

Tabla de conversión de un algoritmo a Snap!

Tipos de datos

entero
real
caracter
cadena
logico



Declaración de variables

<tipo> : identificador_1, ...,
identificador_n



Ejemplo:

real x, y

Operadores:

Asignación a una variable:
 $x \leftarrow 3$



- : número negativo (no resta) -3
^ ó ** : exponenciación
 a^b ó $a^{**}b$ (a elevado a b)

 (a elevado a b)

*, /: multiplicar y dividir



div: división entera

Ejemplo:

a div b



mod: módulo

Ejemplo:

a mod b



(...): paréntesis (a+b)



+: Concatenación de cadenas



Operadores relacionales y lógicos

>: mayor que
<: menor que
=: igual
<>: distinto de



and
or
not



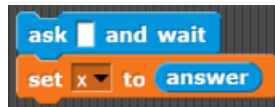
Funciones matemáticas internas:

abs(x)
arctan(x)
cos(x)
exp(x)
ln(x)
log10(x)
redondeo(x)
sen(x)
cuadrado(x)
raiz2(x)
trunc(x)



Funciones de entrada

leer(x)



Funciones internas de salida por pantalla

escribir(elem_1, ..., elem_n)



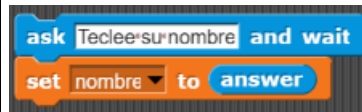
Ejemplo:

escribir('El precio es de ' x, ' euros')



Combinación entrada/salida:

escribir('Teclee su nombre')
leer (nombre)



Plantilla algoritmo:

algoritmo: <nombre_algoritmo>
const
...
var
...
inicio
...
fin

Plantilla equivalente en Snap!:



Plantilla si – sino - finsi:

si condición entonces
 acción 1
sino
 acción 2
fin_si

Plantilla if – else - endif:



Plantilla mientras:

mientras condición hacer
 acción
fin_mientras

Plantilla while:



Plantilla para:

para i = 1 hasta n hacer
 acción
fin_para

Plantilla for:



para $i = 0$ hasta $n-1$ hacer

acción

fin_para



Obtención de recursos en Internet:

Snap!: <https://snap.berkeley.edu/snap/snap.html>

Pautas de conversión de un algoritmo a Python

Tipos de datos

entero
real
caracter
cadena

El tipo se establece al asignar un valor

Declaración de constantes

<IDENTIFICADOR> = valor

<IDENTIFICADOR> = <valor>;

Ejemplo:

A = 100

A = 100

Declaración de variables

No hace falta declarar variables

<tipo> : identificador_1, ..., identificador_n

Ejemplo:

real x, y

Operadores:

Asignación a una variable:

x <- 3

x = 3

- : número negativo (no resta)

- : número negativo

^ ó ** : exponenciación

a ^ b ó a ** b (a elevado a b)

a**b (a elevado a b)

*, /: multiplicar y dividir

*, /: multiplicar y dividir

div: división entera

//

Ejemplo:

a div b

int (a // b)

mod: módulo

Ejemplo:

a mod b

a % b

(...): paréntesis

(...)

+: Concatenación de cadenas

+

<p>Operadores relacionales y lógicos</p> <p>>: mayor que <: menor que =: igual <>: distinto de</p> <p>and or not</p> <p>Funciones matemáticas internas:</p> <p>abs(x) arctan(x) cos(x) exp(x) ln(x) log10(x) redondeo(x) sen(x) cuadrado(x) raiz2(x) trunc(x)</p> <p>Funciones internas de entrada por teclado</p> <p>Destino cadena: cad</p> <p>leer(cad)</p> <p>Destino entero: a</p> <p>leer(a)</p> <p>Destino real: x</p> <p>leer(x)</p> <p>Funciones internas de salida por pantalla</p> <p>escribir(elem_1, ..., elem_n)</p> <p>Ejemplo:</p> <p>escribir('El precio es de ', x, ' euros')</p>	<p>> < == !=</p> <p>and or not</p> <p>Hay que hacer import.math</p> <p>math.abs(x) math.atan(x) math.cos(x) math.exp(x) math.log(x) math.log10(x) math.floor(x + 0.5) math.sin(x) math.pow(x, 2) math.sqrt(x) math.floor(x)</p> <p>cad = raw_input()</p> <p>a = input()</p> <p>x = input()</p> <p>print "El precio es de ", x, " euros"</p> <p>Sin newline poner , al final</p>
---	--

Combinación entrada/salida:

```
escribir('Teclee su nombre: ')
leer (nombre)
```

suponiendo que nombre se ha declarado
cadena: nombre

Plantilla algoritmo:

```
algoritmo: <nombre_algoritmo>
const
...
var
...
inicio
...
fin
```

