```python
import itertools

result = itertools.permutations(['Ali', 'Hakan', 'Merve'], 2)
print(list(result))
# [('Ali', 'Hakan'), ('Ali', 'Merve'), ('Hakan', 'Ali'),
('Hakan', 'Merve'), ('Merve', 'Ali'), ('Merve', 'Hakan')]

# %%
import itertools

result = itertools.combinations(['Ali', 'Hakan', 'Merve'], 2)
print(list(result))
# [('Ali', 'Hakan'), ('Ali', 'Merve'), ('Hakan', 'Merve')]
# %%
# classes (Sınıflar)
"""
class <isim>:
     <deyimler>
"""
# %%
class Sample:
    x = 10
    print('Sample class')
    def foo(self):
        print('foo')
    def bar(self):
        print('bar')
    print(x)

a = Sample()
a.foo()
a.bar()

"""
Sample class
10
foo
bar
"""
# %%
class Sample:
    def foo(self):
        print('Sample.foo')
class Mample:
    def foo(self):
        print('Mample foo')
```

```
s = Sample()
m = Mample()

s.foo()
m.foo()

Sample.foo(s)
Mample.foo(m)
# %%
class Point:
    pass
pt = Point()
pt.x = 10
pt.y = 20
print("x = {}, y = {}".format(pt.x, pt.y))
# x = 10, y = 20

# %%
class Sample:
    pass
s1 = Sample()
s1.x = 10
s1.y = 20

s2 = Sample()
s2.a = 30
s2.b = 'Harun'
s2.c = 12.3

print('x = {}, y = {}'.format(s1.x, s1.y))
print("a = {}, b = {}, c = {}".format(s2.a, s2.b, s2.c))
"""
x = 10, y = 20
a = 30, b = Harun, c = 12.3
"""
# %%
class Point:
    def set(self, x, y):
        self.x = x
        self.y = y
    def disp(self):
        print(self.x, self.y)

pt1 = Point()
pt2 = Point()
```

```
pt1.set(10, 20)
pt2.set(30, 40)
pt1.disp()
pt2.disp()
"""
10 20
30 40
"""
# %%
class Date:
    def set(self, day, month, year):
        self.day = day
        self.month = month
        self.year = year
    def disp(self):
        print("{}/{}/{}".format(self.day, self.month,
self.year))

date = Date()
date.set(29, 4, 2025)
date.disp()
# 29/4/2025
# %%
# dunder (double underscore)   __xxxx__

class Sample:
    def __init__(self):
        self.a = 10
        self.b = 20
    def disp(self):
        print(self.a, self.b)

x = Sample()
x.disp()
# 10 20
# %%
class Date:
    def __init__(self):
        self.day = 1
        self.month = 1
        self.year = 1900
    def disp(self):
        print("{}/{}/{}".format(self.day, self.month,
self.year))

date = Date()
# date.disp() # 1/1/1900
```

```
Date.disp(date) # 1/1/1900
# %%

class Sample:
    def __init__(self, a, b):
        self.a = a
        self.b = b
    def disp(self):
        print(self.a, self.b)
x = Sample(10, 20)
x.disp()
# 10 20

# %%
class Student:
    def __init__(self, name, no):
        self.name = name
        self.no = no
    def disp(self):
        print("İsim = {}, No = {}".format(self.name, self.no))
s = Student('Esra Bal', 123)
s.disp()
# İsim = Esra Bal, No = 123
# %%
import datetime

d = datetime.date(2025, 4, 29)
print('{}/{}/{}'.format(d.day, d.month, d.year))
# 29/4/2025
# %%
import datetime

d = datetime.date(2025, 4, 29)
print('{}/{}/{}'.format(d.day, d.month, d.year))
days = ['Pazartesi', 'Salı', 'Çarşamba', 'Perşembe', 'Cuma',
'Cumartesi', 'Pazar']
wd = d.weekday()
print(days[wd])  # Salı
# %%
from datetime import datetime
dt = datetime(2025, 4, 29, 11, 35, 48)
print(dt.hour, dt.minute, dt.second)  # 11 35 48
# %%
# Türetme (Kalıtım (inheritance))
class A:
    def foo(self):
        print('foo')
```

```
    def bar(self):
        print('bar')
class B(A):
    def tar(self):
        print('tar')
b = B()
b.foo()
b.bar()
b.tar()
"""
foo
bar
tar
"""
# %%
class A:
    def setA(self, a):
        self.a = a
    def dispA(self):
        print(self.a)
class B(A):
    def setB(self, b):
        self.b = b
    def dispB(self):
        print(self.a, self.b)
b = B()
b.setA(10)
b.setB(20)
b.dispA()
b.dispB()
"""
10
10 20
"""
# %%
class A:
    def setA(self, a):
        self.a = a
    def dispA(self):
        print(self.a)
class B(A):
    def setB(self, b):
        self.b = b
    def dispB(self):
        print(self.a, self.b)
b = B()
b.setB(20)
```

```
b.dispA() # error
b.dispB()
# %%
class A:
    def __init__(self, a):
        self.a = a
    def dispA(self):
        print(self.a)
class B(A):
    def __init__(self, a, b):
        A.__init__(self, a)
        self.b = b
    def dispB(self):
        print(self.b)
b = B(10, 20)
b.dispA()
b.dispB()
"""
10
20
"""
# %%
class A:
    def __init__(self, a):
        self.a = a
    def dispA(self):
        print(self.a)
class B(A):
    def __init__(self, a, b):
        super(B, self).__init__(a)
        self.b = b
    def dispB(self):
        print(self.b)
b = B(10, 20)
b.dispA()
b.dispB()
"""
10
20
"""
```