

Clase 05 - Control de flujo condicional

IIC1103-07 - Introducción a la Programación

Cristian Ruz – `cruz@ing.puc.cl`

Jueves 22-Agosto-2019

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

Laboratorios

Recapitulación

Control de Flujo Condicional

Operadores de control de flujo condicional

Laboratorios

Recapitulación

Control de Flujo Condicional

Operadores de control de flujo condicional

Laboratorios: desde el Lunes 27-Agosto

- Laboratorios evaluados automáticamente (*hackerrank*)
- Pueden ir a cualquier laboratorio (Lunes a Jueves)
 - SIN LAPTOP
 - Lunes a Jueves, mód 5 y 6: Lab San Agustín (piso 2)
 - CON LAPTOP
 - Martes, mód 5 y 6: K200, A5, B12
 - Miércoles, mód 5 y 6: **B23**, CS203, C203
 - Jueves, mód 5 y 6: B13, K204, CS101
- ¡Este es un curso práctico! Aprender haciendo

Laboratorios

Recapitulación

Control de Flujo Condicional

Operadores de control de flujo condicional

¿En qué vamos?

1. Variables y expresiones
2. Control de flujo
3. Funciones y recursión \Leftarrow 11: 24-Sept, 18:30
4. Strings
5. Listas
6. Tipos de datos personalizados (objetos)
7. Ordenación y búsqueda
8. Archivos

En la clase anterior...

Tipos de datos: `int`, `float`, `str`, `bool`

Sintaxis asignación

```
1 nombre_variable = valor
```

Una expresión puede mezclar `+`, `-`, `*`, `/`, `**`, `//`

En la(s) clase(s) anterior(es)...

Tipos de datos

```
1 a = 3
2 type(a)          #<class 'int'>
3 a = "3.55"
4 type(a)          #<class 'str'>
5 a = 3.55
6 type(a)          #<class 'float'>
```

Variables, asignaciones y operaciones matemáticas

```
1 a = (0.15 * I1 + 0.15 * I2 + 0.3*E)/0.6
```


Escribiendo en pantalla: print

`print` permite imprimir en consola.

Sintaxis print

```
1 print(variable1, variable2, variable3, ...)
```

```
1 cobreloa = 0           # goles del local
2 antofagasta = 0        # goles de la visita
3 cobreloa += 1          # gol local
4 antofagasta += 1       # gol visita
5
6 print("Local", cobreloa, "-", antofagasta, "Visita")
7 # >>> "Local 1 - 1 Visita"
```

Recibiendo entrada del usuario: input

`input` entrega un *str* con un valor ingresado por el usuario.

Sintaxis input

```
1 valorStr = input(mensaje_para_usuario)
```

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un texto: ")
3 print(a)
```

¿Cómo pido un `int`, `float` o `bool` al usuario?

Input

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un str: ")
3
4 # pido un int al usuario
5 b = int(input("Ingrese un int: "))
6
7 # pido un float al usuario
8 c = float(input("Ingrese un float: "))
9
10 # pido un bool al usuario
11 d = bool(input("Ingrese un bool: "))
12
13 # imprimo valores obtenidos
14 print(a,b,c,d)
```

Para un marcador...

```
1 local = 0          # goles del local
2 visita = 0         # goles de la visita
3
4 # Guardo el equipo que anotó en variable "gol"
5 # [1: local, 2: visita, otro: fin partido]
6 gol = int(input("?Qué equipo anotó el gol?"))
7 print("!Gol del equipo:",gol,"!")
```

¿Y después?

```
1 local = 0          # goles del local
2 visita = 0         # goles de la visita
3
4 # Guardo el equipo que anotó en variable "gol"
5 # [1: local, 2: visita, otro: fin partido]
6 gol = int(input("?¿Qué equipo anotó el gol?"))
7
8 print("Gol del equipo: " + str(gol))
9 local += 1         # gol local
10 visita += 1        # gol visita
11
12 print("Local", local, "-", visita, "Visita")
```

Input

¿Cómo sabemos a quién sumarle el gol?

```
1 local = 0          # goles del local
2 visita = 0         # goles de la visita
3
4 # Guardo el equipo que anotó en variable "gol"
5 # [1: local, 2: visita, otro: fin partido]
6 gol = int(input("?¿Qué equipo anotó el gol?"))
7
8 print("Gol del equipo: " + str(gol))
9 local += 1         # gol local
10 visita += 1        # gol visita
11
12 print("Local", local, "-", visita, "Visita")
```

Laboratorios

Recapitulación

Control de Flujo Condicional

Operadores de control de flujo condicional

Instrucciones condicionales

Permiten ejecutar una o más instrucciones **solamente** si se cumple una condición

Condición

Expresión cuyo valor es de tipo `bool`.

