

Introducción a la programación con Python

Alexis Rodríguez

Marcel Morán C

Instructores

Alexis Rodríguez



Marcel Morán C.



Esquema

- Cronograma del curso
- ¿Qué es programar?
- ¿Qué es Python?
- Sintaxis y semántica
- Tipos de datos
- Variables
- Operadores
- Control de flujo

Información del curso

Cronograma

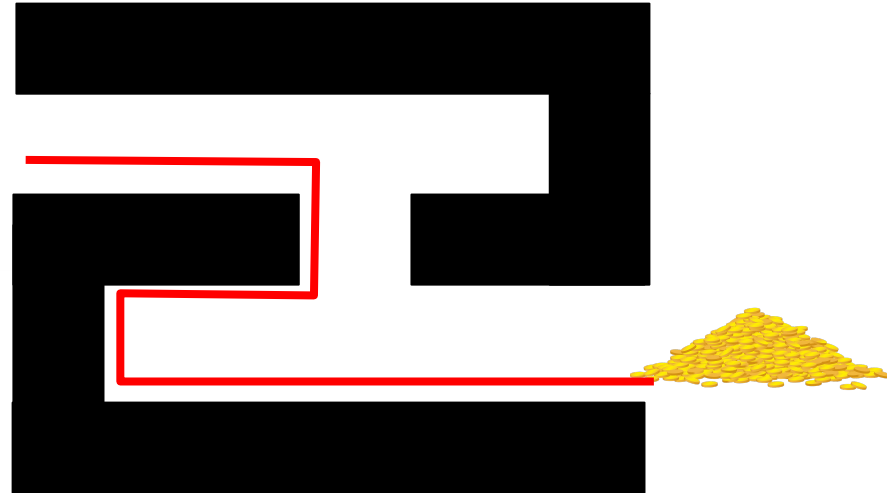
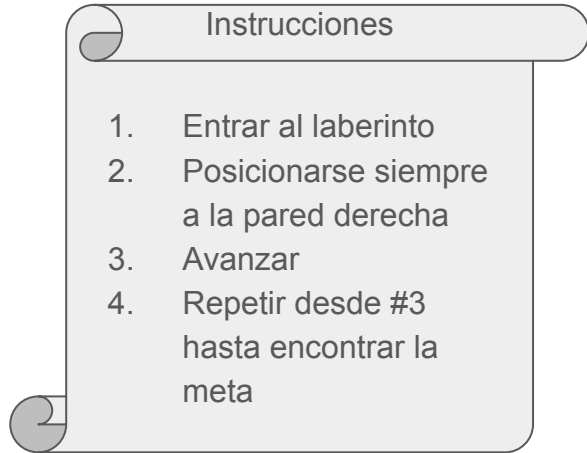
- Semana 1. Python y conceptos fundamentales
- Semana 2. Programación orientada a objetos con Python, GUI
- Semana 3. Desarrollo de juego I
- Semana 4. Desarrollo de juego II y evaluación

Evaluación

- Proyecto en las dos últimas semanas
- Grupos de máximo dos personas
- Creación de un videojuego

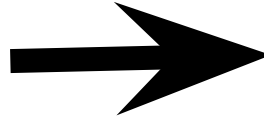
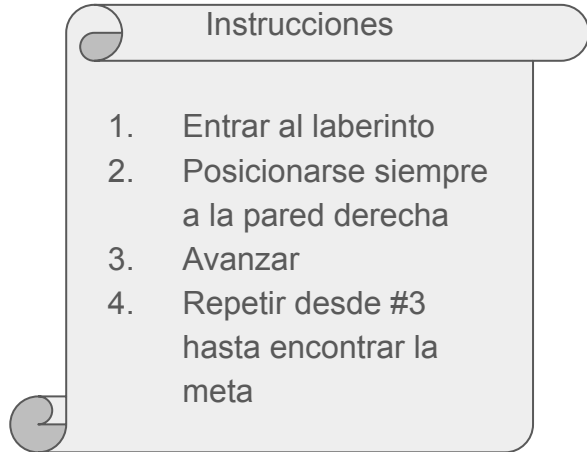
¿ Qué es programar?

- Proporcionar un conjunto o set de instrucciones a una computadora.
- Generalmente con el objetivo de resolver un problema.
- Las instrucciones son proporcionadas a través de lenguajes de programación.



