El nucleo de la programación

Temario de la clase

- Loops
- While.
- For
- Ciclos anidados.
- Contadores.
- Acumuladores.

Bucle.

Repeticion de un conjunto de instrucciones un numero de veces.
La computadora va y vuelve por las mismas instrucciones por lo que se le llama loop, ciclo, bucle, etc.
Representación en un diagrama de flujo.

Bucles con while.

El comando while hace que la computadora repita una serie de órdenes hasta que se cumpla una condición lógica.
Para producir un bucle de 8 iteraciones comenzando con i=1 hasta i=8:

```
i=1
while i<=8:
    print i
    i=i+1
```

Una forma simplificada (pythonica) de poner contadores en python:

```
i+=1
```

esto es exactamente lo mismo que

```
i=i+1
```

Notar que siempre después del while ... : siguen las tabulaciones hasta donde termina la serie de instrucciones que queremos se repitan.

Bucles con for

Otra alternativa para hacer bucles o repeticiones de órdenes es con for.

Este se usa para tomar valores de una lista:

for i in lista de valores:

```
>>> a=['a','b','c']
>>> for i in a:
...     print i,')'
a )
b )
c )
```

Otra forma muy utilizada es usando la generación de listas con range:

```
>>> for i in range(3):
...     print i,')'
0 )
1 )
2 )
```

Bucles con for

El for es similar al while pero mucho mas compacto! bien pythonic!
Recordar que el range permite empezar de cualquier número y terminar, ej. range(2,10,2)
Si tenemos un range(2,5) comenzará en 2 pero terminará en 4! uno antes del número máximo, pero respetando que el número de ciclos es max-min (5-2=3).
Si queremos cortar un ciclo de repeticiones for si se cumple alguna condición usamos break.

```
for i  in range(100):
     < calculos >
     if error:
          print 'Ocurrio un error en el bucle'
          break
     < mas calculos >
```

Contadores

Variable Entera que cuenta cuantas veces ocurre una situación.

Ejemplo: Cantidad de múltiplos de 2, entre 1 y un número n.

```
n=input('Ingrese el numero ')

j=0 # Inicializo el contador
for i in range(n):
    if i % 2 == 0:
        j = j + 1 # Cada vez que ocurre la condicion le
print 'Hasta el numero ',n,'hay ',j,'multiplos de 2'
```

En python hay una forma corta para expresar el update de contadores

$j = j + 1 \rightarrow j + = 1$

Significan exactemente lo mismo.

Acumulador

Variables que acumulan resultados en forma repetitiva.

Ejemplo: Sumatoria de todos los numeros enteros menores o iguales a un número n, i.e. $S = \sum_{i=1}^{n} i$.

```
n=input('Ingrese el numero ')

sum=0 # Inicializo el acumulador
for i in range(n):
        sum = sum + i # Cada vez que ocurre la condicion
print 'Hasta el numero ',n,', la sumatoria es: ',sum
```

La operacioń que estamos realizando es igual a la del contador, y también podemos escribirla en forma pythonica como:

sum = sum + i $\rightarrow$ sum + = i

Bucles anidados. Ejemplo

Queremos que un estudiante de la primaria, Manuel mi hijo, estudie las tablas de multiplicación.

```
ierror=0
for i  in range(1,10):
    for j in range(1,10):
        cadena='Cuanto es:  '+str(i)+'x'+str(j)+' ? '
         res=input(cadena)
         if (res != i*j):
            ierror+=1
            print 'Has cometido ',ierror,' errores'
            if (ierror >= 3):
                print 'Te quedaste sin futbol.'
                break
    if  (ierror >= 3): # aqui rompo con el 2do bucle anid
        break
if  (ierror < 3):
    print 'Te felicito. Podes ir a jugar'
```