UNIVERSIDAD NACIONAL DE HURLINGHAM

Introducción a Python
Parte 1

Profesores

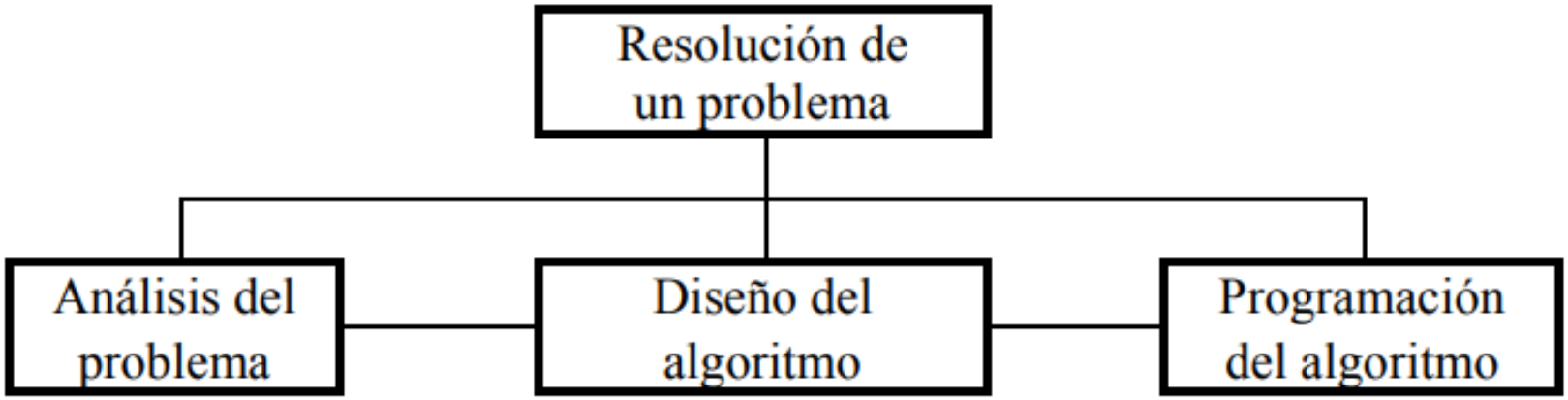
Martin Pustilnik

Sergio Gonzalez

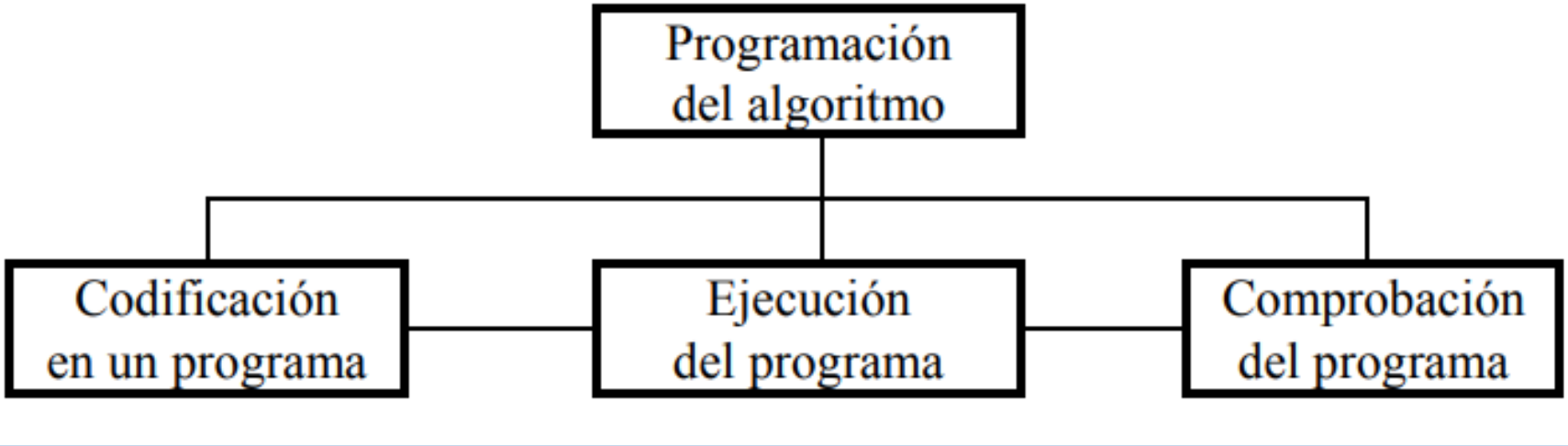
Algoritmos y programas

- Un algoritmo es un conjunto de pasos para resolver un problema.
- Programa: Forma de implementar un algoritmo en una computadora.
- Conjunto de instrucciones que interpreta el procesador.