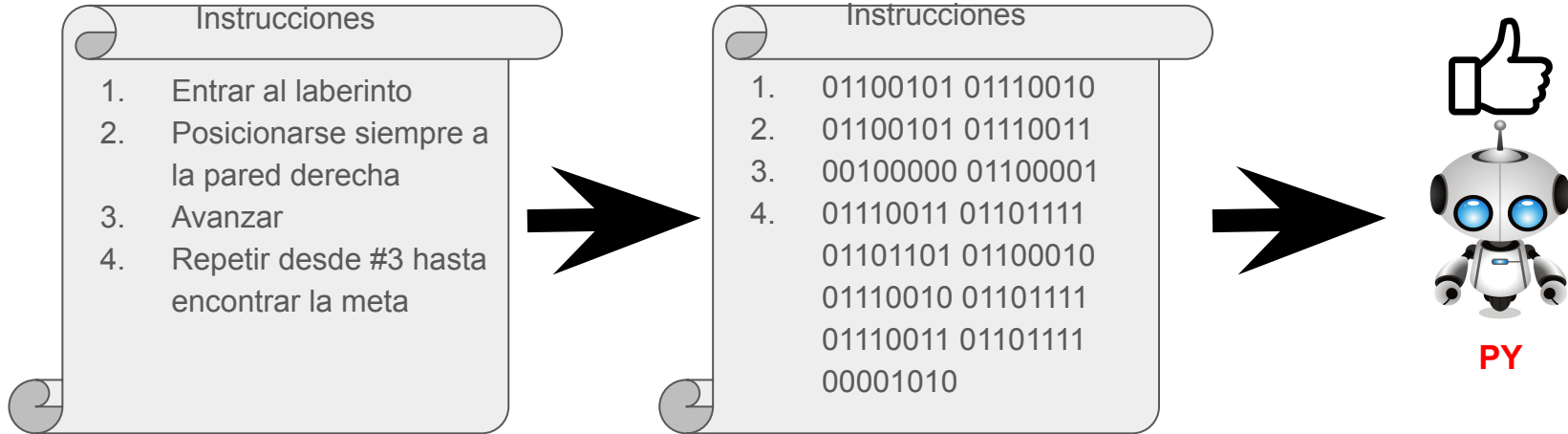
¿ Qué es programar?

- Proporcionar un conjunto o set de instrucciones a una computadora.
- Generalmente con el objetivo de resolver un problema.
- Las instrucciones son proporcionadas a través de lenguajes de programación.



¿ Qué es programar?

- Proporcionar un conjunto o set de instrucciones a una computadora.
- Generalmente con el objetivo de resolver un problema.
- Las instrucciones son proporcionadas a través de lenguajes de programación.

