

Programación Orientada a Objetos

Daniel González Duque

Historia

- **1960s SIMULA 67** - Kristen Nygaard and Ole-Johan Dahl in Norway
- **El término “Programación Orientada a Objetos”** – Xerox PARC en Smalltalk
- **1980s** – C++, Interfaces gráficas.

¿Qué es un objeto?

“Objects are like people. They’re living, breathing things that have knowledge inside them about how to do things and have memory inside them so they can remember things. And rather than interacting with them at a very low level, you interact with them at a very high level of abstraction, like we’re doing right here.” (Jobs, 1994)

Python

Un poco del programa

- **1980s** – Creado por Guido Van Rossum (CWI)
- **Lenguaje Multipropósito**
- **Se puede instalar en todos los sistemas operativos**
- **Contiene una terminal interactiva (Python, iPython)**

Tipos de variables

- Entero – **Int()**
- Real – **float()**
- Cadena de Caracteres – **str()**
- Booleano – **True()** ó **False()**

Tipos de estructuras de datos

- Constantes – **C = 3** ó **C = 'abs'**
- Listas – **Lista = []** ó **Lista = list()**
- Diccionarios – **Dicc = {key: <objeto> }** ó **Dicc = dict()**
- Tupla – **T = ()** ó **T = tuple()**

NOTA 1: Dentro de las listas y los diccionarios puede haber lo que sea.

NOTA 2: Una vez creada una Tupla esta no se podrá cambiar internamente.

Funciones cadenas de caracteres

- **upper()** -> Todo el texto en Mayúsculas.
- **lower()** -> Todo el texto en minúsculas.
- **split('<string>')** -> Separa la cadena en una lista.
- **replace('<1>,<2>')** -> Remplaza el valor 1 por el valor 2.

Funciones listas

- **append()** -> Agregar una nueva posición.
- **pop()** -> Elimina el último valor o la posición que se indique.
- **index('<valor>')** -> Busca un valor en particular y dice la posición.

Buscar más funciones.

Condicionales

`if` <Condicional>:

 <Operaciones dentro del condicional>

<Operaciones fuera del condicional>

Condicionales

- $==$ -> Igualdad
- $!=$ -> Desigualdad
- $>=$ -> Mayor o igual que
- $<=$ -> Menos o igual que
- $>$ -> Mayor a
- $<$ -> Manor a

Condicionales

if <Condicional>:

<Operaciones dentro del primer condicional>

elif <Condicional>:

<Operaciones dentro del segundo condicional>

else:

<Operaciones dentro del tercer condicional>

<Operaciones fuera del condicional>

Ciclo For

```
for i in range(<VI>,<VF>-1):  
    <Operaciones dentro del ciclo>  
<Operaciones fuera del ciclo>
```

- El ciclo **for** itera sobre **arreglos de datos**.

Ciclo while

```
while <Condición>:  
    <Operaciones dentro del ciclo>  
<Operaciones fuera del ciclo>
```

- El ciclo **while** depende de una condición verdadera.
- Puede causar ciclos infinitos.

Referencias

- Huan, J. (2017). A Brief History of Object-Oriented Programming. Recuperado 25 de enero de 2018, a partir de <http://web.eecs.utk.edu/~huangj/CS302S04/notes/oo-intro.html>
- Jobs, S. (1994). Rolling Stones Magazine.