



Fundamentos de Lenguajes de Programación

Introducción a Ciencias de la Computación (CS)
Est. Johel Heracio Batista Cárdenas



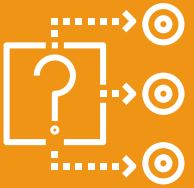
@batistajohel



@batistajohel



Propósito y Objetivo en esta Vida



Entender, Comprender, Analizar y Reflexionar acerca de los diferentes tipos de Lenguajes de Programación, sus características y por qué Python es tan hermoso.

Un poco de la Historia Pythonística



¿Quién fue el creador del Lenguaje de Programación Python?

1. Carl Sagan
2. Darth Vader
3. Cuenta Ventanas (Bill Gates)
4. Guido van Rossum
5. Linus Torvalds
6. Marques Brownlee (MQBHD)

**¿QUE GUARDA DARTH
VADER EN SU HELADERA?**

HELADO OSCURO

La Fuerza Hemos de Necesitar.

Un poco de la Historia Pythonística



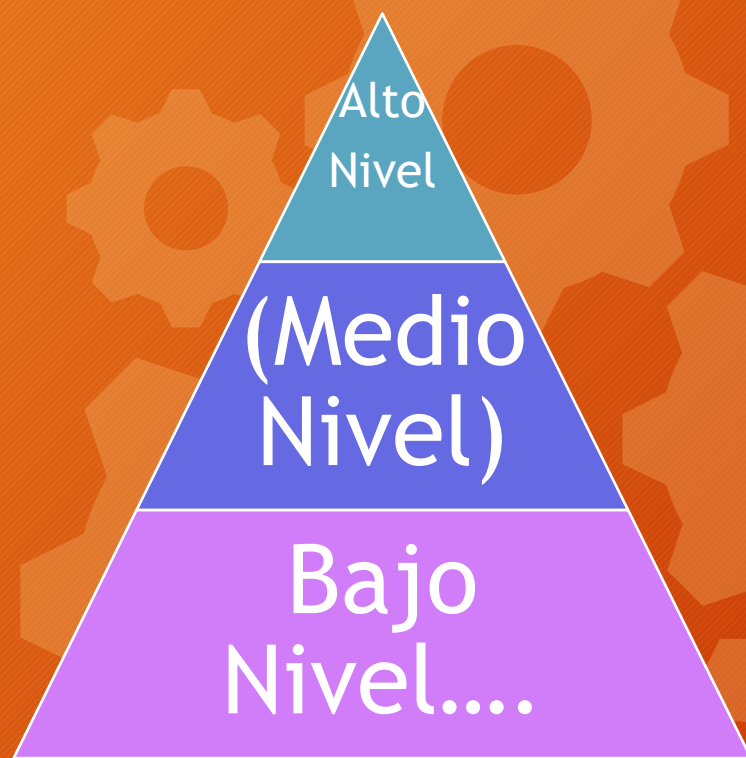
Guido van Rossum bajándose una buena pinta en 2006.

¿Quién fue el creador del Lenguaje de Programación Python?

1. Carl Sagan
2. Darth Vader
3. Cuenta Ventanas (Bill Gates)
4. R/. Guido van Rossum, en 1989 aprox.
5. Linus Torvalds
6. Marques Brownlee (MQBHD)



KennyBell: Lenguajes de Programación



Una pirámide (Y no de Forex), de los tipos de Lenguajes de Programación

Algunos autores sugieren que los Lenguajes de Programación se clasifican en 3 Niveles

1. Lenguajes de Bajo Nivel

1. Lenguaje de Máquina
2. Lenguaje Ensamblador (Assembler)

2. Lenguajes de Medio Nivel

1. Lenguaje C

3. Lenguajes de Alto Nivel

1. Python, Java, Javascript, Ruby on Rails, C#, etc.

KennyBell: Bajo (Programación)



*Solo hay 1, dentro de 10 personas que
se rieron de este meme*

Un Lenguaje de Programación de Bajo Nivel, solamente lo entiende la computadora; ya que utiliza el Sistema Binario, es decir 0 y 1.