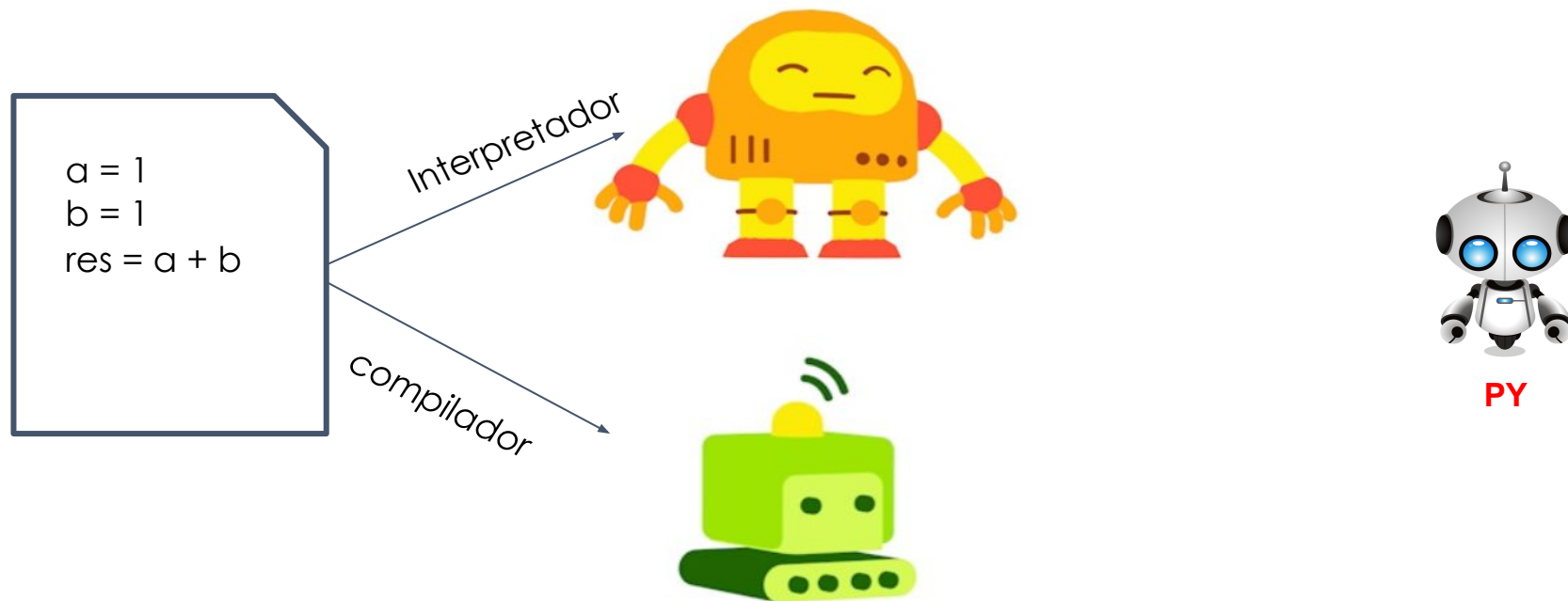


¿ Qué es Python?

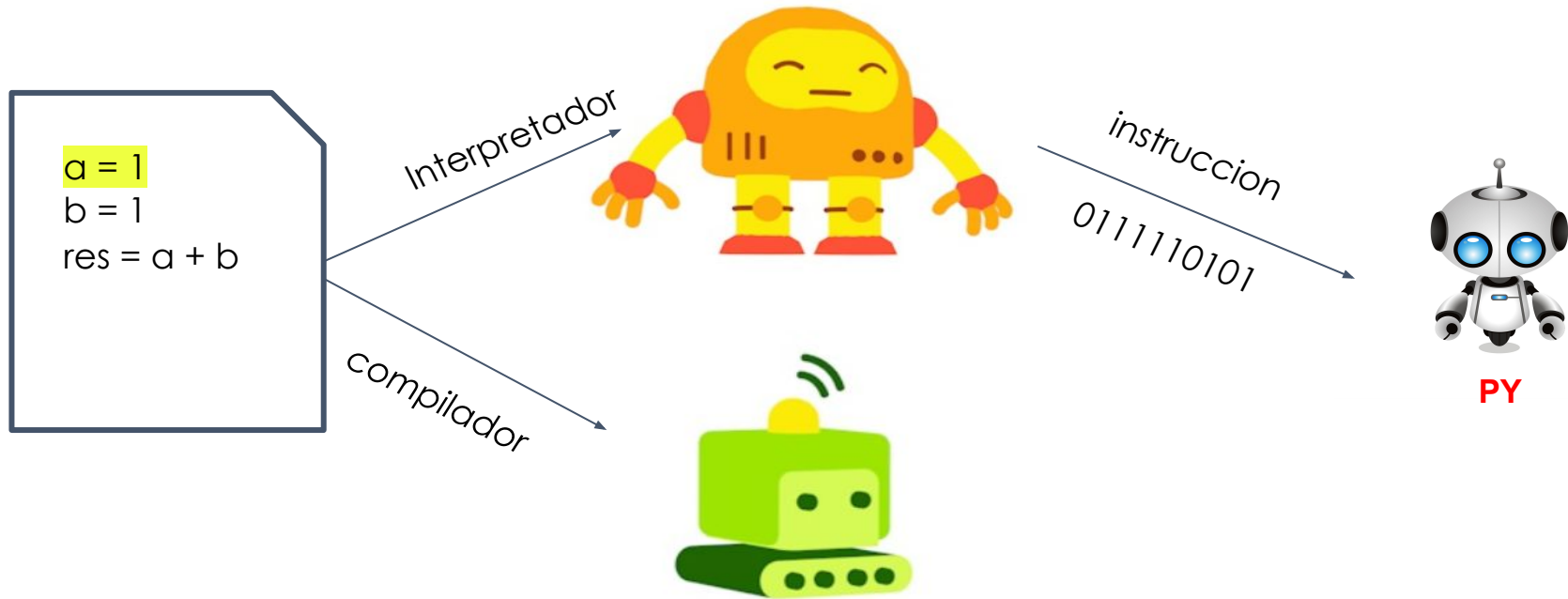
- Un lenguaje de programación de alto nivel
- Orientado a Objetos
- Desarrollo de scripts
- Se interpreta en lugar de compilarse



