

```

rectangulo.py x
1  # Ingreso de datos
2
3  base = int(input("Ingrese la base del rectángulo: "))
4  altura = int(input("Ingrese la altura: "))
5
6  # Cálculos
7
8  perimetro = (base + altura) * 2
9  superficie = base * altura
10
11 # Presentación de resultados
12
13 print("El rectangulo tiene un perímetro de", perimetro)
14 print("Y una superficie de", superficie)

```

Observaciones:

1. Todas las líneas empiezan en la misma columna.
2. Las variables son sensibles a mayúsculas y minúsculas.
3. El ingreso de datos se hace con input, pero para obtener un número hay que convertir.
4. Los cálculos se realizan con operadores, se agregan paréntesis para cambiar la prioridad.
5. Print permite imprimir varios valores constantes y variables separados por comas. Agrega espacios y retorno de carro automáticamente.
6. Las líneas que comienzan con # son comentarios.
7. Los programas se escriben en archivos de extensión .py
8. Un programa puede ser iniciado desde la línea de comandos con python (en Windows) o python3 (en linux) seguido del nombre del programa.

Detalle de cada una:

1. En python los espacios tienen significado. A diferencia de la mayoría de los lenguajes modernos, en los que podemos dejar indentación arbitraria, en python todas las líneas deben estar encolumnadas, y si queremos indentar debemos dejar todas las líneas del programa siempre con el mismo margen. Esto parece caprichoso, pero termina siendo útil, porque se simplifican las construcciones condicionales y repetitivas y además nos obliga a ser prolijos con el código fuente. A medida que avancemos vamos a entender mejor para que sirve esta regla.
2. Python al igual que la mayoría de los lenguajes modernos es sensible a mayúsculas y minúsculas, es decir que dos variables pueden tener el mismo nombre pero al menos una diferencia de mayúscula en una letra y son dos variables diferentes. Lo mismo pasa con las instrucciones, print() imprime en la pantalla pero Print() no existe y dará error.
3. Para ingresar datos por teclado se usa la función input. La misma recibe como parámetro (es decir, dentro de los paréntesis) el texto que se desea mostrar al usuario. La función muestra ese texto y detiene el programa hasta que el usuario escriba algo y presione la tecla Enter. Luego lo que el usuario escriba es retornado y el programa continúa. Por eso lo más habitual es que la función input siempre esté a la derecha de una operación de asignación.  
Sin embargo, input sólo devuelve datos de tipo texto, si necesitamos que escriba un número para guardarlo en una variable numérica poseemos la función int(). La misma recibe un texto como parámetro y lo intenta convertir a un número, en tal caso retorna el número como un dato de tipo

entero. Si no lo puede convertir porque el texto no representa un número, el programa se interrumpe y muestra un mensaje de error.

Para números con decimales disponemos de la función `float()`.

4. Para hacer operaciones aritméticas disponemos de operadores similares a los de la matemática, como `+`, `-` o `/`. Para multiplicar se usa `*`. De la misma forma que en matemática, hay una prioridad que establece cuáles operaciones se realizan primero y cuales despues, y en general el orden que python va a usar es el más cercano a la aritmetica. Es decir, que en la operación  $2 + 3 * 4$ , el resultdo será 14 y no 20. De la misma forma que en la matemática podemos agregar paréntesis para cambiar este orden, así si en el cálculo anterior necesitamos que se realice primero la suma hay que redactar  $(2+3)*4$ .

5. La función `print` permite imprimir una secuencia de valores constantes o variables. En las líneas 13 y 14 se están imprimiendo textos constantes, por eso van separados por comillas dobles y luego sendas variables. Al imprimir en la pantalla se agrega automáticamente un espacio entre cada valor y un retorno de carro al final. En el caso de necesitar cambiar esta funcionalidad se le pueden indicar mediante parámetros otros separadores y caracteres de finalización.

Los valores constantes de tipo texto van separados por comillas dobles como en la mayoría de los lenguajes pero también pueden usarse comillas simples indistintamente, con la precaución obvia de utilizar el mismo tipo de comillas al inicio y al final del texto.

6. Python tiene tres tipos de comentarios al igual que C# y Java. Por ahora vamos a usar solamente el comentario de fin de línea. En cualquier línea podemos escribir `#` y todo el resto de la línea va a ser ignorado, permitiéndonos escribir cualquier tipo de texto. Los comentarios son muy importantes y aunque nos dé fiaca hay que tomarlos con seriedad. Generalmente se utilizan para dejar algún tipo de aclaración o explicación para otros programadores (o nosotros en el futuro).

7. Los programas de python deben escribirse en archivos de texto simple, pero con extensión `.py`. Para escribirlos podemos usar cualquier editor de texto, pero es preferible usar alguno que colabore con la redacción, agregando coloreo de sintaxis, marcas de error, completamiento de código, etc.

8. Para ejecutar un programa que hayamos escrito podemos invocar al intérprete del lenguaje directamente de la línea de comandos del sistema operativo. En el caso de windows, vamos hasta la carpeta donde hayamos guardado el programa y arrancamos el intérprete de comandos o el powershell y escribimos: `python nombreprograma.py`

En el caso de linux vamos a necesitar escribir: `python3 nombreprograma.py`

```
diego@diego-Inspiron-5567:~/Trabajo/Cursos/Python-Mensa$ python3 rectangulo.py
Ingrese la base del rectángulo: 10
Ingrese la altura: 20
El rectangulo tiene un perímetro de 60
Y una superficie de 200
diego@diego-Inspiron-5567:~/Trabajo/Cursos/Python-Mensa$ █
```