```
def sum (a, b, c):
    return a+b+c
```

a = 3
b = 4
c = 18

sum (3, 4, 18)

```
def sum_args (*args):
    sum=0
    for i in args:
        sum += i
    retur sum
```

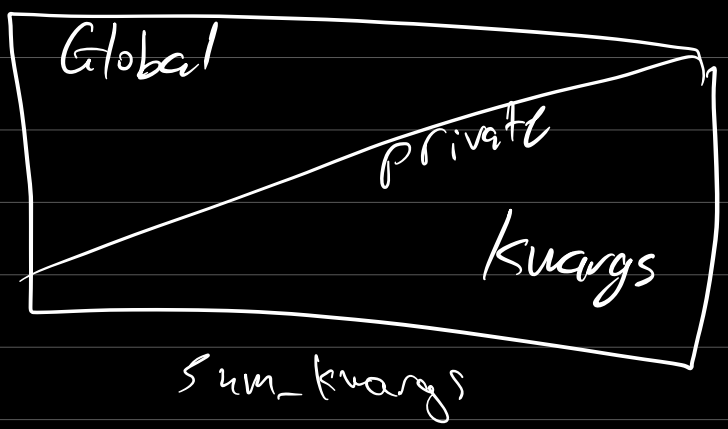
args = [3, 4, 18, 5, 13, -1]

sum_args (3, 4, 18, 50, 13, -1)
sum_args (3, 4, 4)

```
def sum_kwargs (**kwargs)
    return kwarg['a'] + kwargs['b'] + kwarg['b']
```

sum_kwargs (x=3, y=8, a=40, h=0, b=38)

kwargs = {'x': 3, 'y': 8, 'a': 40, 'h': 0, 'b': 38}



OOP

dog

data
age name
kind color
price

functionality

barke
eat
dring
sleep
pee

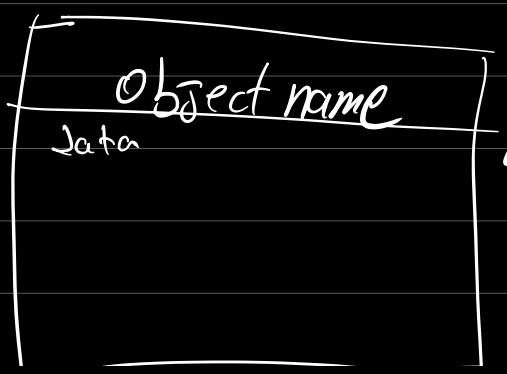
Car

Data

- kind
- make
- Year
- color
- number_of_wheels
- hp
- cc

func

- forward
- Reverse
- brake
- turn on
- turn off
- horn
- open SunRoof



UML

Dog

Data

name : str var

age : int

kind : str

func :

bark()

Run()

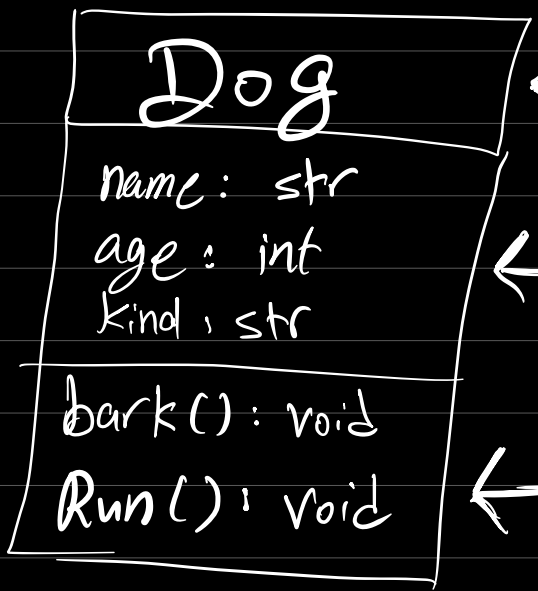
data

variables

func

functions

blueprint



object Name

Data

functionality

str x = "a":

→ d1 = Dog()

d2 = Dog()

d1.bark() → method

x.upper() ← method

len('a') → func

file1.py

```
def sum(x,y):
    return x+y

sum(3,4)

class math:
    def sum(x,y):
        return x+y
n = math()
n.sum(3,4)
```

function

← method

len('str') → length of string

'str'.lower → Chang the current object and return a lower case str

x= [3,4,8]

x.append(11)