c='r', s=10)

plt.show()
```



```
$ python subplots_scatter.py
```

subplots_bar

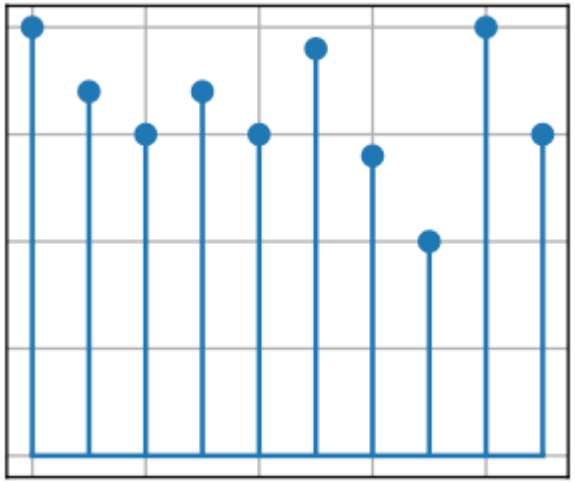
```
import matplotlib.pyplot as plt
import random as r

plt.style.use("_mpl-gallery")

x = [i for i in range(10)]
y = [r.randint(10, 20) for i in x]

fig, ax = plt. subplots()

ax.bar(x, y, width=1, edgecolor="white")

plt.show()
```



```
$ python subplots_bar.py
```

subplots_stem

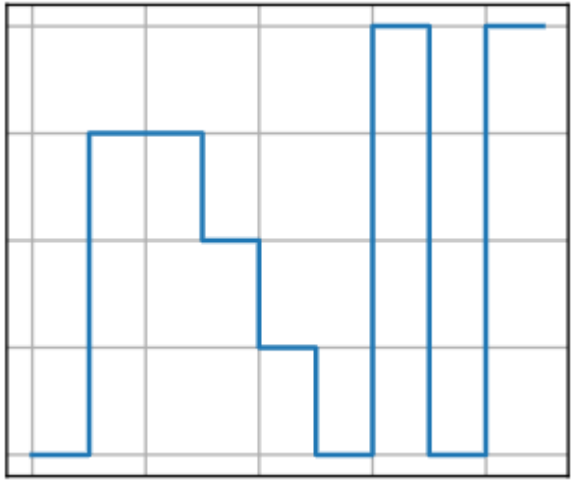
```
import matplotlib.pyplot as plt
import random as r

plt.style.use("_mpl-gallery")

x = [i for i in range(10)]
y = [r.randint(10, 20) for i in x]

fig, ax = plt. subplots()

ax.stem(x, y)

plt.show()
```



```
$ python subplots_stem.py
```

subplots_step

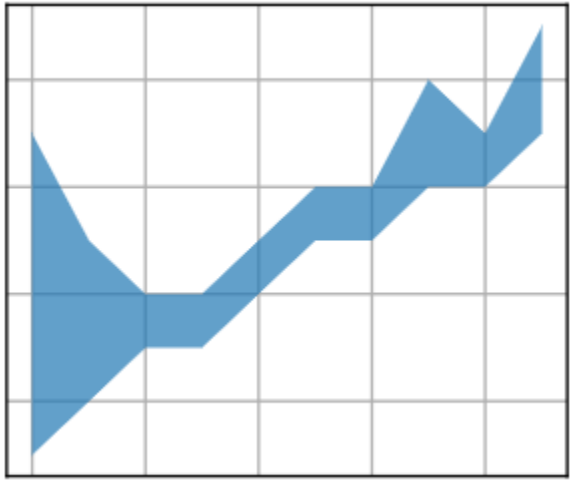
```
import matplotlib.pyplot as plt
import random as r

plt.style.use("_mpl-gallery")

x = [i for i in range(10)]
y = [1, 1, 4, 4, 3, 2, 1, 5, 1, 5]

fig, ax = plt. subplots()

ax.step(x, y)

plt.show()
```



```
$ python subplots_step.py
```

subplots_fill_between

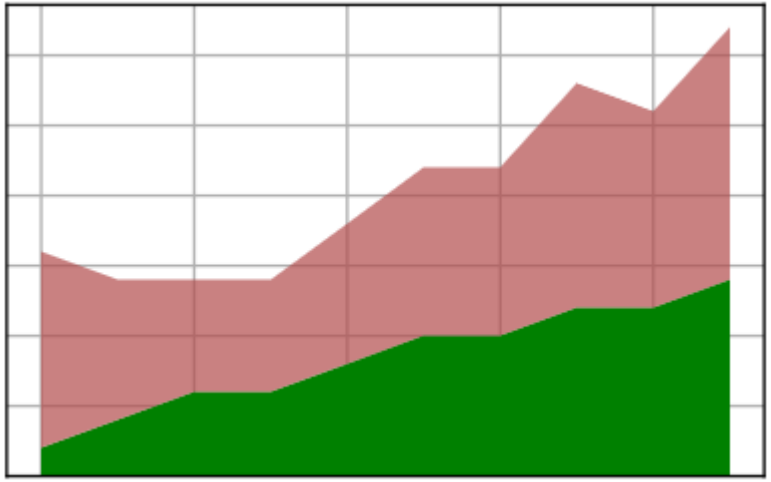
```
import matplotlib.pyplot as plt

plt.style.use("_mpl-gallery")

x = [i for i in range(10)]
y1 = [1, 2, 3, 3, 4, 5, 5, 6, 6, 7]
y2 = [7, 5, 4, 4, 5, 6, 6, 8, 7, 9]

fig, ax = plt. subplots()

ax.fill_between(x, y1, y2, alpha=0.7, linewidth=4)

plt.show()
```



```
$ python subplots_fill_between.py
```

subplots_stackplot

```
import matplotlib.pyplot as plt

plt.style.use("_mpl-gallery")

x = [i for i in range(10)]
y1 = [1, 2, 3, 3, 4, 5, 5, 6, 6, 7]
y2 = [7, 5, 4, 4, 5, 6, 6, 8, 7, 9]

fig, ax = plt. subplots()

alpha = 0.7
ax.stackplot(x, y1, y2, colors=['g', (0.7, 0.3, 0.3, alpha)])

plt.show()
```



```
$ python subplots_stackplot.py
```

Diskusi dan latihan

Tugas sebelum dan setelah kuliah

- Isi kehadiran di SIX.
- Kerjakan tugas yang tersedia di Issue 8.

Diskusi

- Silakan bila ada pertanyaan.
- Setelah kuliah pertanyaan dapat diajukan secara asinkron di url https://github.com/dudung/sk5003-02-2022-2/issues/9

Terima kasih

-