¿Enseñar a programar con python o con python enseñar a programar?

Por Lucas Spigariol

Objetivo o Herramienta

¿es el lenguaje una herramienta para el objetivo de aprender a programar bien, o son los conceptos de programación herramientas para el objetivo de aprender bien el lenguaje?

Diferentes formas de programar

Python como herramienta versátil

Formas de pensar un sistema

- Procedural
- Objetos
- Funcional

Procedural

Inspirada en lenguajes

de la familia de C

```
def mayoresDeEdad(personas):
  mayores = [None]*len(personas)
  j = 0
  for i in range(len(personas)):
    if personas[i][1] > 18:
      mayores[j] = personas[i]
      j+=1
  return mayores[:j]
```

```
mayoresDeEdad([["juan",19],["ana",15],["dani",20]])
```

```
[['juan', 19], ['dani', 20]]
```

Procedural

Con un poco más de abstracción y delegando

```
def mayoresDeEdad(personas):
    mayores = [ ]
    for persona in personas:
        if  esMayor( persona):
            mayores.append(persona)
    return mayores
```

```
def esMayor(persona):
    return persona[0] > 18
```

Procedural

Con mayor conocimiento del lenguaje

```
def mayoresDeEdad(personas):
  return  [persona for persona in personas if esMayor(persona) ]
```

Funcional

Usando funciones de orden superior

```
def mayoresDeEdad (personas ):
  return list(filter(esMayor, personas ))
```

Procedural / Funcional

Representando a un tipo de dato con tuplas en vez de listas

```
def esMayor (persona):
  return persona[1] >= 18

jn = ("juan",19)
an = ("ana",15)
da = ("dani",20)
mayoresDeEdad([jn,an,da])
```

Procedural

Con diccionarios

```
def esMayor(persona):
  return persona["edad"] >= 18

jn = {"nombre":"juan", "edad":19}
an = {"nombre":"ana", "edad":15}
da = {"nombre":"dani", "edad":20}
mayoresDeEdad([jn,an,da])
```

Objetos

Reemplazar por un objeto, sin tener que modificar la funcion principal

```
class Persona:
  def __init__(self, nombre, edad):
    self.nombre = nombre
    self.edad = edad

def esMayor(persona):
  return persona.edad >= 18
```

Objetos

O haciéndolo más "objetoso", modificando la función principal

```
def mayoresDeEdad(personas):
  mayores = [ ]
  for persona in personas:
    if persona.esMayor():
      mayores.append(persona)
  return mayores
```

```
class Persona:
  def __init__(self, nombre, edad):
    self.nombre = nombre
    self.edad = edad

  def esMayor(self):
    return self.edad >= 18
```

Objetos

Incorporando polimorfismo

```
class Robot:
  def __init__(self, año):
    self.año = año

  def esMayor(self):
    return self.año < 2000
```

```
class Persona:
  def __init__(self, nombre, edad):
    self.nombre = nombre
    self.edad = edad

  def esMayor(self):
    return self.edad >= 18
```

```
len(mayoresDeEdad([Persona("juan",19),Robot(1999),Persona("dani",20)]))
```

```
3
```