

Uso y definición de funciones

CC1001–Prof. Romain Robbes

Solución Ejercicio 1

Escribir un programa que establezca el diálogo indicado en el siguiente ejemplo:

Calcular velocidad de un móvil

```
distancia(metros)?  
100.0
```

```
tiempo(segundos)?  
9.78
```

```
velocidad = 36.81 km/hora
```

Programa

```
print "Calcular velocidad de un movil"  
  
print "distancia (metros)?"  
distancia = input()  
  
print "tiempo (segundos)?"  
tiempo = input()  
  
velocidad = (distancia/1000.0) / (tiempo/3600.0)  
print "velocidad = ", velocidad, " km/hora"
```

Explicaciones detalladas

sintaxis

```
print exp, ... ,
```

semántica

escribe expresiones hacia el lado y deja el cursor al final de la línea

Nota

```
print "velocidad = ", 3.6 * d/t, " km/hora"
```

equivale a

```
print "velocidad = ",  
print 3.6 * d/t,  
print " km/hora"
```

Alternativa:

```
Calcular velocidad de un movil  
distancia(metros)?100.0  
tiempo(segundos)?9.78  
velocidad = 36.81 km/hora
```

Nota. Pregunta del computador y respuesta del usuario aparecen en la misma línea

```
print "Calcular velocidad de un movil"  
  
d = input("distancia (metros)?")  
  
t = input("tiempo (segundos)?")  
  
print "velocidad = ", 3.6 * d/t, " km/hora"
```

Explicaciones detalladas

```
d = input("distancia(metros)?")
```

equivale a

```
print "distancia(metros)?", #no salta linea  
d = input()
```

Strings, tipos, y comentarios

```
print "hola"

"hoLa" es de tipo string
42 es de tipo int
2.5 es de tipo float

int(9.9)
float(4)
int("3")
int("hello")
```

comentario (observación, explicación)
 #un comentario no se ejecuta
 #es solo por el usuario

"""Eso es un comentario tambien,
 pero puede ocupar varias lineas
 de texto. """

Problema

Establecer las instrucciones principales de un programa que establezca el siguiente diálogo:

```
Perimetro y area de triangulo de lados a,b, y c
a? ____
b? ____
c? ____
Perimetro = numero
Area = numero
```

Nota. El area se calcula como $a = \sqrt{s(s-a)(s-b)(s-c)}$, con s el semi-perimetro del triangulo $s = (a + b + c) / 2$

¿Como calcular \sqrt{x} ?

La función que hace la computation de la raiz ya existe; tenemos que llamar (o hacer una invocación) de **sqrt**. Cualquier expression puede llamar una funcion.

ejemplo:
 import math
 math.sqrt(s * (s - a) * (s - b) * (s - c))

sintaxis
 modulo.nombre(argumentos)
 El modulo es opcional. Hay 0 o mas argumentos.

semantica

1. Evaluar argumentos, si hay.
2. Evaluar función con valores de argumentos
3. Recibir resultado de la función (al punto de invocación)

Funciones matemáticas definidas en el modulo **math**

función	significado	ejemplo	resultado
math.sqrt(x)	\sqrt{x} , $x \geq 0$	math.sqrt(4)	2.0
math.pow(x,y)	x^y	math.pow(4,0.5)	2.0
math.exp(x)	e^x	math.exp(1)	2.71...
math.log(x)	$\log_e x$	math.log(math.e)	1.0
math.sin(x)	seno de angulo x	math.sin(math.pi)	0.0
math.cos(x)	coseno de x	math.cos(math.pi)	-1.0
math.tan(x)	tangente de x	math.tan(math.pi)	0.0
math.asin(x)	arco-seno de x	math.asin(0)	3.14...
math.acos(x)	arco-coseno x	math.acos(-1)	3.14...
math.atan(x)	arco-tangente x	math.atan(0)	3.14...