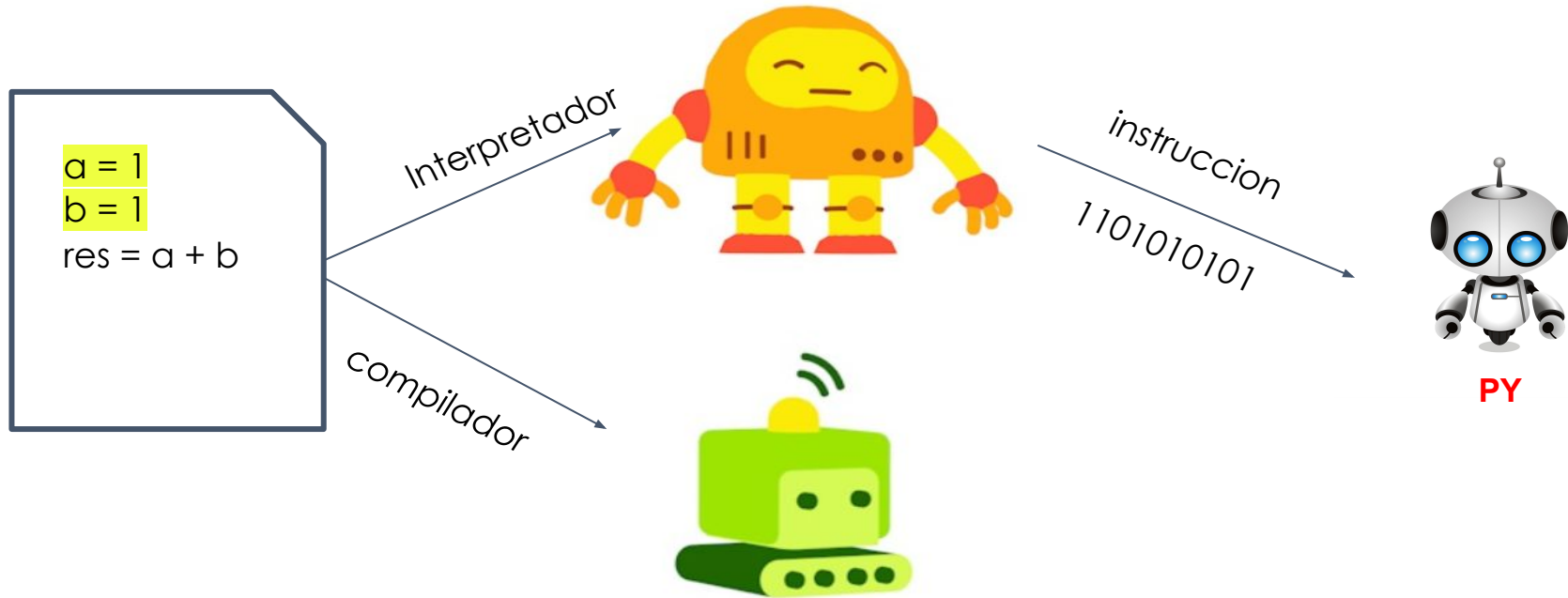
¿ Qué es Python?



