¿Computacion?

El pensamiento computacional son los procesos de pensamiento implicados en la formulación de problemas y sus soluciones para que las soluciones están representados en una forma que pueda ser efectivamente realizada por un agente de procesamiento de información. [Cuny, Snyder, Wing]

Hay dos tipos de conicimientos: declarativo y imperativo

La raíz cuadrada de X es el numero Y, tal que Y cuadrado es igual a X, con Y positivo

vs:

1. Adivinar un numero Y
2. Si Y cuadrado es "cerca" de X, devolver Y
3. Otro, Y se cambia a: 0.5 * (Y + X/Y). Repetir el paso 2

Nececitamos operaciones "primitivas"

In 1936, Turing ha mostrado que se necesita solo 6 primitivas, para describir un algoritmo. (Turing completeness)

Cada programa escrito en un idioma Turing-complete se puede traducir en otro idioma Turing-complete (Turing equivalence).

Elegimos Python porqué permite de aprender facilmente.

Problema

Escribir un programa para que el computador calcule los porcentajes
de hombres y mujeres de acuerdo al siguiente diálogo con una
persona (usuario):

```
nº de hombres?
37  <– persona

nº de mujeres?
4   <– persona

% hombres = 90.2

% mujeres = 9.8
```

Algoritmo

Etapas (pasos, instrucciones) que debe realizar (ejecutar) el computador para resolver el problema. Ejemplo:

1. escribir (mostrar) en la pantalla la frase `"nº de hombres?"`

2. leer (obtener) el n° ingresado por la persona usando el teclado

3. escribir (mostrar) en la pantalla `"nº de mujeres?"`

4. leer el n° ingresado por la persona

5. calcular el %(p) de hombres c/r a suma de hombres(h) y mujeres(m)
como h / (h + m) x 100

6. escribir una línea con la frase **"**`% hombres = `**"** y el n° p

7. escribir una línea con **"**`% mujeres = `**"** y el n° 100-p

Programa: Algoritmo expresado en un lenguaje de programación

Programa en lenguaje Python

```
print "numero de hombres?"
h = input()
print "numero de mujeres?"
m = input()
p = 100.0 * h / (h+m)
print "% hombres = ",p
print "% mujeres = ",100 – p
```

Lenguaje Python: características generales

- **inspirado en idioma inglés**
  - palabras print, input
- **uso de notación matemática**
  - expresiones: 100.0*h/(h+m), p-100
  - funciones: input( )
- **sintaxis** (gramática, escritura) reglamentada
  - cada instrucción en una línea
  - palabras con significados especiales (input, print)
  - uso de minúsculas (salvo casos especiales)
- **semántica** (interpretación) precisa (sin ambigüedades)
  - instrucciones con significado/efecto bien definido

Sintaxis vs semantica

Azul gato transantiago el.

Ideas verdes sueñan con furia.

Esta frase es correcta y tiene sentido.

Explicaciones detalladas

```
print "numero de hombres?"
```

escribir (mostrar) en la pantalla la frase "nº de hombres?"

Sintaxis:
`print "..."`

Semántica:
escribe … en la pantalla del computador

Explicaciones detalladas

```
h = input()
```

lee un número desde el teclado y lo guarda en la variable h

Sintaxis
`variable = input()`

Semántica
Lee un nº y lo guarda en la variable indicada

¿Variable?

representación simbólica de un valor (número)
representa una ubicación (celda) en la memoria del computador
capacidad: **un** valor/número
puede variar (cambiar) su valor
posee un nombre (letra seguida de letras, dígitos o_)

Tipo de la variable

int si contiene un nº entero (sin punto decimal)
Ejs: 0, -1, 123

float si contiene un nº real (con punto decimal)
Ejs: 100.0, 0.001

Explicaciones detalladas

```
input()
```

Función que lee un nº desde el teclado y lo entrega como resultado

Semántica:

1. espera que el usuario ingrese un número (dígitos y enter)
2. lee (obtiene) el número
3. entrega el número como resultado

Explicaciones detalladas

```
p = 100.0 * h / (h+m)
```

calcula 100.0*h/(h+m) y guarda el resultado en la variable p

Instrucción de asignación

sintaxis

`variable = expresión`

semántica

1. evaluar expresión (calcular resultado)

2. asignar/guardar resultado a/en la variable

Expresión

sintaxis
- se escribe en una línea (hacia el lado) y no en varios niveles
- componentes: variables, constantes, operadores, funciones

semántica
se calcula el resultado de la expresión de acuerdo a la siguiente prioridad de los operadores:

1. operadores unarios: `+, −`
2. operador de elevación a potencia: `**`
3. operadores "multiplicativos": `*, /`
4. operadores "aditivos": `+, −`

Ej: `−a+b*c` es `(−a)+(b*c)`, similar a álgebra

Expresión: semántica

- a **igual prioridad**, se evalúa de izquierda a derecha.
  Ej: a*b/c es (a*b)/c

- **paréntesis** modifican o confirman orden de evaluación.
  ejs: (a+b)*c modifica, a+(b*c) confirma

- ¿**tipo del resultado**?
  tipo común. Ej: 1 / 2 es 0 (int), 1.0 / 2.0 es 0.5 (float)
  float(real) si operandos son de tipos distintos. Ej: 1.0/2 es 0.5
  ¿resultado de `h/(h+m)*100.0`?

Explicaciones detalladas

```
print "% hombres = ", 100 – p
```

escribe en una misma línea la frase "% hombres = " y el valor de la expresión 100 - p

Sintaxis
`print expresión, …`

expresión puede ser una frase. Ej: print "% hombres = "

Semántica
1. evaluar expresiones (ejs: 100 - p, p)
2. escribir resultados (separados por un espacio)
3. posicionar el cursor al comienzo de la sgte línea en la pantalla

Ejercicio

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)?"
d = input()

print "tiempo (segundos)?"
t = input()

v = (d/1000.0) / (t/3600.0)

print "velocidad = ", v, " km/hora"
```

Alternativa

```
print "Calcular velocidad de un movil"

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

print "tiempo(segundos)?"
t = input()

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

Diálogo:

```
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()
```

comentario (observación, explicación)
#...

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"
```