Python OOP

Christoffer Berglund

September 13, 2018

Overview

- Short repetition
  - Variables, arrays, if
- Running other application
- OOP programming repetition
- Functions, demo
- Classes, demo
- GUI(if time)

Short repetition

Variables

```
age = 29
name = "Christoffer"
print(name + " is " + str(age) + " years old")
print(name,"is",age,"years old")
```

Asking from a user

```
age = input('What is your age')
nextyear = int(age) +1
print("Next year you are:",nextyear)
```

if

```
age = 32
if(age>32):
    print("You are old")
else:
    print("You are young")
```

arrays

list / arrays

bash (array):

```
os=('linux' 'mac')
for i in "${os[@]}"
do
    echo $i
done
```

python (list):

```
os = ['linux','mac']
for i in os:
    print(i)
```

making lists

splitting lists:

```
ip = '192.168.1.1'
l = ip.split('.')
```

getting one item in a list:

```
l[0]
```

getting more than one item

```
l[0:3]
```

5 Minute Assignment

- You have the text "This is a long sentence"
- Split the sentence and print out every word in a for loop
- Start with the following code:

```
s = "This is a long sentence"
```

Solution

```
s = "This is a long sentence"
l = s.split(' ')
for i in l:
    print(i)
```

For loops

```
tekst = "abcdef"
for bokstav in tekst:
    print(tekst)
```

or

```
for i in range(0,10):
    print(i)
```

Double loops

```
for i in 'ab':
   for j in 'ab':
     print(i,j)
```

5 Minute Assignment

- Create a small application that generates every 4-digit pin code available.

Solution

```
for i in range(0,10):
    for j in range(0,10):
        for k in range(0,10):
            for l in range(0,10):
                print(i,j,k,l)
```

Dictionaries (Associative Arrays)

Bash:

```
declare -A user
user=([mortengo]="Morten Goodwin" [sigurda]="Sigurd Assev")
for i in "${!user[@]}"
do
  echo "key  : $i"
  echo "value: ${user[$i]}"
done
```

Python:

```
passwordlist = {'mortengo':'nd2978ey', 'sigurda':'hemmelig123'}
for user,password in passwordlist.items():
    print(user)
    print(password)
```

Using dictionaries

```
passwordlist = {'mortengo':'nd2978ey', 'sigurda':'hemmelig123'}
username = input('Skriv ditt brukernavn')
print("Ditt passord er",passwordlist[username])
```

5 Minute Assignment

- Create a dictionary with two/three Norwegian and English words.
- Let the user enter a Norwegian word and translate it to English.

Solution

```
ordbok = {'hei':'hi', 'bil':'car', 'rom':'room'}
norsk = input('Skriv norsk ord')
print ("Det engelske ordet er:",ordbok[norsk])
```

Running other applications

Running other applications

Bash:

```
ls
```

Python:

```
import os
os.system("dir")
```

Checking success:

```
import os
success = os.system("dir")
if(success==0):
    print('success')
```

Reding into a variable

Windows:

```
import subprocess
proc = subprocess.Popen("dir c:",
                        shell=True,
                        stdout=subprocess.PIPE,
                        stderr=subprocess.PIPE)
print(proc)
```

Reding into a variable

Windows:

```
import subprocess
proc = subprocess.Popen("dir c:",
                        shell=True,
                        stdout=subprocess.PIPE,
                        stderr=subprocess.PIPE)
#print(proc)  This is wrong because this is an object
out, err = proc.communicate()
print(out)
```

More OS

Windows:

```
os.environ['HOMEPATH']
```

Linux:

```
os.environ['HOME']
```

5 Minute Assignment

- Develop a script that runs ping towards another host.
- The user should provide the IP to ping.

Solution

```
import os
ip = input('Enter an IP to ping')
os.system('ping '+ip)
```

Even more os

File handling:

```
os.chdir(...)
os.mkdir(...)
os.remove(...)
os.rename(...)
os.stat(...)
```

Grabbing stdout from application

bash:

```
a=$(ls)
echo $a
```

python:

