

Introducción a la Programación

Práctica 1 – Introducción

Versión del 26 de febrero de 2020

Se recuerda que tanto esta práctica como las siguientes, y los demás recursos de la materia, se encuentran disponibles para descargar desde moodle <http://virtual.grado.ungs.edu.ar/moodle/> en la materia Introducción a la Programación en la comisión correspondiente.

La matemática y la computación

En este capítulo vamos a trabajar con dos mundos que están muy relacionados: el de la *matemática* y el de la *computación*, empezando con el primero que nos es mucho más conocido.

Números, fórmulas y resultados

La base de la matemática son los números. Si uno escribe

$$0$$

cualquiera puede entender que nos referimos al *cero*. Lo mismo pasa con 1, 20 ó 45. Pero, ¿qué pasa si escribimos $1 + 1$? Simple, todos sabemos que *uno mas uno es dos*. Lo interesante acá es que hemos logrado *expresar* al *dos* sin escribir 2 en ninguna parte. Lo hicimos a partir de una fórmula, calculando cuanto daba la suma de 1 más 1. Decimos que el *más* es un *operador*, el de la suma, pero hay otros: el operador resta (-), el de multiplicación(*) y el de división(/) son ejemplos clásicos. Veamos otra fórmula:

$$2 + 3 * 4$$

Esta vez, tenemos que recordar que la multiplicación tiene *prioridad* sobre la suma, por lo que se resuelve primero. Es decir, lo anterior es equivalente a

$$2 + (3 * 4)$$

lo cual a su vez equivale a $2 + 12$, ya que $3 * 4$ es 12. Luego, el resultado final es 14. Una vez más, hemos expresado a un número –el *catorce*– sin escribirlo explícitamente, sin poner 14 en ningún lado. Y hemos podido llegar a él *reduciendo* la fórmula original, achicándola de a un paso a la vez.

Además, en matemática contamos con *variables*, que sirven para denotar números que no se sabe en principio cuánto valen. Por ejemplo, si vamos a comer entre 4 amigos y llega la hora de pagar la cuenta, sabemos que a cada uno le tocará pagar

$$x/4$$

donde x es el precio total de la comida, no hace falta saber a priori cuanto será el total para saber la cuenta que hay que hacer, para cualquier comida siempre dividimos el total por 4. Ahora sí, si mañana vamos y gastamos 40, haremos $40/4$ y cada uno pagará 10, o si gastamos 80 haremos $80/4$ y cada uno pagará 20. Lo útil es que la fórmula aplica para cualquier precio (para todo lo demás tenemos MasterCard).

Ahora, ¿qué pasa si tampoco sabemos cuantos comensales seremos? Podemos tomar la ecuación de arriba y cambiar el 4 por una variable:

$$x/y$$

donde x sigue siendo el precio total e y es ahora la cantidad de comensales. Si sabemos que seremos cuatro, como en el ejemplo original, nos queda $x/4$, como era de esperarse.

Expresiones y reducciones

Pasamos ahora al aspecto técnico, así que ¡presten mucha atención! Diremos que una expresión es una combinación de números y variables mediante operaciones. Ejemplos de expresiones son:

$$0 \quad 2 + 2 \quad x + 1 \quad (x - x) + 1 \quad \frac{\sqrt{x+y} - 4}{2xy} \quad \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Las expresiones suelen poder reducirse a otras equivalentes pero más sencillas. Cuando no pueden reducirse más decimos que son irreducibles. En el ejemplo anterior, la expresión 0 representa al *cero*, y no puede reducirse más. $2 + 2$ en cambio, se reduce a 4 y ahí ya no puede reducirse más. Decimos entonces que $2 + 2$ evalúa a 4. $x + 1$ en cambio no puede reducirse a un número ya que a priori no sabemos cuanto vale x . $(x - x) + 1$ por otro lado, puede reducirse a $0 + 1$, luego $0 + 1$ se reduce a 1. Entonces podemos decir que $(x - x) + 1$ evalúa a 1.

Existen expresiones que no evalúan a nada, como por ejemplo $20/0$, ya que no se puede dividir por cero. También debemos cuidarnos de las cosas que parecen expresiones pero no lo son, por estar mal escritas, como por ejemplo:

$$2 + 3 + \quad 2 + * 5 \quad x + 5/$$

Programas

Un programa no es muy distinto de una serie de expresiones matemáticas escritas una atrás de la otra, con números, variables y operaciones. Al correr el programa, las líneas se han de ejecutar en orden, una a la vez, como si las fuéramos poniendo en una calculadora. Dicho de otra manera, un programa es una lista de operaciones que deben seguirse en orden para poder realizar un cálculo (de manera similar a lo que una receta de cocina es a un postre). Veamos un ejemplo bien simple:

```
x = 5
print(x)
```

Nos preguntamos ¿qué hace este programa al ejecutarse? Lo primero que vemos es que el programa tiene dos líneas, lo primero que se calculará es la primera línea, `x = 5` y luego la segunda `print(x)`.

Una de las diferencias más notorias entre la matemática y un programa es que en el caso de los programas, las variables pueden tener un valor en cierto momento y otro distinto luego, algo que no pasa en matemática.