Argumentos son de tipos int o float, el resultado es de tipo float
 No olvidarse de importar el modulo math (**import math**)
 Documentación disponible con **help(math)** y **dir(math)**

Otras funciones predefinidas

función	significado
abs(x)	$ x $, es decir, x si $x \geq 0$ o $-x$ si $x < 0$
max(x,y,...)	mayor entre x,y,...
min(x,y,...)	menor entre x,y,...

función	significado
random.random()	$0 \leq n < 1$: float al azar en $[0,1[$
random.randint(x,y)	$x \leq n \leq y$: int al azar en $[x,y]$ con $x \leq y$

```
abs(-1) # 1
max(3,4,2) # 4
min(3,4,2) # 2
max(2,3.5) # 3.5
min(2,2) # 2

import random
random.random()
# 0.3504750773904878
random.random()
# 0.013316766955305481
random.randint(1,6)
# lanzar un dado: 4
random.randint(0,1)
# cara o sello: 1
```

No olvidar help() y dir()!

Funciones definidas por el programador

sintaxis
 def nombre(parámetros):
 instrucciones (cero o más)
 return expresión

¿parámetros? nombre, ... (cero o más)

semántica

1. copiar (asignar) argumentos de llamada en parámetros de función
2. ejecutar las instrucciones (si es que existen)
3. calcular valor de la expresión
4. devolver valor de la expresión al lugar de invocación (uso)

Ejemplo: triangulo

Parametros y variables en funciones

Una función puede definir variables en su cuerpo.

Los parámetros y variables definidos en una función desaparecen al final de la función.

Las funciones pueden ver variables definidas a fuera de la función.

Si hay nombres de variables/parametros iguales (...), se usa la variable/parametro definida en la función.

Ejemplo: [online python tutor](#)

Beneficios de funciones

Modularización de código
Documentación de código
Reuso de código

Que hace eso?

```
def area(x,y): return (3.14*x**2)-(3.14*y**2)
```

El mismo código

```
def area_circulo(radio):  
    """  
    area_circulo: num -> float  
    calcula la area de un circulo con su radio (positivo)  
    ejemplo: area_circulo(3) devuelve 28.26  
    """  
    pi = 3.14  
    return pi * radio ** 2  
  
def area_anillo(exterior, interior):  
    """  
    area_anillo(exterior, interior): num num -> float  
    calcula la area de un anillo con sus radios exterior y  
    interior  
    (exterior >= interior)  
    ejemplo: area_anillo(3,2) devuelve 15.70  
    """  
    area_total = area_circulo(exterior)  
    area_hoyo = area_circulo(interior)  
    return area_total - area_hoyo
```

La receta de diseño

```
def area_circulo(radio):  
    """  
    area_circulo: num -> float  
    calcula la area de un circulo con su radio (positivo)  
    ejemplo: area_circulo(3) devuelve 28.26  
    """  
    pi = 3.14  
    return pi * radio ** 2
```

Nombre descriptivos: por función, parámetros, y variables

Contrato: tipos de argumentos y valor de retorno

Propósito: qué hace la función (**no como**)

Ejemplos: ejemplos de uso de la función, y resultado esperado

Cuerpo: instrucciones para hacer su computación
el cuerpo se escribe **en último!**

La receta de diseño es obligatoria!

Parece demasiado por ejemplos chicos, pero sirve a acostumbrarse.

Una vez que el código crece, se hace muy útil para tener un buen diseño, y para evitar errores.

Si Uds no siguen la receta de diseño, van a perder puntos.

La receta de diseño es inspirada del libro "How to design programs" (muy recomendado!). Se encuentra gratis a:

<http://htdp.org/>

Antes de hacer el ejercicio, algo sobre disciplina

No puedo responder a todas las preguntas, entonces no respondo a ninguna.

Si Ud. tiene una pregunta, pensar más y mirar el "handout". La capacidad de buscar de su mismo es muy importante en computación (y en la vida).

No se puede levantar de su silla. No se puede hablar tampoco, por el ruido que hace y la molestia por otras clases.

Los ejercicios son hechos para aprender. Tienen un peso muy bajo por las notas. Entonces no es un problema si uno no puede hacer uno perfectamente.

Al final del tiempo, devolver las hojas por adelante.

Ejercicio (seguir la receta de diseño)

1. Escribir una función que calcule la distancia entre dos puntos ubicados en las coordenadas (x_1, y_1) y (x_2, y_2) .

2. Escribir un programa que lea las coordenadas de 3 puntos y que, usando funciones, calcule el área del triángulo que forman de acuerdo al siguiente diálogo:

x1? ____
y1? ____
x2? ____
y2? ____
x3? ____
y3? ____
Area = n°