



Clase 1 – Variables y expresiones

IIC1103 Sección 9 – 2019-2

Profesor: Felipe López

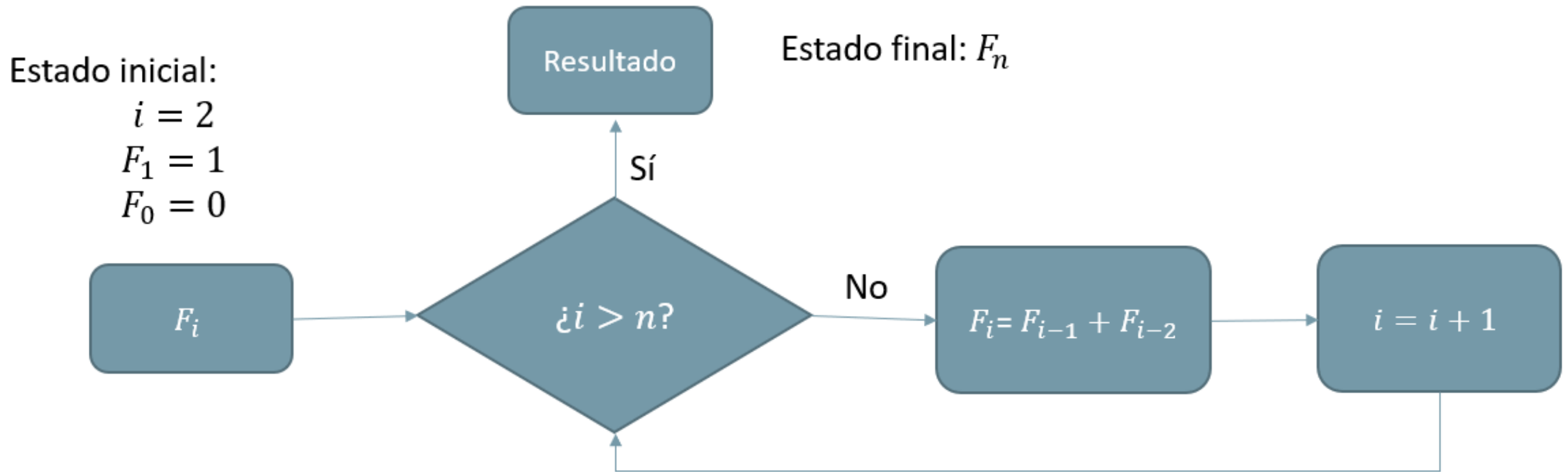
Fecha: 20 de agosto del 2018

20-08-2018

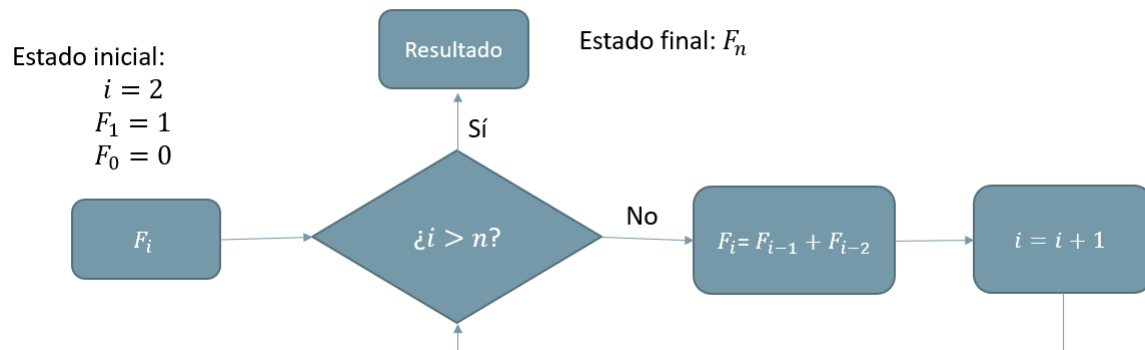
Resumen clase anterior

- **Algoritmo**
 - “Secuencia ordenada de pasos cuyo objetivo es resolver un problema (o generar un output) desde un estado inicial (o input)”
- **Diagrama de flujo:** Representación gráfica de un algoritmo
 - Pueden tener condicionales y/o ciclos.
 - Recordar: Que SIEMPRE tengan una “salida”.
 - Existen algoritmos “mejores” que otros.