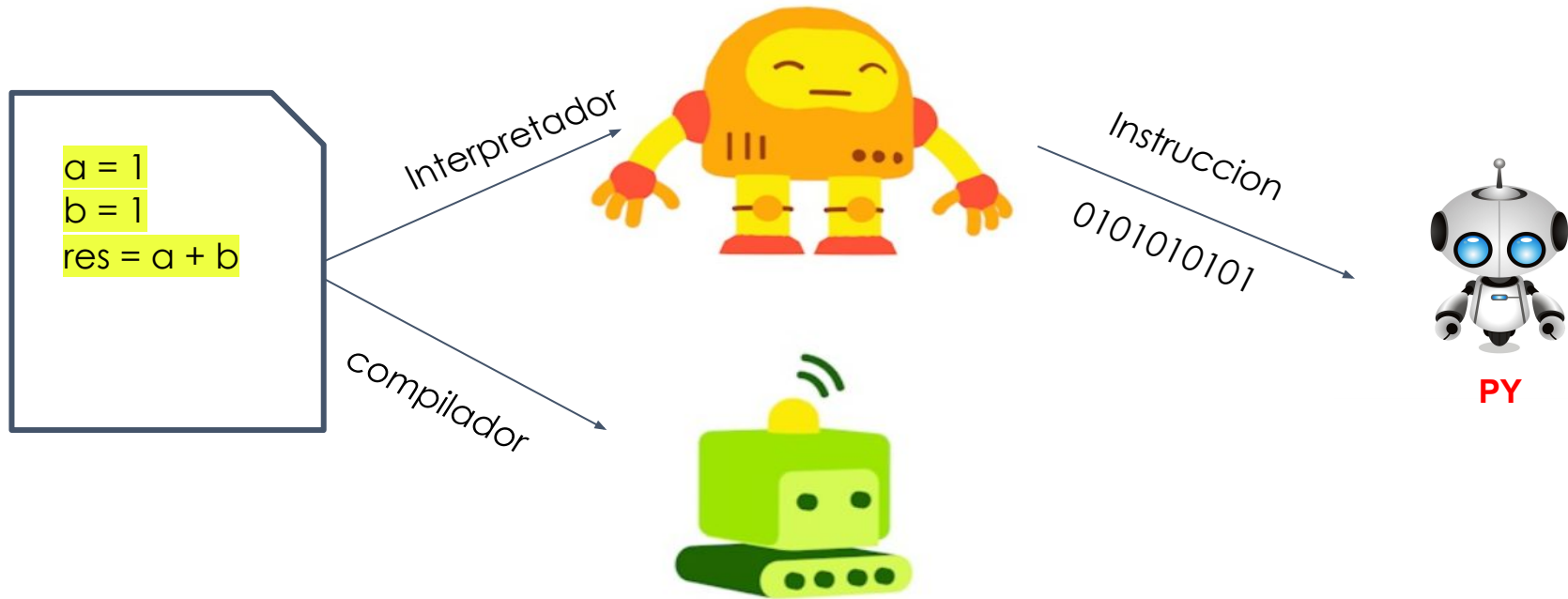
¿ Qué es Python?



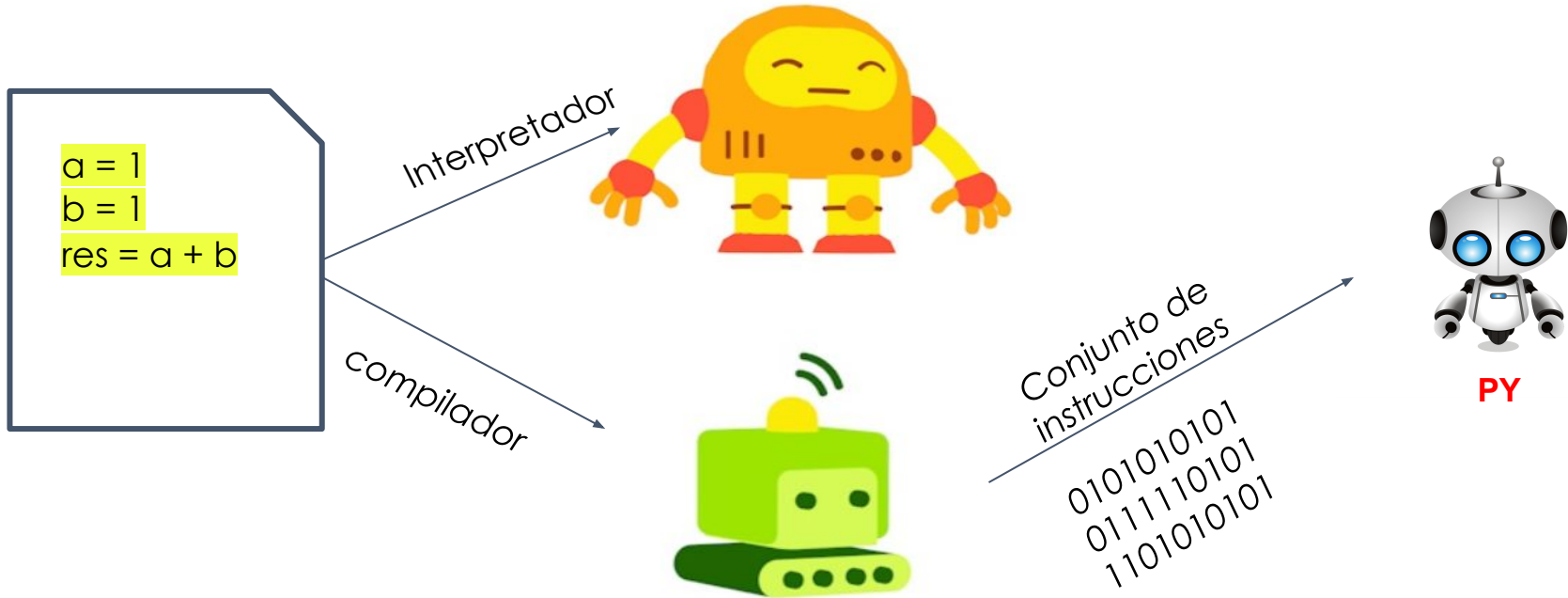
¿ Qué es Python?