Dos valores posibles: `True` ó `False`

Ejemplo práctico:

```
1 a = int(input("Ingrese número del 0 al 9: "))  
2 condicion = (0 <= a) and (a <= 9)  
3 print(condicion)
```

Control de flujo: if, elif y else

if: Ejecuta un trozo de código *si* se cumple una condición.

Sintaxis if

```
1 if condicion:                #debe entregar bool
2     bloque_de_codigo_if
3     bloque_de_codigo_if
4     ...
5     bloque_de_codigo_if
6 bloque_de_codigo_fuera_de_if
```

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 condicion = (0 <= a) and (a <= 9)
3 if not(condicion):
4     print("!Error! número mal ingresado")
5 print("! Gracias!")
```

Control de flujo: if, elif y else

if: Ejecuta un trozo de código *si* se cumple una condición.

Sintaxis if

```
1 if condicion:                #debe entregar bool
2     bloque_de_codigo_if
3     bloque_de_codigo_if
4     ...
5     bloque_de_codigo_if
6 bloque_de_codigo_fuera_de_if
```

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 if not((0 <= a) and (a <= 9)):
3     print("!Error! número mal ingresado")
4 print("!Gracias!")
```

También se puede poner la condición directamente en el *if*.

Control de flujo: if, elif y else

else: Ejecuta un trozo de código *si* no se cumple la condición de un **if** anterior.

Sintaxis if-else

```
1 if condicion:           #debe entregar bool
2     bloque_de_codigo_if
3     ...
4     bloque_de_codigo_if
5 else:                   #si la condicion de if es False
6     bloque_de_codigo_else
7     ...
8     bloque_de_codigo_else
9 bloque_de_codigo_fuera_de_if_else
```

Control de flujo: if, elif y else

Ejemplo:

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 condicion = (0 <= a) and (a <= 9)
3 if condicion:
4     print("! Gracias!")
5 else:
6     print("! Error! número mal ingresado")
```

Control de flujo: if, elif y else

elif: ejecuta un trozo de código si se cumple una condición y no se ha cumplido ningún **if** o **elif** anterior.

Sintaxis

```
1 if condicion:           #debe entregar bool
2     bloque_de_codigo_if ...
3     bloque_de_codigo_if
4 elif condicion:         #si if es False y elif es True
5     bloque_de_codigo_elif ...
6     bloque_de_codigo_elif
7 else:                   #si if es False y elif es False
8     bloque_de_codigo_else ...
9     bloque_de_codigo_else
10 bloque_de_codigo_fuera_de_if_elif_else
```

- (1) debe existir un **if**
- (2) pueden haber varios **elif**
- (3) podría no haber **else**

Control de flujo: if, elif y else

Ejemplo:

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 if 0 > a:
3     print("Error, número menor que cero")
4 elif a > 9:
5     print("Error, número mayor que 9")
6 else:
7     print("! Gracias!")
```

Obs: cada `if` puede contener varias instrucciones.

Control de flujo: if, elif y else

¿Cuál es la diferencia entre estos dos códigos?

```
1 a = int(input("a: "))
2 if 0 < a:
3     print(1)
4 if 5 < a:
5     print(2)
6 if 10 < a:
7     print(3)
8 if 15 < a:
9     print(4)
10 else:
11     print(5)
```

```
1 a = int(input("a: "))
2 if 0 < a:
3     print(1)
4 elif 5 < a:
5     print(2)
6 elif 10 < a:
7     print(3)
8 elif 15 < a:
9     print(4)
10 else:
11     print(5)
```


Control de flujo: if, elif y else

Código modificado

```
1 local = 0          # goles del local
2 visita = 0         # goles de la visita
3 # [1: local, 2: visita, otro: fin partido]
4 gol = int(input("?¿Qué equipo anotó el gol?"))
5 print("Gol del equipo: " + str(gol))
6 local += 1         # gol local
7 visita += 1        # gol visita
8
9 print("Local", local, "-", visita, "Visita")
10 # Ahora veo quien gana
11 if local > visita:
12     print("!Ganó el local!")
13 else if local < visita:
14     print("!Ganó la visita!")
15 else:
16     print("!Empate!")
```