

Python



Primeros Pasos

Algoritmos y Programación I
Lic. Gustavo Bianchi

Origen

- Guido Van Rossum, en el año 1989, viendo la necesidad de disponer de un lenguaje con características que le permita desarrollar rápida y fácilmente, aplicaciones para el proyecto del S.O. Amoeba, en el que estaba interviniendo; y en base a su experiencia adquirida en el diseño e implementación del lenguaje ABC; logra en 3 meses el desarrollo de un primer prototipo para mostrar en el Centro para las Matemáticas y la Informática (CWI), en los Países Bajos.
- Un año después, en febrero de 1991, se lanza la primer versión, comenzando a recibir aportes de terceros para mejorarlo.
- Su nombre proviene del grupo de humoristas británicos Monty Python
- Videos en Youtube:
 - The early years of Python
 - The modern era of Python
 - Otros



Algoritmos y Programación I
Lic. Gustavo Bianchi

Características

- **Es interpretado interactivamente** (permite el ingreso de comandos mediante el uso de una terminal o consola), a diferencia de los lenguajes compilados o que usan compiladores.
- **Multiparadigma** (programación orientada a objetos, programación imperativa y programación funcional)
- **Fuertemente tipado** (una variable con un valor de un tipo determinado, no se puede usar como si fuera de otro tipo, a menos que se haga una conversión)
- **Tipado dinámico** (una variable puede almacenar valores de distinto tipo en distintos momentos)
- **Utiliza conteo de referencias** (técnica que permite contabilizar la cantidad de referencias que tiene un determinado recurso)

Algoritmos y Programación I
Lic. Gustavo Bianchi

Filosofía del diseño de Python

(Por Tim Peters)

- **Bello es mejor que feo.**
- **Explícito es mejor que implícito.**
- **Simple es mejor que complejo.**
- **Complejo es mejor que complicado.**
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- **La legibilidad cuenta.**
- **Los casos especiales no son tan especiales como para quebrantar las reglas.**
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.
- Ahora es mejor que nunca.
- Aunque *nunca* es a menudo mejor que *ya mismo*.
- **Si la implementación es difícil de explicar, es una mala idea.**
- **Si la implementación es fácil de explicar, puede que sea una buena idea.**
- Los espacios de nombres (*namespaces*) son una gran idea ¡Hagamos más de esas cosas!

Algoritmos y Programación I
Lic. Gustavo Bianchi

Compilador Vs Intérprete

Compilador		Intérprete	
Es el encargado de traducir nuestro programa fuente, usualmente a lenguaje de máquina		Es el encargado de traducir instrucciones de código a medida que se solicitan, instrucción por instrucción	
Ventajas	Desventajas	Ventajas	Desventajas
Listo para ser Ejecutado	No es multiplataforma	Es multiplataforma	Se requiere intérprete instalado
Usualmente más rápido	Poco flexible	Fácil de testear	Más lento
El código fuente es privado	Es necesario realizar pasos adicionales	Fácil de buscar y corregir errores (Debugging)	El código fuente es público

Algoritmos y Programación I
Lic. Gustavo Bianchi

Algunos elementos del lenguaje

- Operadores Aritméticos
- Operadores de Comparación
- Operadores Lógicos
- Estilos de programación, comentarios y otros
- Definición de Variables
- Ayuda y documentación

Algoritmos y Programación I
Lic. Gustavo Bianchi

Operadores Aritméticos

Operación	Operador	Ejemplos para x=5 y=4
Suma	+	<code>>>> x+y</code> 9
Resta	-	<code>>>> x-y</code> 1
Multiplicación	*	<code>>>> x*y</code> 20
División Real	/	<code>>>> x/y</code> 1.25
División Entera	//	<code>>>> x//y</code> 1
Resto de la División	%	<code>>>> x%y</code> 1
Potencia	**	<code>>>> x**y</code> 625

Algoritmos y Programación I
Lic. Gustavo Bianchi

Operadores de Comparación

Comparaciones		Ejemplos para x=5 y=4
Mayor	>	<code>>>> x > y</code> True
Menor	<	<code>>>> x < y</code> False
Igual	==	<code>>>> x == y</code> False
Distinto	!=	<code>>>> x != y</code> True
Mayor Igual	>=	<code>>>> x >= y</code> True
Menor Igual	<=	<code>>>> y <= x</code> True
Varios		<code>>>> 3 < y <= x < 10</code> True

Algoritmos y Programación I
Lic. Gustavo Bianchi

Operadores Lógicos

Conector Lógico		Ejemplos para a=True b=False
Disyunción	or	>>> a or b True
Conjunción	and	>>> a and b False
Negativa	not	>>> not a False