Algoritmos y programas



Programación del algoritmo



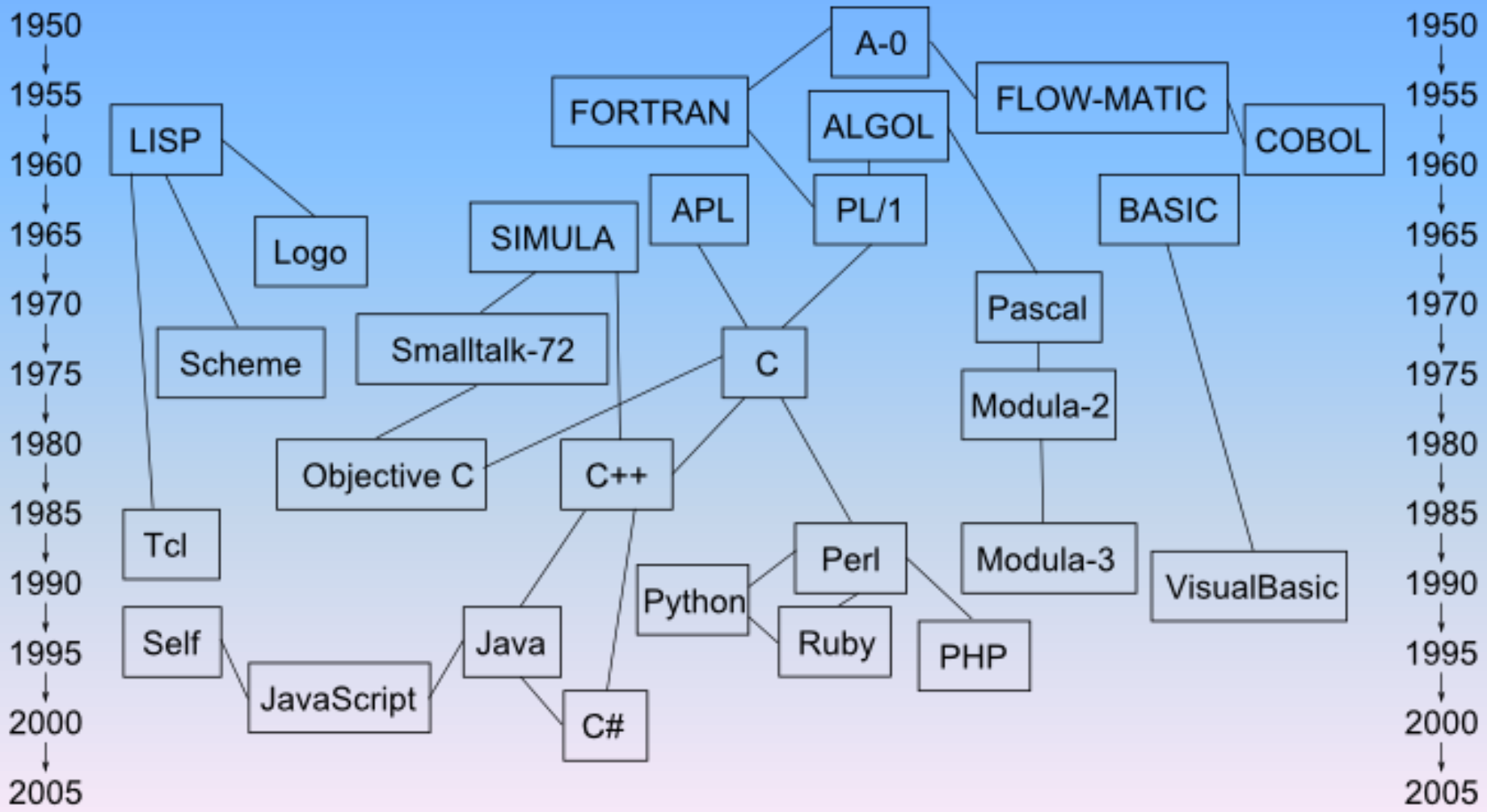
Codificación de un programa

- Lenguaje maquina.
  - 1111s y 0000s
- Lenguajes de programación de alto nivel.
  - Texto similar a lenguaje humano