Resumen clase anterior



Resumen clase anterior



Las variables son muy importantes, ya que “almacenan” y representan información.
¡Son la base de Python!

Con $i = 2, F_1 = 1, F_0 = 0, n = 6$

Mientras i menor o igual que n , entonces:

F_i es igual a F_{i-1} más F_{i-2} e

i es igual a i más 1

Cuando i sea mayor a n , entonces el n -ésimo término de la sucesión de Fibonacci es F_n

		Iteración				
		2	3	4	5	6
Variables	i	2	3	4	5	6
	F_i	1	2	3	5	8

Contenidos

1. Operadores matemáticos
2. Variables
3. Tipos de datos
4. Ejemplos

Tarea Clase 0

- Bajar e instalar Python
- Abrir IDLE
- Escribir la siguiente línea de código

```
print("Hola mundo")
```

- Ejecutarla
- <https://repl.it/@FelipeLopez/IIC1103Clase1>

Secuencias de comandos en Python

¿Qué es lo que acabamos de hacer?

Entregarle una secuencia de comandos a Python, que pudo interpretar y ejecutar

¿Qué otras cosas se pueden hacer?

Muchas... pero partamos por lo más básico

Operadores matemáticos

- Partiremos definiendo los operadores matemáticos más básicos. Estos son:
- $+$: Suma los valores de la izquierda y la derecha

```
#ejemplo operador suma  
print(4+7)  
>>11  
print(1001+817)  
>>1818
```

- $-$: Resta el valor de la derecha al valor de la izquierda

```
#ejemplo operador resta  
print(4-7)  
>>-3
```


Operadores matemáticos

- *: Multiplica el valor de la izquierda por el de la derecha

```
#ejemplo operador multiplicación  
print(5*8)  
>>40  
print(25*0)  
>>0
```

- /: Divide el valor de la derecha por el de la izquierda.

```
#ejemplo operador división  
print(12/4)  
>>3.0  
print(4/12)  
>>0.3333333333333333
```

Operadores matemáticos avanzados

Existen operadores un poco más complejos que los anteriores.

- **: Calcula la potencia. El valor de la izquierda es la base y el de la derecha es el exponente.

```
#ejemplo operador potencia  
print(2**5)  
>>32
```

- //: Divide el valor de la izquierda por el de la derecha, y entrega el valor entero del resultado.

```
#ejemplo operador división entera  
print(5//4)  
>>1
```

En esta operación, el resultado de la división es 1.25
El resultado y lo que se imprime en la consola es 1 porque el operador retorna la parte entera del número.

Operadores matemáticos avanzados

- %: Divide el valor de la izquierda por el de la derecha. No entrega el resultado sino que el resto de esta división.

```
#ejemplo operador módulo  
print(7%5)  
>>2
```

Se puede observar que el 5 cabe una vez en el 7 y sobran 2 unidades, que es el resto de la operación. Por lo tanto, este es el resultado de la operación con el operador módulo y lo que se imprime en la consola.

¿Cómo podríamos saber si n es un número es par?

Si $n\%2$ es igual a 0, entonces n es par.

Variables

Los operadores que acabamos de definir son muy útiles. Pero no permiten operaciones muy complejas.

¿Y si quisiéramos guardar el resultado de una operación para utilizarlo nuevamente?

Para esto introducimos el concepto de **variables**.

Variables

Las variables son objetos donde podemos almacenar información.

Variables

Las variables son objetos donde podemos almacenar información.

Para crear una variable, simplemente hay que escribirla en nuestro código.

Variables

Las variables son objetos donde podemos almacenar información.

Para crear una variable, simplemente hay que escribirla en nuestro código. A esto le llamamos **declarar una variable**.

