**Sınıflar-Nesneler(Python Classes/Objects)**

Python, nesne yönelimli bir programlama dilidir.

self Parametre sınıfı mevcut örneği için bir referanstır ve sınıfa ait erişim değişkenleri için kullanılır. Herhangi bir işlevin ilk parametresi olmalıdır.Self yerine başka ifadeler de kullanılabilir.

**\_\_**

**İnit \_\_ () İşlevi**

Tüm sınıfların **\_\_init \_\_ ()** adında bir işlevi vardır ve bu işlev sınıf başlatılırken her zaman çalıştırılır.

Nesne özelliklerine değerler atamak için \_\_init \_\_ () işlevini kullanırız.

**class Player:**

**def \_\_init\_\_(self, name, age,position):**

**self.name = name**

**self.age = age**

**self.position=position**

**def printPlayer(self):**

**print("Name: " + self.name+" "+"Age: "+str(self.age)+" "+"Position: "+self.position)**

**p1 = Player("Ali", 32,"Middle")**

**p2=Player("Mehmet",27,"Goalkeaper")**

**p1.printPlayer()**

**p2.printPlayer()**

**Kalıtım(Python Inheritance)**

Miras, tüm yöntemleri ve özellikleri başka bir sınıftan miras alan bir sınıfı tanımlamamıza olanak tanır.

**Ana sınıf** , temel sınıf olarak da adlandırılan, miras alınan sınıftır.

**Çocuk sınıfı** başka bir sınıftan devralır, aynı zamanda elde edilen sınıf denilen sınıftır.

**class Player:**

**def \_\_init\_\_(self, name, age,position):**

**self.name = name**

**self.age = age**

**self.position=position**

**def printPlayer(self):**

**print("Name: " + self.name+" "+"Age: "+str(self.age)+" "+"Position: "+self.position)**

**class substitute(Player):**

**def \_\_init\_\_(self, name, age,position):**

**Person.\_\_init\_\_(self, name, age,position)**

**x = substitute("Ahmet",19,"Forward")**

**x.printPlayer()**

## Super () İşlevi

super()alt sınıfın tüm yöntemleri ve özellikleri üst sınıfından devralmasını sağlayacak bir işleve sahiptir. Metotları ve özellikleri otomatik olarak üst öğesinden devralacağı için self kullanmamıza gerek yok.

**class substitute(Player):**

**def \_\_init\_\_(self, name, age,position):**

**super().\_\_init\_\_(name, age,position)**

**class Person:**

**def \_\_init\_\_(self, fname, lname):**

**self.firstname = fname**

**self.lastname = lname**

**def printname(self):**

**print(self.firstname, self.lastname)**

**class Student(Person):**

**def \_\_init\_\_(self, fname, lname, year):**

**super().\_\_init\_\_(fname, lname)**

**self.gradyear = year**

**def welcome(self):**

**print("Welcome", self.firstname, self.lastname, "to the class of", self.gradyear)**

**x = Student("Burak", "Aydın", 2018)**

**x.welcome()**

# Python Scope

## Yerel Scope(Local Scope) : Sadece tanımlandığı kısmın içinde erişilebilir.

**def func1():**

**a = "Deneme"**

**print(a)**

**func1()**

## Global Scope: Her yerden erişilebilir.

**a = "Deneme"**

**def func1():**

**print(a)**

**func1()**

**print(a)**

**a= 700**

**def func1():**

**a = 200**

**print(a)**

**func1() #200**

**print(a) #300**

**Global Anahtarı(Global Keyword)**

**a=700**

**def func1():**

**global a**

**a= 200**

**func1()**

**print(a) #200**

**Modüller(Python Modules)**

İlk olarak mymodule.py adlı bir dosya oluşturalım ve bu kodları yazalım :

def say\_hello(name):  
  print("Hello, " + name)  
  
person1 = {  
  "name": "Ali",  
  "age": 27,  
  "country": "Turkey"  
}

Daha sonra mymodule\_2.py adlı diğer bir dosya oluşturalım ve bu kodları yazalım.

import mymodule as mx  
  
a = mx.person1["age"]  
print(a) #27 yazar.

from mymodule import person1  
  
print (person1["age"])

#Burada mymodule adlı dosyadan person1’i çekerek işlem yaptık ve onu yazdırdık. 27 yazar.

# Tarihler(Python Datetime)

import datetime yazarak işe başlayalım.

**Tarih Oluşturmak**

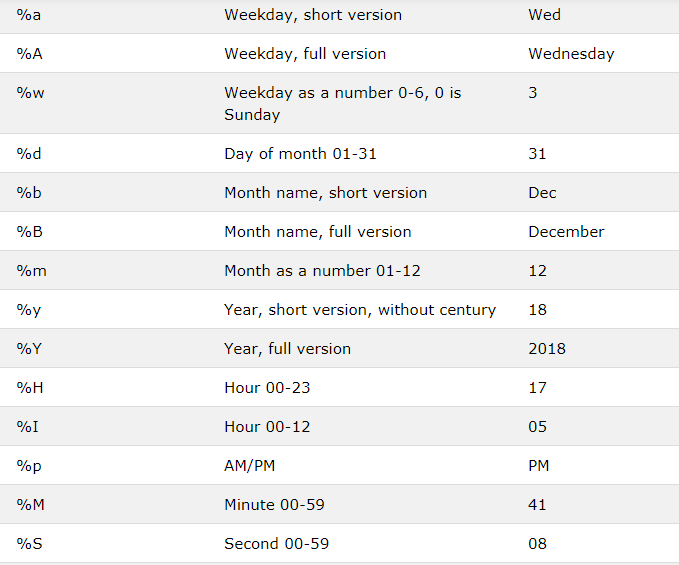
import datetime  
  
x = datetime.datetime(2020, 5, 17) #17.05.2020 tarihini oluşturduk  
  
print(x)

## Strftime () Yöntemi

import datetime  
  
x = datetime.datetime(2020, 5, 17)  
  
print(x.strftime("%A")) #Oluşturulan yönteme %A metodu uygulandı.

#Sunday yazar.

Metotlar:



**Matematiksel İşlemler(Python Math)**

## Yerleşik Matematik İşlevleri

x = min(61, 14, 25,11) #11

y = max(3, 27, 37) #37

print(x)

print(y)

x = abs(-4.75) #Mutlak değer alınır. 4.75 yazar.  
  
print(x)

x = pow(6, 3) #Üs işlemi. 6 üzeri 3  
  
print(x) #216

## Matematik Modülü

## import math x = math.sqrt(25) $Kök 25 print(x) $5

x = math.ceil(1.4) #üste yuvarlar  
y = math.floor(1.4) #alta yuvarlar.  
  
print(x) # returns 2  
print(y) # returns 1

Bu komutların devamını Python Math Modules yazıp bulabilirsiniz.

## Python Hata Blokları( PythonException Handling)

tryBlok hatalar için kod bloğu test etmenizi sağlar.

exceptBlok Hatayı ele almanızı sağlar.

finallyBlok ne olursa olsun çocuğunun EEG ve blokların dışında sonucunun, kod yürütmesine olanak tanır.

try:

print(x)

except NameError:

print("Değişken x tanımlanmadı.")

#x tanımlanmadığı için burası çalışır.

except :

print(“Başka bir hatayla karşılaşıldı.”)

# Python Kullanıcıdan Veri Girişi Alma(Python Input From Users)

Bu işlem için;

Python 3.6 input()yöntemi kullanır .

Python 2.7, raw\_input()yöntemi kullanır .

username = input("Enter username:")  
print("Username is: " + username)