Pero, clasifiquémoslos en dos tipos:

1. **Lenguaje de Máquina:** Son instrucciones de únicamente 0's y 1's que se ejecutan directo en el CPU → ALU (Arithmetical Logical Unit).
2. **Lenguaje Ensamblador (Assembler):** Es un derivado del Lenguaje de Máquina, pero utiliza letras y números para sus instrucciones.
 1. Igual se necesita un compilador para poder pasar de las **Código de Assembler** → **Lenguaje de Máquina**



KennyBell: Medio (Programación)

- Podríamos decir que son lenguajes que nos permiten una mayor abstracción, pero mantienen características de un lenguaje de bajo nivel.
- Su código es enviado a un compilador que luego lo convierte a “Lenguaje de Máquina”.
- El Lenguaje C es el mejor ejemplo, porque puede acceder a los registros del sistema, direcciones de memoria y mucho más.



*Un Stormtrooper frustrado porque él
solo C que nada C.*

KennyBell: Alto (Programación)



Los lenguajes de Alto Nivel los más cercanos a nuestro lenguaje humano, ya que usan conceptos (Variables, datos, lógica, etc).

Ahora hay un par de temas sobre eso:

- No podemos utilizarlos sin un intérprete o compilador que traduzca nuestro código al “**Lenguaje de Máquina**”
 - Existen dos tipos de Lenguajes de Alto Nivel
 - Interpretados: PHP, Python, Javascript, etc.
 - Compilados: C, C#, C++, Visual Basic, Java, etc.

Se siente representado



¿Cómo es eso de Interpretado y Compilado?

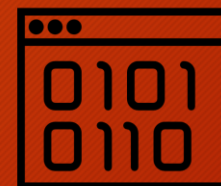
Lenguajes
Compilados

Código
Fuente

Compilador

Código de
Máquina

Salida



Lenguajes
Interpretados

Código
Fuente

Intérprete

Byte Code

Salida



```
>>> import dis # "dis" - Desensamblador de código d
>>> dis.dis('print("¡Hola, Mundo!")')
1 0 LOAD_NAME      0 (print)
2 0 LOAD_CONST     0 ('¡Hola, mundo!')
4 0 CALL_FUNCTION  1
6 0 RETURN_VALUE
```

CIVIL WAR: Interpretados vs. Compilados



Lenguajes Compilados	Lenguajes Interpretados
Traducen a código de máquina y crean un archivo traducido que se ejecuta más rápido	El código se traduce línea por línea, cada vez que se ejecute el programa
Necesita que las instrucciones sean traducidas, así que esto lo hace más “eficiente”	Usualmente son entre 10-20 veces más lentos que los lenguajes compilados
Depende de un compilador diseñado para dicho lenguaje.	Muchísima mayor independencia de plataformas computacionales.
Mayor tamaño de programas	Menor tamaño de programas



Vs.





Introducción a Python 3.0

Introducción a Ciencias de la Computación (CS)
Est. Johel Heraclio Batista Cárdenas



@batistajohel

¿Qué es eso del Lenguaje Python?



- Es un lenguaje de programación de Alto Nivel, creado por Guido van Rossum, aproximadamente en 1989.
- Tiene como principal característica que es “*Multiparadigma*”, lo que quiere decir que soporta Programación Orientada a Objetos, Programación Funcional y Programación Imperativa.
- ¡Lo más cool es que la sintaxis es súper sencilla y es de Código Abierto! Aparte de que es el más jat...



pythonTM

Logo oficial del Lenguaje de
Programación Python



¿La Filosofía detrás de Python?

- El Creador de Python, Guido van Rossum, ha seguido siendo el principal autor del lenguaje con el título del “Benevolente Dictador Vitalicio”.
- Pero la comunidad de Python es tan grande, que un desarrollador llamado Tim Peters, declaró “La Filosofía de Python” en su libro “El Zen de Python”.

**Bello es
Mejor que
Feo**

**Complejo es
mejor que
complicado**

**Explícito es
mejor que
Implícito**

**Lo práctico
gana a lo
bueno**

**Simple es
mejor que
complejo**

**La legibilidad
cuenta**

*Estos manes no se complican la vida,
van a lo que van de una sola vez.*

¿Qué son las Variables en Python? (Cambio)



*La típica imagen que mandaría:
*inserte miembro de la familia que
todos los días manda algo en el
grupo de Whatsapp de la familia**

- Digamos que las variables son un nombre que se refieren a algo (objeto) que vive en la memoria de nuestras computadoras, puede ser números, texto, etc.
- Cada una de ellas tiene un nombre único y a este se le llama identificador.
- Miremos a las variables como “Contenedores”, ellas contienen data, es decir “Las Zapatillas de 500 dola”, que pueden luego ser cambiados con técnicas de programación.

Tipos de Datos en Python: ¿Inmutables?