En este caso, la primera línea se lee así: a la variable x le asignamos el valor 5. Ojo, no significa que x y 5 son lo mismo, ni que son iguales. Lo que significa es que la variable x va a guardar el valor 5 a partir de esa línea hasta que otra línea le cambie el valor. Después, se pasa a ejecutar a la segunda línea `print(x)`, que se lee así: mostrar el valor de x en la pantalla. Como a x se le ha asignado el valor 5, en la pantalla se mostrará un 5. Después de eso no hay más líneas de código así que el programa termina. Nótese que x es tan solo el nombre que tiene la variable dentro del programa, y el usuario eso no lo ve.

Cadenas

Además de los números, nos será útil trabajar con texto. Para eso vamos a utilizar las cadenas. Una cadena es simplemente una tira de símbolos (como los que vemos en el teclado de una computadora y algunos más). Las cadenas se escriben entre comillas, veamos algunos ejemplos:

```
"Soy una cadena"
"Veinticinco"
"25"
```

Es importante notar que la computadora distingue entre el número 25 (escrito sin comillas) y la cadena "25". Si utilizamos la expresión sin comillas, nos estaremos refiriendo al número, y si utilizamos las comillas, nos estaremos refiriendo a la cadena de longitud 2, compuesta por los símbolos 2 y 5. A la hora de hacer cuentas, podremos hacerlas sólo cuando trabajemos con números, y nunca con cadenas.

Veamos cómo escribir un programa que le pide al usuario una cadena y la muestra por pantalla:

```
x = input("Ingrese su nombre")

print("El nombre ingresado es")
print(x)
```

En este caso, la primera línea del programa indica que al ejecutarse,

```
x = input("Ingrese su nombre")
```

se asignará a la variable `x` la cadena que ingrese el usuario, y a su vez la cadena **“Ingrese su nombre”** es el texto que se le mostrará al usuario al momento de ejecución, para que sepa qué dato se le está pidiendo.

Luego tenemos al mismo `print(x)` que teníamos antes, pero precedido por un `print("El nombre ingresado es")`, que sirve para que en la pantalla se lea al terminar “El nombre ingresado es” y luego el valor de la variable `x`, que fue asignado con la cadena que ingresó el usuario.

Inicialmente, vamos a esquematizar a los programas dividiéndolos de la siguiente manera:

1. Entrada de datos
2. Cálculo y procesamiento
3. Salida de resultados

Por ahora, vamos a hacer programas que sirvan para resolver cuentas, como una calculadora. Más tarde, a lo largo de las prácticas, iremos introduciendo de a poco otras aplicaciones de los programas.

Programas en Python

A la hora de escribir programas en una hoja de papel, uno puede tomarse ciertas licencias porque en esos casos no hace falta que el programa sea entendido por una computadora. Pero cuando quiere que así sea, es necesario escribir los programas en un lenguaje de programación real. En nuestro caso, el lenguaje es Python. El lenguaje Python es relativamente sencillo de entender comparado con otros lenguajes de programación como C o Java, sin embargo, como todo lenguaje de programación, deben respetar estrictamente sus reglas, o de lo contrario la computadora no será capaz de entenderlo.

Notas preliminares

- Los ejercicios marcados con el símbolo ★ constituyen un subconjunto mínimo de ejercitación. Sin embargo, aconsejamos fuertemente hacer todos los ejercicios.

Ejercicio 1

Escribir en papel, separando con una flecha (\rightarrow), cada paso de las reducciones de las siguientes expresiones matemáticas hasta hallar el valor final al cual se reducen. (Ejemplo: $2+3*5 \rightarrow 2+(3*5) \rightarrow 2+15 \rightarrow 17$).

- | | |
|-------------------|---|
| 1. $20 + 12 * 13$ | $20 + 12 * 13 \rightarrow 20 + (12 * 13) \rightarrow 20 + (156) \rightarrow 176$ |
| 2. $17/15 + 30$ | $17 / 15 + 30 \rightarrow (17 / 15) + 30 \rightarrow 1.1333... + 30 \rightarrow 31.1333...$ |
| 3. $17/15 * 30$ | $17 / 15 * 30 \rightarrow 34$ |
| 4. $(17/15) * 30$ | $(17 / 15) * 30 \rightarrow 34$ |

Ejercicio 2 ★

Dar expresiones de código tal cual como las escribiría en python para calcular las siguientes expresiones matemáticas.

- $\frac{x}{y} + 1$
- $\frac{x+y}{x-y}$
- $\frac{x+\frac{y}{z}}{x-\frac{y}{z}}$
- $[(x+y)^2]^2$

Ejercicio 3

- Escribir la expresión matemática que representa al resultado de la suma de dos números cualesquiera.
- Escribir en papel un programa que pida al usuario dos valores cualquiera e imprima en pantalla la suma de ambos.

Ejercicio 4

- Escribir una expresión matemática que dado un número natural evalúe al siguiente número más grande. Por ejemplo, dado el número 1 el resultado de evaluar la expresión es 2, dado 37 el resultado es 38.
- Escribir en papel un programa que pida al usuario un valor entero y devuelva el siguiente número más grande.