Algoritmos y Programación I
Lic. Gustavo Bianchi

Delimitando Bloques y respetando un estilo

- El contenido de los **bloques de código**, es **delimitado por blancos o tabulaciones** (indentación o sangrado). Se recomienda utilizar 4 espacios en blanco en lugar de tabular, y NO mezclar blancos y tabulaciones.
- Limitar los **caracteres por línea** a un **máximo de 79**.
- Para **continuar una instrucción** en otra línea porque se alcanzó el máximo, utilizar **"**" al final de la primer línea.
- Para **comentarios de varias líneas**, iniciar y finalizar con **"**" (3 apóstrofes)
- Para **comentarios de una línea** o a continuación de código, anteponer el carácter **"#"**.
- Para seguir un estilo aceptado de programación ver el:
PEP 8 -- Style Guide for Python Code
<https://www.python.org/dev/peps/pep-0008/#introduction>

*PEP: Python Enhancement Proposals

Algoritmos y Programación I
Lic. Gustavo Bianchi

Definiendo Variables

Nombre de la Variable = Valor

↑
Operador de Asignación

```
>>>x = 5
```

Python, a diferencia de otros lenguajes, no requiere que previamente declaremos el tipo de la variable.
Al darle a x el valor 5, implícitamente le asignará como tipo, **int** (entero)

```
>>>x = "Hola"
```

Dado que Python es de tipado dinámico, es factible, asignarle a posteriori, un valor de otro tipo.
Ahora, x, almacena un tipo de dato **str** (cadena de caracteres)

```
>>>x = 5.2
```

Y ahora de tipo **float** (real)

```
>>>x = True
```

Y ahora de tipo **boolean** (lógico)

Algoritmos y Programación I
Lic. Gustavo Bianchi

Más sobre Variables

```
>>>edad = 25
```

```
>>>nombre = "Juan"
```

```
>>>nombre + edad
```

ERROR

Al ser **fuertemente tipado**, no es posible realizar operaciones con variables de distinto tipo, sin efectuar la conversión que corresponda.

```
>>>nombre + str(edad)
```

CORRECTO!

↑
Conversión del tipo **int** al tipo **str**

Algoritmos y Programación I
Lic. Gustavo Bianchi

Cómo muestro el valor de una variable?

```
>>>print(edad)
```

```
25
```

```
>>>print(nombre)
```

```
Juan
```

```
>>>print(nombre,edad)
```

```
Juan 25
```

print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
 Prints the values to a stream, or to sys.stdout by default.
 Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.

Algoritmos y Programación I
 Lic. Gustavo Bianchi

Cómo solicito el ingreso de un valor para una variable?

```
>>>edad = input("Edad: ")
Edad: _
```

ATENCIÓN! El dato ingresado, siempre será almacenado como string

```
>>>edad = int(input("Edad: "))
```

En este caso, antes de asignar, se convierte a entero

input(prompt=None, /)
 Read a string from standard input. The trailing newline is stripped.

 The prompt string, if given, is printed to standard output without a trailing newline before reading input.

 If the user hits EOF (*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError.
 On *nix systems, readline is used if available.

Algoritmos y Programación I
 Lic. Gustavo Bianchi

Estructura de Control – Condicional

IF – USO SIMPLE (Sólo rama if)	
Formato	Ejemplo
if (condición): acción 1 acción 2 acción N	edad = int(input("Edad: ")) if edad >= 18: print("Ud. es mayor de edad")
IF – USO COMPUESTO (Rama if y else)	
Formato	Ejemplo
if (condición): acción/es else: acción/es	edad = int(input("Edad: ")) if edad >= 18: print("Ud. es mayor de edad") else: print("Ud. es menor de edad")

Algoritmos y Programación I
Lic. Gustavo Bianchi

Estructura de Control – Condicional

IF – USO ANIDADO (Ya sea por la rama del if ó del else, hay uno ó más if)		
Formato		Ejemplo
if (condición): acción/es elif (condición): acción/es elif (condición): acción/es	if (condición): if (condición): acción/es else: acción/es elif (condición): acción/es else: acción/es	valor = int(input("Valor: ")) if valor > 0: print("Número Positivo") elif valor < 0: print("Número Negativo") else: print("Número Neutro")

Algoritmos y Programación I
Lic. Gustavo Bianchi

Estructura de Control – Condicional

IF – USO CORTO	
Formato	variable = valor1 if (condición) else valor 2
Ejemplo	<pre>num = int(input("Valor: ")) print("Es Par") if (num&2==0) else print("Es Impar")</pre>