Plantilla si – sino - finsi:

```
si condición entonces
    acción 1
sino
    acción 2
fin_si
```

Plantilla mientras:

```
mientras condición hacer
    acción
fin_mientras
```

(Con salto de línea al final: añadir \n)

```
nombre = input("Teclee su nombre: ")
```

Plantilla equivalente en Python:

```
def <nombre_algoritmo>():
    inicio
    ...
    fin
```

Plantilla if – else - endif:

```
if condición:
    acción 1
else:
    acción 2
```

Plantilla while:

```
while condición:
    acción
```


Plantilla para:

para i = 1 hasta n hacer

 acción

fin_para

para i = 0 hasta n-1 hacer

 acción

fin_para

Plantilla for:

for i in range(1, n+1):

 acción

for i in range(0, n):

 acción

Iteración:

for x in array:

 print x

Obtención de recursos en Internet:

Editor Notepad++: <http://notepad-plus-plus.org/download>

Entorno Python: <http://www.python.org/getit>

Pautas de conversión de un algoritmo a C

Tipos de datos

entero
real
caracter
cadena

int
double
char
char[]

Declaración de constantes

<IDENTIFICADOR> = valor

const <tipo> <IDENTIFICADOR> = <valor>;

Ejemplo:

A = 100

const int A = 100;

Declaración de variables

<tipo> : identificador_1, ..., identificador_n

<tipo> identificador_1, ..., identificador_n;

Ejemplo:

real x, y

double x, y;

Operadores:

Asignación a una variable:

x <- 3

x = 3;

- : número negativo (no resta)

- : número negativo

^ ó ** : exponenciación

a ^ b ó a ** b (a elevado a b)

pow(a, b) (a elevado a b)

*, /: multiplicar y dividir

*, /: multiplicar y dividir

div: división entera

Ejemplo:

a div b

(int) (a / b)

mod: módulo

Ejemplo:

a mod b

a % b

(...): paréntesis

(...)

+: Concatenación de cadenas

Función estándar strcat(s1, s2)

<p>Operadores relacionales y lógicos</p> <p>>: mayor que <: menor que =: igual <>: distinto de</p> <p>and or not</p> <p>Funciones matemáticas internas:</p> <p>abs(x) arctan(x) cos(x) exp(x) ln(x) log10(x) redondeo(x) sen(x) cuadrado(x) raiz2(x) trunc(x)</p> <p>Funciones internas de entrada por teclado</p> <p>Destino cadena: cad</p> <p>leer(cad)</p> <p>Destino entero: a</p> <p>leer(a)</p> <p>Destino real: x</p> <p>leer(x)</p> <p>Funciones internas de salida por pantalla</p> <p>escribir(elem_1, ..., elem_n)</p> <p>Ejemplo:</p> <p>escribir('El precio es de ', x, ' euros')</p>	<p>> < == (strcmp(...) = 0 para cadenas) != (strcmp(...) != 0 para cadenas)</p> <p>&& !</p> <p>fabs(x) atan(x) cos(x) exp(x) log(x) log10(x) floor(x + 0.5) sin(x) pow(x, 2) sqrt(x) floor(x)</p> <p>scanf("%s", cad);</p> <p>scanf("%d", &a);</p> <p>scanf("%lf", &x);</p> <p>printf("El precio es de %lf euros\n", euros);</p>
---	--

<p>Combinación entrada/salida:</p> <p>escribir('Teclee su nombre: ') leer (nombre)</p> <p>suponiendo que nombre se ha declarado cadena: nombre</p> <p>Plantilla algoritmo:</p> <p>algoritmo: <nombre_algoritmo> const ... var ... inicio ... fin</p> <p>Plantilla si – sino - finsi:</p> <p>si condición entonces acción 1 sino acción 2 fin_si</p> <p>Plantilla mientras:</p> <p>mientras condición hacer acción fin_mientras</p>	<p>(Con salto de línea al final: añadir \n)</p> <p>printf("Teclee su nombre: "); scanf ("%s", nombre);</p> <p>suponiendo que nombre se ha declarado char nombre[<tamaño>];</p> <p>Plantilla equivalente en C:</p> <pre>#include <stdio.h> main() { /* const */ ... /* var */ ... /* inicio */ ... } Plantilla if – else - endif: if (condición) { acción 1 } else { acción 2 } Plantilla while: while (condición) { acción }</pre>
---	--

Plantilla para:

para i = 1 hasta n hacer

 acción

fin_para

para i = 0 hasta n-1 hacer

 acción

fin_para

Plantilla for:

```
for (i = 1; i <= n; i++) {
```

```
    acción
```

```
}
```

```
for (i = 0; i < n; i++) {
```

```
    acción
```

```
}
```