Ejercicio 5

- Escribir una expresión matemática que describa el promedio entre dos números.
- Escribir en papel un programa que pida al usuario dos números y muestre por pantalla su promedio.

Ejercicio 6 ★

Escribir un programa en la computadora que imprima en pantalla

Mi primer programa Python

Ejercicio 7

Escribir un programa en la computadora que imprima en pantalla

```
v  
e  
r  
t  
i  
c  
a  
l
```

con un solo `print`. Ayuda, usar: `print("\n")`

Ejercicio 8 ★

Escribir un programa en Python que imprima la siguiente figura en la pantalla:

```
*****  
*   *  
*   *  
*   *  
*   *  
*****
```

con un solo `print`.

Ejercicio 9 ★

Escribir un programa en Python que imprima la siguiente figura en la pantalla utilizando un solo `print`:

```
*  
***  
*****  
*****  
*****
```

Ejercicio 10 ★

Escribir un programa en Python que pida un dato al usuario y lo imprima por pantalla.

Ejercicio 11 ★

Escribir un programa en Python que pida al usuario que ingrese un valor y luego imprima en la pantalla:

```
*****  
* El resultado es: <valor>  
*****
```

donde en vez de `<valor>` debe mostrarse el valor que el usuario ingresó por teclado.

Ejercicio 12 ★

Pasar los ejercicios 2, 3, 4 y 5 hechos en papel a programas en Python y probarlos.

Ejercicio 13 ★

Suponga que una persona desea invertir su capital en un banco y quiere saber cuánto dinero tendrá en su cuenta si el banco incrementa el 6% mensual(no acumulativo). Se le debe pedir al usuario el capital a invertir y la cantidad de meses.

Ejercicio 14 ★

Una empresa telefónica desea un programa para calcular el importe de sus clientes. Sabiendo que el costo por comunicación es de \$7 y por cada segundo se cobra \$0,5 hacer dicho programa. Se deben ingresar la cantidad de llamadas realizadas y el tiempo total de comunicación, el programa debe devolver el precio a cobrar.

Ejercicio 15 ★

Un vendedor recibe un sueldo base de \$22000 más un 10% extra del total de sus ventas, el vendedor desea saber cuánto dinero obtendrá en total en el mes si ha logrado realizar 3 ventas este mes. Tenga en cuenta el sueldo básico y la comisión por las ventas. Hacer un programa que solicite el monto de cada una de las ventas del mes e indique cuál será el sueldo final del vendedor.

Ejercicio 16 ★

- Determinar cuántos segundos tiene una hora, y cuántos tiene un día.
- Escribir una expresión matemática que transforme un lapso de tiempo expresado en segundos a uno expresado en minutos.
- Escribir otra para transformar a horas y una última que transforme a días.
- Escribir un programa en Python que pida al usuario una cantidad de segundos y muestre cuantos minutos son, así como también cuantas horas y cuantos días.

Ejercicio 17 ★

La empresa que administra los cajeros automáticos lo contrata para que programen la entrega de billetes, el usuario ingresa la cantidad de dinero que desea y usted debe indicar cuantos billetes de cada denominación se debe entregar. Es importante entregar siempre la menor cantidad de billetes. **Ayuda:** El operador % da el resto de la división a/b , y el operador // da la parte entera del cociente de a/b .

Ejercicio 18 ★

- Escribir una expresión que dada una cantidad de segundos, evalúe a la cantidad de horas que éstos representan, pero esta vez descontando el total de días completos. Por ejemplo: 90.000 segundos equivalen exactamente a 1 día y 1 hora, por lo que dados 90.000 segundos la expresión debería dar 1 hora.
- Ídem punto anterior para minutos y segundos.
- Escribir un programa en Python que pida al usuario una cantidad de segundos y muestre en pantalla la cantidad de días, minutos y segundos que representa. Por ejemplo, si el usuario ingresa 86461 segundos el programa debe mostrar por pantalla: 1 día 0 horas 1 minuto 1 segundo.

Ejercicio 19 ★

Escribir en Python un programa que pida al usuario que ingrese dos valores y los guarde en dos variables, x e y . El programa deberá intercambiar los valores de x e y y luego mostrar en pantalla:

El valor de x es: $\langle x \rangle$

El valor de y es: $\langle y \rangle$

donde en lugar de $\langle x \rangle$ e $\langle y \rangle$ deberá mostrarse el valor de las respectivas variables.

Ejercicio 20 ★

Escribir en Python un programa que pida al usuario que ingrese tres valores y los guarde en tres variables, x , y , y z . El programa deberá intercambiar circularmente los valores de x , y y z , es decir, x debe tomar el valor de y , y el de z y z el de x . Y luego mostrarlos en pantalla:

```
El valor de x es: <x>
El valor de y es: <y>
El valor de z es: <z>
```

donde en lugar de $\langle x \rangle$, $\langle y \rangle$ y $\langle z \rangle$ deberá mostrarse el valor de las respectivas variables.