¿ Qué es Python?

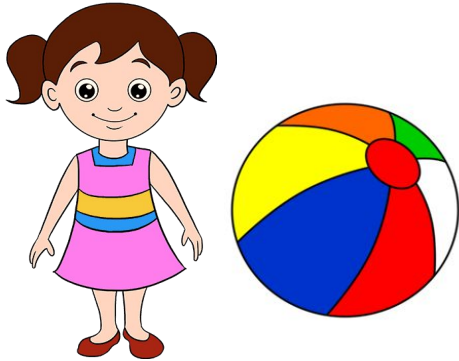


¿ Qué es Python?



Sintaxis

- La sintaxis de Python define la combinación de palabras que son consideradas correctas o válidas.
- Python espera recibir las instrucciones de una manera predeterminada



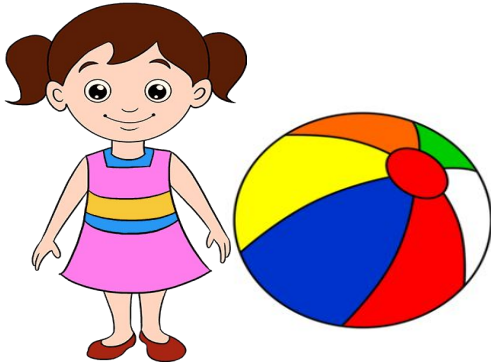
Una niña juega con la pelota.

Con Jugar niña la pelota una.

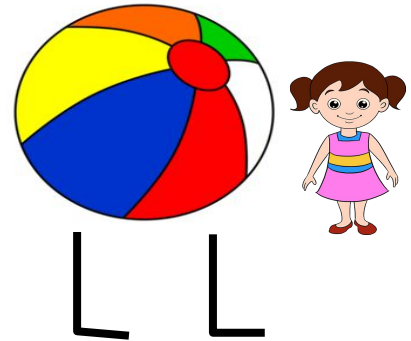
Semántica

- En el lenguaje español, la semántica trata del significado de las expresiones.
- En programación, un programa semánticamente incorrecto realiza una tarea diferente a la esperada.

Una niña juega con la pelota.



Una pelota juega con la niña.



Tipos de datos

- Strings, floats, booleans, e integers
- Datos numéricos pueden ser representados con integers: **-1, -2, 0, 1, 2**
- Representaciones decimales con floats: **-1.01, -2.001, 1.0001, 2.00001**
- Representación de valores condicionales con booleans: **True** o **False**
- Representación de palabras con strings: **“Hola”, “Mundo”, “Hola Mundo”**

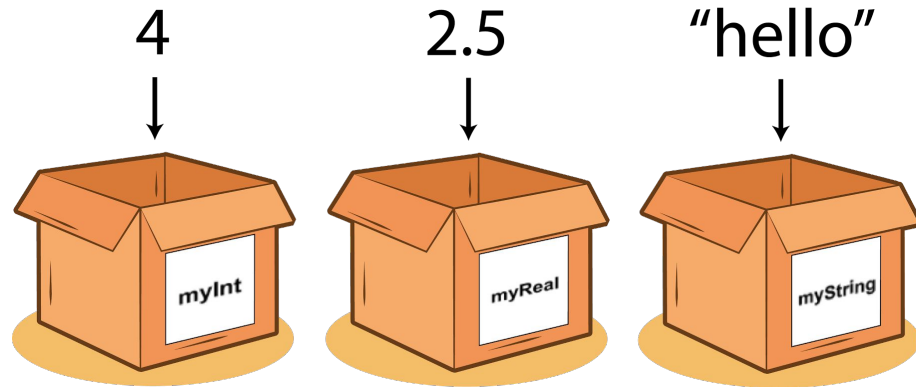
Variables

- Las variables contienen datos.
- Representación simbólica de un dato.

