

# **CAPACITACIÓN PYTHON**

**NELSON R. SALINAS**

**IDEAM - SMBYC**

# CONTROL DE FLUJO

1. Utilidad
2. if, elif, else
3. for, while, break, continue
4. Paréntesis

# CONTROL DE FLUJO

Manipular el comportamiento del código durante su ejecución.

1. ¿Temprano en la oficina?
  - a. Si: Café + noticias
  - b. No: Solo café

# PRUEBAS CONDICIONALES

- `if`
- `else`
- `elif`

# PRUEBAS CONDICIONALES

```
myname = "Nelson"

if myname == "Nelson":
    print "Pasé"

if myname != "Nelson":
    print "Adiós"

if not myname == "Nelson":
    print "Por favor no insista"
```

# PRUEBAS CONDICIONALES

```
myname = "Nelson"
mylastname = "Salinas"

if myname == "Nelson" and mylastname == "Salinas":
    print "Acceso garantizado"

if myname != "Nelson" or mylastname != "Salinas":
    print "Adiós"

if not myname == "Nelson" and not mylastname == "Salinas":
    print "Por favor no insista"
```

# PRUEBAS CONDICIONALES

```
precipitation = 9000 # mm anuales
chave_forest = None

if precipitation <= 1500:
    chave_forest = 'dry'

elif precipitation <= 3500:
    chave_forest = 'moist'

else:
    chave_forest = 'wet'
```

# ITERACIONES

- Repeticiones de expresiones
- `for`, `while`
- `continue`, `break`



# ITERADORES

```
for item in [1, 2, 3, 4]:  
    print item ** 2  
# 1  
# 4  
# 9  
# 16
```

# ITERADORES

```
counter = 5
while counter > 0:
    print "Tomar cafe en", counter, "segundos"
    counter = counter - 1
```

# ITERADORES

```
for i in range(100):  
    print i  
    if i > 20:  
        print "Me canse, voy por un cafe."  
        break
```

# ITERADORES

```
for i in range(1950,2020):  
    if i >= 1980 and i < 1990:  
        print "Los 80s no me gustaron."  
        continue  
    print "Feliz año", i
```

# ITERADORES

Evitar repeticiones infinitas.

```
counter = 1
while counter > 0:
    print "Loop infinito"
    counter += 1
```

# PARÉNTESIS

Asigna orden de ejecución a expresiones dentro de un comando.

```
12 + 18 / 3.0 ** 2  
# 14.0
```

```
((12 + 18) / 3.0) ** 2  
# 100.0
```