- Los tipos de Datos en Python, se pueden clasificar en dos tipos:
 - **Mutable:** Su contenido (o su valor) puede cambiar en función del tiempo
 - **Inmutables:** Su contenido (o su valor) jamás cambiará en función del tiempo.

Categoría de tipo	Nombre	Descripción
Números inmutables	int	entero
	long	entero long
	float	coma flotante
	complex	complejo
	bool	booleano
Secuencias inmutables	str	cadena de caracteres
	unicode	cadena de caracteres Unicode
	tuple	tupla

Podemos resumir los tipos de datos compuestos estándar inmutables, en esta tabla



Tipos de Datos en Python: ¿Mutables?

Categoría de tipo	Nombre	Descripción
Secuencias mutables	list	lista
	range	rango mutable
Mapeos	dict	diccionario
Conjuntos mutables	set	conjunto mutable
Conjuntos inmutables	frozenset	conjunto inmutable

- Los Tipos de Datos Mutables en Python son aquellos que podemos cambiar, modificar o actualizar el contenido que se encuentra almacenado en ellos.

Podemos resumir los tipos de datos compuestos estándar mutables, en esta tabla



Operadores Ariméticos: ¿Básicos?

Operador	Función	Ejemplo en Código
+	Hace la Suma o Adición entre los operandos	<code>>>> 5 + 9 + 6 = 20</code>
-	Hace la Resta o Sustracción entre los operandos	<code>>>> 10 - 90 - 47 = -127</code>
*	Hace la Multiplicación entre los operandos	<code>>>> 80 * 454 = 399,520</code>
/	Hace la División entre los operandos	<code>>>> 200 / 10 = 20</code>
%	Encuentra el Módulo o Residuo de la división de dos operandos, sean o no enteros.	<code>>>> 200 % 12 = 8</code>
**	Hace la potencia entre los operandos	<code>>>> 9 ** 5 = 59,049</code>
//	Realiza la división con resultado de un número entero	<code>>>> 200 // 12 = 16</code>

Si quieres obtener un resultado en tipo flotante, solo basta que uno de los operandos sea de tipo flotante.

Operadores Relacionales: ¿Tinder?



Operador	Función	Ejemplo en Código
>	Devuelve True si el operador de la izquierda, es mayor que el operador de la derecha	>>> 900 > 3 → True
<	Devuelve True si el operador de la derecha es mayor que el operador de la izquierda	>>> 900 < 3 → False
==	Devuelve True si todos los operadores son iguales	>>> 900 == 900.0001 → False
>=	Devuelve True si el operador de la izquierda es mayor o igual que el operador de la derecha	>>> 420 >= 787 → False
<=	Devuelve True si el operador de la derecha es menor o igual que el operador de la izquierda	>>> 420 <= 420 → True
!=	Devuelve True si los operandos no son iguales	>>> 420 != 420 → False

Los operadores relacionales se usan para comparar datos o variables y devolverán un valor booleano (True or False) basados en su condición



Entrada/Salida en Python: ¿Parece MetroBus?

Ejemplo con la función input() para un Str

```
>>> print("¿Cuál es tu nombre bro?")
>>> nombre = input()
>>> print (nombre)
>>> Johel Heraclio
```

Ejemplo con la función input() para un int

```
>>> print("¿Cuál es tu edad bro?")
>>> edad = int(input())
>>> print (edad)
>>> 25
```

- A veces nos toca pedirle información al usuario, pero tenemos un buen set de funciones integradas en el intérprete de Python para echarnos una mano!

La función input nos permite trabajar con diferentes tipos de datos, pero tenemos que decirle que tipo de dato es



Vamos pa' la Terminal...



- En CMD (Windows), escribamos *python* para entrar al modo interactivo del lenguaje.
- Si usas Mac OS, entra a Terminal y escribe *python3*.
- Si ya tienes instalado VisualStudio Code, vamos pa' encima, ¿Tas?

Enunciado de un Típico Problema de Python

- Johel tiene n años, por ende está gordo, viejo, cansado y tiene el autoestima por el piso...
 - Para subirle (o bajarle) el autoestima a Johel vamos a pedirle la siguiente información:
 - Primer y Segundo Nombre:
 - Edad:
 - Estatura:
 - Peso:
 - Haremos un pequeño programa en Python que nos dará en Índice de Masa Corporal (IMC) de Johel y nos dirá si está por dependiendo de su edad es (Mirar la Tabla):

Edad	Mensaje
Menor de 25 años	Tas' pelaito todavía bro
Mayor de 25	Vaya comprando su hueco en el Jardín de Paz bro, F en el chat.