Lenguajes de programación

- Independientes de la maquina
- Similar a lenguaje humano
- Una sola instrucción equivale a varias instrucciones de lenguaje maquina.
- Uso de variables, símbolos y términos usados por los humanos
- No nos preocupamos por organización real en memoria

Evolución de los Lenguajes de Programación



Elementos básicos de lenguajes de programación

- Variables
- Palabras reservadas
- Expresiones / Operadores
- Entrada / Salida
- Comentarios
- Estructura de programa

Variables

- Contenedores de datos
- Ocupa espacio en RAM
- Nombre / Identificador
  - Letras
  - Números
  - Caracteres especiales
- Tipo

Tipos de datos

- Al programar, elegimos los tipos de datos a utilizar
  - Define rango acotado (Ahorro de memoria)
  - Operaciones permitidas

Tipos de datos primitivos

- Implementados en el lenguaje
- Representación en la computadora de datos enteros, reales, lógicos, caracteres, etc.
- Interpretación de un patrón de bits
- Limite en función del tamaño
  - Cantidad de bits asignados
  - Signo
  - Desbordamiento (*Overflow*)

Tipos de datos primitivos

Tipo de variable	Bytes que ocupa	Rango de valores
boolean	2	true, false
byte	1	-128 a 127
short	2	-32.768 a 32.767
int	4	-2.147.483.648 a 2.147.483.649
long	8	$-9 \cdot 10^{18}$ a $9 \cdot 10^{18}$
double	8	$-1{,}79 \cdot 10^{308}$ a $1{,}79 \cdot 10^{308}$
float	4	$-3{,}4 \cdot 10^{38}$ a $3{,}4 \cdot 10^{38}$
char	2	Caracteres (en Unicode)

Booleanos

- Boolean
- Lógica booleana
  - Verdadero (True)
  - Falso (False)

Caracteres

- Char
- Tabla de caracteres
  - UNICODE
  - ASCII

https://unicode-table.com/es/

https://elcodigoascii.com.ar/

Enteros

- Byte
- Short
- Int
- Long

- Números con signo, rango posible depende del tamaño de la variable en memoria.

Números reales – Punto flotante

- Float
- Double

- Precisión: Cantidad de cifras decimales
- Similar a notación científica

$$5.75 \times 10^{4} = 57\,500$$
$$6.7 \times 10^{-5} = 0.000\,067$$

Cadenas de caracteres (Strings)

- Serie de caracteres
  - Palabras

Palabras reservadas

Reserved Words								
False	as	continue	else	from	in	not	return	yield
None	assert	def	except	global	is	or	try	
True	break	del	finally	if	lambda	pass	while	
and	class	elif	for	import	nonlocal	raise	with	

Expresiones / Operadores

- Operar con tipos de datos primitivos
  - Asignación
  - Aritméticos
  - Condicionales
  - Bit

Operadores de asignación

- Asignar valores en variables
  - =
  - <-
  - += / -=

Operadores Aritméticos

operador	significado
+	Suma
-	Resta
*	Producto
/	División
//	División entera
%	Módulo (resto)

- Diferente función con diferentes tipos de datos

Operadores condicionales

operador	significado
<	Menor
>	Mayor
>=	Mayor o igual
<=	Menor o igual
==	Igual
!=	Distinto
not	No lógico (NOT)
and	"Y" lógico (AND)
or	"O" lógico (OR)

Entrada / Salida

- Por pantalla / teclado
- Archivos

Estructura de programa

- Indentado
- Uso de ";"
- Sentencias entre llaves "{}"

Comentarios

- Líneas que no son ejecutadas

Estructura de un programa

- Declaración de variables
  - Tipo
  - Nombre
- Sentencias que implementan el algoritmo

```
Identificador del programa o módulo
{sección de declaraciones}
inicio
        {datos de entrada}
        {sentencias imperativas, que ejecutan el algoritmo correspondiente}
        {datos de salida}
fin
```

Traducción de un programa

- Editor de texto -> Programa fuente
- Programa traductor -> Texto a lenguaje maquina
  - Compilador - Montador (Linker)
    - Ejecutable
  - Interprete
    - Realiza traducción
    - Ejecuta directamente archivo fuente
    - No genera ejecutable

Que es PYTHON?

- Python es un lenguaje de programación interpretado con una sintaxis que favorece la legibilidad del código.