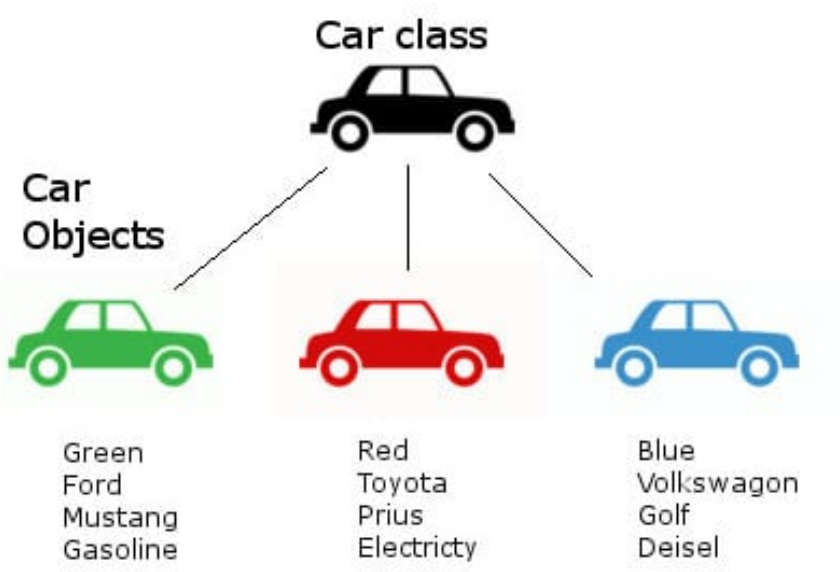
```
p = subprocess.Popen('ls',stdout=subprocess.PIPE)
d = p.stdout.readline()
while(d!=b''):
    print(d)
    d = p.stdout.readline()
```

OOP programming repetition

Python dynamically strongly typed

- Dynamically
  - Does not restrict variable usage
  - No need to declare variable type
- Strongly typed
  - Interpreter keeps track of variables
  - Binds variables to types

Classes and objects



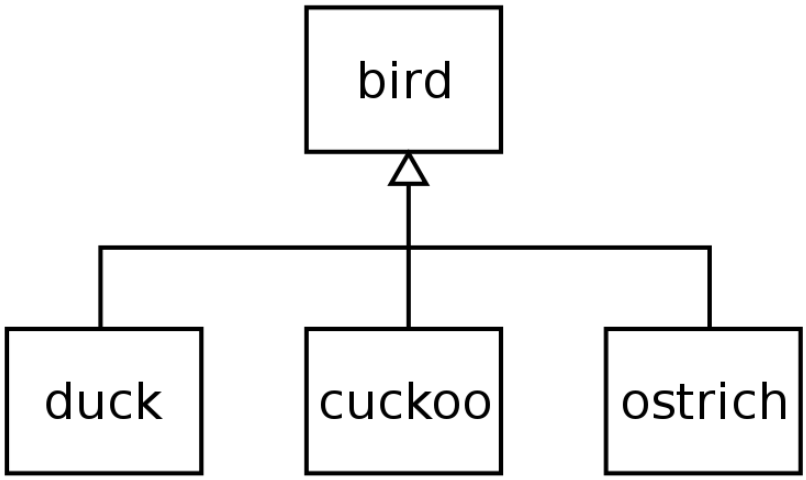
http://www.wordane.com/2016/07/object-oriented-programming-classes-vs-objects.html

Constructor

In python: `__init__`

- Called once when creating an instance
- Populating the instance

Inheritance



https://en.wikipedia.org/wiki/Subtyping

GUI

- Aim: GUI with button that executes ping.

Simple GUI

```
from tkinter import *

class Vindu(Tk):
    def __init__(self):
        Tk.__init__(self,None)

Vindu()
input()
```

Entry text

```
from tkinter import *

class Vindu(Tk):
   def __init__(self):
      Tk.__init__(self,None)
      entry = Entry()
      entry.grid(column=0,row=0)
Vindu()
input()
```

Button

```
from tkinter import *

class Vindu(Tk):
   def __init__(self):
      Tk.__init__(self,None)
      entry = Entry()
      entry.grid(column=0,row=0)
      knapp = Button(text="Press me")
      knapp.grid(column=0,row=1)
Vindu()
input()
```

5 Minute Assignment

- Make a GUI with:
  - Entry form for writing IP
  - Button for executing ping

Button

```
from Tkinter import *

class Vindu(Tk):
   def __init__(self):
      Tk.__init__(self,None)
      entry = Entry()
```

```
        entry.grid(column=0,row=0)
        knapp = Button(text="Execute ping")
        knapp.grid(column=0,row=1)
Vindu()
input()
```

Working button

```
from tkinter import *

class Vindu(Tk):
    def __init__(self):
        Tk.__init__(self,None)
        entry = Entry()
        entry.grid(column=0,row=0)
        knapp = Button(text="Execute ping",command=self.runping)
        knapp.grid(column=0,row=1)
    def runping(self):
        import os
        os.system('ping localhost')
Vindu()
input()
```

Reading from text field

```
from tkinter import *
class Vindu(Tk):
    def __init__(self):
        Tk.__init__(self,None)
        self.entry = Entry()
        self.entry.grid(column=0,row=0)
        knapp = Button(text="Execute ping",command=self.runping)
        knapp.grid(column=0,row=1)
    def runping(self):
        import os
        os.system('ping '+self.entry.get())
Vindu()
input()
```