Al crear una variable, podemos asignarle inmediatamente un valor. Esto es una **declaración explícita**.

```
#ejemplo declaración de una variable  
a=1  
print(a)  
>>1
```

Variables

También podemos crear una variable asignándoles el resultado de alguna operación

```
#ejemplo declaración de una variable  
como resultado de una operación
```

```
a=20*860  
print(a)  
>>17200
```

```
b=26%2  
print(b)  
>>0
```


Variables

Es importante notar que las variables pueden tener cualquier nombre

```
#ejemplo nombres de variables
cantidad_de_personas=1
print(cantidad_de_personas)
>>1

pasos=9-3
print(pasos)
>>6

esteEsElNombre_de_una_variable=7
print(esteEsElNombre_de_una_variable)
>>7
```

Variables

También se puede operar con variables. Digamos que tienes 30 manzanas y 6 niños. Se quiere saber cuántas manzanas puede comer cada niño.

```
#ejemplo operación con variables  
manzanas=30  
niños=6  
cantidad_de_manzanas_por_niño=manzanas/niños  
print(cantidad_de_manzanas_por_niño)  
>>5.0
```

Variables

En el ejemplo anterior, el resultado de la división lo asignamos a la variable `cantidad_de_manzanas_por_niño`.

Sin embargo, y a diferencia de las variables anteriores, el resultado de la división (aunque fuera un número entero), apareció como 5.0

Esto es porque este tipo de dato es **decimal**, a diferenciar de los ejemplos anteriores donde el tipo de dato es **entero**.

break

5 min.

Tipos de datos

Python tiene varios tipos de datos, por ahora, solo debes manejar estos:

- `int`: Variables que contienen números enteros.

```
#ejemplo variable de tipo int  
variable_a=1  
print(variable_a)  
>>1
```

- `float`: Variables que contienen números decimales

```
#ejemplo variable de tipo float  
variable_b=3.9  
print(variable_b)  
>>3.9
```

Tipos de datos

Python tiene varios tipos de datos, por ahora, solo debes manejar estos:

- `string`: Variables que contienen texto.

```
#ejemplo variable de tipo string  
variable_c="este es un texto de ejemplo"  
print(variable_c)  
>>este es un texto de ejemplo
```

Conversión de datos

Saber el tipo de datos de una variable sirve para poder transformar una variable en otra. Por ejemplo, ¿Qué pasa si queremos transformar un texto a número?

- Para transformar de texto a entero: Hay que ocupar el comando `int(...)`

```
#transformar texto a número entero
texto1="710"
print(texto1)
>>710

variable_tipo_entera=int(texto1)
print(variable_tipo_entera+1)
>>711
```

```
a="2"
print(a+1)
```

```
Traceback (most recent call last):
  File "python", line 80, in <module>
TypeError: must be str, not int
```

Conversión de datos

Saber el tipo de datos de una variable sirve para poder transformar una variable en otra. Por ejemplo, ¿Qué pasa si queremos transformar un texto a número?

- Para transformar de texto a decimal: Hay que ocupar el comando `float(...)`

```
#transformar texto a número decimal
texto2="45.6891"
print(texto2)
>>45.6891

variable_tipo_float=float(texto2)
print(variable_tipo_float+1)
>>46.6891
```

```
a="2.67"
print(a+1)

Traceback (most recent call last):
  File "python", line 80, in <module>
TypeError: must be str, not int
```


print(...)

- A pesar que ya hemos visto el uso del comando print(...) en ejemplos anteriores, es bueno destacar algunos detalles:

```
#ejemplo print
texto1="ejemplo1"
texto2="ejemplo2"
variable_numerica1=50
variable_numerica2=40.56

print("Como ya hemos visto, se puede imprimir solo texto. Pero también se pueden concatenar valores de variables con el operador ,")
>>Como ya hemos visto, se puede imprimir solo texto. Pero también se pueden concatenar valores de variables con el operador ,
print("Por ejemplo:",texto1)
>>Por ejemplo: ejemplo1
print("O también:",variable_numerica1)
>>O también: 50
print("También se pueden concatenar varios valores juntos, como estoy haciendo ahora",texto1,texto2,"aunque sean de distinto tipo",variable_numerica1,variable_numerica2)
>>También se pueden concatenar varios valores juntos, como estoy haciendo ahora ejemplo1 ejemplo2 aunque sean de distinto tipo 50 40.56
```

Input del usuario

- Es muy importante que el usuario pueda ingresar información.
- Para hacerlo, se ocupa el comando `input(...)`
- Al ocupar este comando, la ejecución del código se detiene y se espera que el usuario ingrese algún tipo de información y luego presione la tecla "Enter".
- Dentro del paréntesis se puede ingresar un texto que se imprimirá en pantalla.
- Python asume que cualquier información ingresada por el usuario es de tipo texto.