`un_número = 1`

`pi = 3.1415`

`un_saludo = "hola"`



Convirtiendo el tipo de variables

- Las variables pueden convertirse de un tipo a otro
- Sintaxis: float(), str(), int()
- **3** -> **3.0**
- **3.0** -> **"3.0"**
- **"3.0"** -> **3**

Operadores

- Operadores aritméticos

OPERADOR	DESCRIPCIÓN	USO
+	Realiza Adición entre los operandos	$12 + 3 = 15$
-	Realiza Substracción entre los operandos	$12 - 3 = 9$
*	Realiza Multiplicación entre los operandos	$12 * 3 = 36$
/	Realiza División entre los operandos	$12 / 3 = 4$
%	Realiza un módulo entre los operandos	$16 \% 3 = 1$
**	Realiza la potencia de los operandos	$12 ** 3 = 1728$
//	Realiza la división con resultado de número entero	$18 // 5 = 3$

- Operadores lógicos

OPERATOR	DESCRIPTION	USAGE
and	Returns True if both the operands are True	a and b
or	Returns True if any one of the operands are True	a or b
not	Returns True if the operand is False	not a

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Operadores

- Operadores relacionales

SÍMBOLO	DESCRIPCIÓN	EJEMPLO	BOOLEANO
==	IGUAL QUE	5 == 7	FALSE
!=	DISTINTO QUE	ROJO != VERDE	TRUE
<	MENOR QUE	8 < 12	TRUE
>	MAYOR QUE	12 > 7	TRUE
<=	MENOR O IGUAL QUE	16 <= 17	TRUE
>=	MAYOR O IGUAL QUE	67 >= 72	FALSE

- Operadores de pertenencia

Identificar un elemento dentro de una secuencia.

in / not in

```
correo_recibido = "Pablo se encuentra a 10km de distancia  
y quiere conocerte."
```

```
es_spam = "quiere conocerte" in correo_recibido
```

Operadores

- Operadores bit a bit

&	Bitwise AND	$x \& y$
	Bitwise OR	$x y$
~	Bitwise NOT	$\sim x$
^	Bitwise XOR	$x \wedge y$
>>	Bitwise right shift	$x >>$
<<	Bitwise left shift	$x <<$

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Truth Table

A	B
0	1
1	0

```
a = 0b10110110 #182
b = 0b10001100 #140
```

```
a_and_b = a & b
print("a & b:", "{0:b}".format(a_and_b))
```

Output: a & b: 10000100

```
a_or_b = a | b
print("a | b:", "{0:b}".format(a_or_b))
```

Output: a | b: 10111110

```
a_desplazada_derecha = a >> 1
print("a >> 1:", "{0:b}".format(a_desplazada_derecha))
```

Output: a >> 1: 01011011

Flujo del programa

- Hasta ahora no hemos podido especificar qué instrucciones se ejecuten
- Podemos controlar la ejecución de un programa con cláusulas condicionales
- Se requiere de una **condición** y una **cláusula** o bloque de código a ejecutar
- Condiciones siempre se evalúan a True o False
- Cláusulas pueden contener cláusulas dentro de sí mismo
- Las cláusulas necesitan estar **indentadas**

```
if condición:  
    _Cláusula  
elif condición:  
    _Cláusula  
else:  
    _Clausula
```

Flujo del programa

```
edad_mama = 45
edad_papa = 50

if edad_mama == edad_papa:
    print('Mamá y papá tienen la misma edad')
elif edad_mama > edad_papa:
    print('Mamá es mayor que papá')
else:
    print('Papá es mayor que mamá')
```

Conclusión

- Programar es escribir un conjunto de instrucciones con el fin de que se ejecuten en una computadora.
- La comunicación entre el programador y la computadora se hace a través de lenguajes de programación.
- Python es un lenguaje interpretado.
- Python contiene 4 principales tipos de datos: strings, floats, booleans e integers.
- Las variables son representaciones simbólicas usadas para hacer referencia a algún tipo de dato.
- Usando operadores podemos realizar operaciones aritméticas, lógicas e incluso comparar variables.
- El flujo del programa puede ser controlado usando cláusulas condicionales.