Lenguaje de programación Python

- Creado a finales de los 80s por Guido Van Rossum
- Multiparadigma: Orientado a objetos, programación funcional y programación imperativa.
- Licencia GNU GPL
- Codigo legible y transparente

Consideraciones antes de programar en Python

- En Python hay diferencia entre mayúsculas y minúsculas.
- Las línea de código no deben terminar con "; "
- Los comentarios; si son de una línea debe comenzar con "#" y si ocupan más de una línea van entre ‘ ’ ’
- El indentado es obligatorio, ya que no se usan llaves para separar bloques de codigo

Imprimir cosas por pantalla

Función "print"

```
print("texto a imprimir")

print(variable) -> Imprime contenido de la variable
```

Ingreso de datos por teclado

- Para leer los datos que se introducen en el teclado se utiliza la funcion **input()**

- Se puede utilizar de la siguiente manera:

```
variable = input("Ingrese un numero: ")
```

Ingreso de otros tipos de datos

- Por defecto, la funcion convierte la entrada a una variable de tipo texto

- Recuerden que es posible castear las variables

```
variable = int(input("Ingrese un numero entero: "))
```

Declaración de variables

- Las variables no se declaran definiendo un tipo
- El tipado es automatico y dinamico
- No se pueden realizar operaciones entre variables de distinto tipo
- Existen funciones para el casteo:
  - int()
  - float()
  - str()

Declaración de variables

```
nombreVariable = valor
```

Ejemplos:

```
dias = 2
decision = True
letra = "C"
radio = 25.63
dias = "lunes"

x , y , z = 34 , 25 , 12
x = y = "Hola"
```

Estructuras de control

- Ya sabemos declarar variables, hacer operaciones de distintos tipos entre ellas, ingresar y sacar datos del programa

- Nos falta estructurar el codigo

Estructuras de control

- Orden de realización de los pasos del algoritmo
  - Selectivas
  - Repetitivas

Estructuras de control selectivas

- Decidir que hacer a partir de evaluar una condición.
- Uso de operadores y expresiones lógicas.
- Bifurcaciones en el flujo del programa.

Alternativas simples

- Si – Entonces / If – then.
- Ejecuta una acción solo cuando se cumple una condición

```
si <condición> entonces
    <acciones>
fin_si
```

Alternativas simples

- Si – Entonces / If – then.
- En Python

```
if condicion:
    S1
    S2
    ...
    Sn
```

Alternativas dobles

- Si – Entonces – si_no / If – then – else.

```
si <condición> entonces
    <acciones S1>
si_no
    <acciones S2>
fin_si
```

```
if condicion:
    acciones S1
else:
    acciones S2
```

Alternativas múltiples

- Si – Entonces – si_no – entonces …
- If – then – else if – then …

```
si <condición1> entonces
    <acciones S1>
si_no <condición2> entonces
    <acciones S2>
si_no <condición3> entonces
    <acciones S3>
fin_si
```

```
if condicion1:
    acciones S1
elif condicion2:
    acciones S2
elif  condicion3:
    acciones S3
```

Estructuras de control repetitivas

- Conjunto de operaciones que se deben repetir varias veces.
- Ciclo o bucle: Parte de un programa que se repite (iteración) un numero dado de veces o mientras se cumpla una condición.
- Evitar bucles infinitos.

Estructuras de control repetitivas

- Tres componentes básicos:
  - Decisión
  - Cuerpo del bucle
  - Salida del bucle

Estructuras de control repetitivas

- Desde – Hasta / For

desde v=vi **hasta** vf **hacer**
  <acciones>
  .
  .
fin_desde

v: variable índice
vi, vf: valores inicial y final de la variable

```
for variable in elemento_iterable:
    <acciones>
```

Estructuras de control repetitivas

- Desde – Hasta / For

- Repeticion de bloque de codigo 10 veces:

```
for var in range(10):
    <bloque codigo>
```

Estructuras de control repetitivas

- Mientras / While

mientras condición **hacer**
    <acciones>
fin_mientras

```
while condicion:
    <accion>
```

Entornos de desarrollo Python

- Spyder
- Jupyter Notebook
- Sublime
- Atom
- Thonny
- PyCharm
- Eclipse + PyDev

Componentes de entorno de desarrollo

- Editor de texto y consola
- Facilidades en edicion
  - Coloreado de Palabras según la sintaxis
  - Indentación automática.
  - Emparejamiento de parentesis y llaves.
  - Introducción de comentarios