Input del usuario

- Ejemplo:

```
#ejemplo input
texto_ingresado_por_el_usuario=input("ingresa un texto")
<<texto_escrito_por_el_usuario

print("el texto ingresado es:")
>>el texto ingresado es:

print(texto_ingresado_por_el_usuario)
>>texto_escrito_por_el_usuario
```

- Notemos que la información ingresada por el usuario se guardó en la variable `texto_ingresado_por_el_usuario`. Si no hubiésemos ocupado esta variable entonces el `input` del usuario no se guardaría, y por lo tanto no podría ser usado posteriormente.

Ejemplo 1: Cambiar dólares a pesos

Queremos escribir un programa que haga un cambio simple de dólares a pesos. Para eso, le pediremos al usuario que ingrese cuántos dólares tiene y cuál es el precio del dólar del día de hoy.

```
#ejemplo 1: Cambiar dólares a pesos
```

```
precio_dolar=input("Ingrese el precio del dólar hoy:")
```

```
cantidad_dolares=input("Ingrese la cantidad de dólares que desea cambiar")
```

```
cantidad_pesos=precio_dolar*cantidad_dolares
```

```
print("Tus",cantidad_dolares," a un precio de",precio_dolar,"equivalen a",cantidad_pesos,"pesos")
```

```
Ingrese el precio del dólar hoy: 644
```

```
Ingrese la cantidad de dólares que desea cambiar: 1000
```

```
Traceback (most recent call last):
```

```
  File "python", line 5, in <module>
```

```
TypeError: can't multiply sequence by non-int of type 'str'
```

Ejemplo 1: Cambiar dólares a pesos

El problema está en que estoy intentando multiplicar dos variables que son de tipo texto (recordemos que el input del usuario siempre es de tipo texto). Por lo tanto, debo convertir el input del usuario en float.

```
#ejemplo 1: Cambiar dólares a pesos
```

```
precio_dolar=float(input("Ingrese el precio del dólar hoy:"))
```

```
cantidad_dolares=float(input("Ingrese la cantidad de dólares que desea cambiar:"))
```

```
cantidad_pesos=precio_dolar*cantidad_dolares
```

```
print("Tus",cantidad_dolares," a un precio de",precio_dolar,"equivalen a",cantidad_pesos,"pesos")
```

```
Ingrese el precio del dólar hoy: 645.78
```

```
Ingrese la cantidad de dólares que desea cambiar: 1000
```

```
Tus 1000.0 a un precio de 645.78 equivalen a 645780.0 pesos
```

Ejemplo 2: Transformar de grados Celsius a Fahrenheit

Queremos escribir un programa que haga un cambio de grados Celsius a Fahrenheit. Recordemos que la fórmula es así:

$$^{\circ}\text{F} = ^{\circ}\text{C} \cdot \frac{9}{5} + 32$$

Ejemplo 2: Transformar de grados Celsius a Fahrenheit

Queremos escribir un programa que haga un cambio de grados Celsius a Fahrenheit. Recordemos que la fórmula es así:

```
#ejemplo 2: Cambiar de grados Celsius a Fahrenheit  
grados_celsius=float(input("Ingrese la temperatura en grados Celsius que desea convertir a grados Fahrenheit."))  
grados_fahrenheit=grados_celsius*(9/5)+32  
print("Una temperatura de",grados_celsius,"°C es equivalente a",grados_fahrenheit,"°F")
```

```
Ingrese la temperatura en grados Celsius que desea convertir a grados Fahrenheit. 23  
Una temperatura de 23.0 °C es equivalente a 73.4 °F
```

Ejemplo 2: Transformar de grados Celsius a Fahrenheit

Ahora escribe un programa que transforme de grados Fahrenheit a Celsius.

Links

- <https://repl.it/@FelipeLopez/IIC1103Clase1> que contiene todos los ejemplos cortos de la clase

Bibliografía

- <http://runest.ing.puc.cl/basics.html>
- A. B. Downey. Think Python: How to think like a computer scientist. Green Tea Press, 2013 -> Capítulo 2



Clase 2 - Variables

IIC1103 Sección 3 – 2018-2

Profesor: Felipe López

Fecha: 14 de agosto del 2018

20-08-2018