T04_Control_de_flujo

January 29, 2021

1 Python de cero a experto

Autor: Luis Miguel de la Cruz Salas

1.1 Pythonico es más bonito

1.1.1 Control de flujo

While Los número de Fibonacci, denotados con F_n forman una secuencia tal que cada número es la suma de dos números precedentes e inicia con el 0 y el 1. Matemáticamente se escribe como:

$F_0 = 0$, $F_1 = 1$ y $F_n = F_{n-1} + F_{n-2}$ para $n > 1$.

La secuencia es entonces: 0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 , 89 , 144 , ...

Vamos a calcular esta secuencia usando la instrucción `while`:

[1]:
```
# Secuencia de Fibonacci
a, b = 0, 1
while a < 145:
    print(a, end=',')
    a, b = b, a+b
```

```
0,1,1,2,3,5,8,13,21,34,55,89,144,
```

If Los siguientes condiciones lógicas son usadas en Python: - ¿son iguales?:

`a == b`

- ¿no son iguales?:

`a != b`

- ¿a es menor que b?:

`a < b`

- ¿a es menor o igual que b?:

`a <= b`

- ¿a es mayor que b?:

`a > b`

- ¿a es mayor o igual que b?:

`a >= b`

- ¿La expresión A y la expresión B son verdaderas?:

`A and B`

- ¿La expresión A o la expresión B es verdadera?:

`A or B`

[2]:
```
a = 20
b = 20
if a < b:
    print('a es menor que b')
elif a > b:
    print('a es mayor que b')
elif a == b:
    print('a es igual a b')
else:
    print('Esto nunca pasa')
```

```
a es igual a b
```

Operador ternario

[3]:
```
c = 3
r = c if c > 5 else 0
print(r)
```

```
0
```

For Permite iterar sobre el contenido de cualquier secuencia (cadena, lista, tupla, conjunto, diccionario, archivo, …)

[4]:
```
gatos = ['Persa', 'Sphynx', 'Ragdoll','Siamés']
for i in gatos:
    print(id(i))
    print(i, len(i))
    i = 'a'
```

```
139970037006448
Persa 5
139970037007280
Sphynx 6
139970037007344
Ragdoll 7
139970036938512
Siamés 6
```

[5]:
```
print(gatos)
```

```
['Persa', 'Sphynx', 'Ragdoll', 'Siamés']
```

[6]:
```
dict(zip(gatos, [1,2,3,4]))
```

[6]:
```
{'Persa': 1, 'Sphynx': 2, 'Ragdoll': 3, 'Siamés': 4}
```

[7]:
```
for i in dict(zip(gatos, [1,2,3,4])):
    print(i)
```

```
Persa
Sphynx
Ragdoll
Siamés
```

[8]:
```
for i in dict(zip(gatos, [1,2,3,4])):
    print(i)
```

```
Persa
Sphynx
Ragdoll
Siamés
```

[9]:
```
dicc = dict(zip(gatos, [1,2,3,4]))
for i in dicc:
    print(i, dicc[i])
```

```
Persa 1
Sphynx 2
Ragdoll 3
Siamés 4
```

[10]:
```
for i, v in enumerate(dicc):
    print(i, v, dicc[v])
```

```
0 Persa 1
1 Sphynx 2
2 Ragdoll 3
3 Siamés 4
```

[11]:
```
gatos.append([1,2,3,4,5])
```

[12]:
```
gatos
```

[12]:
```
['Persa', 'Sphynx', 'Ragdoll', 'Siamés', [1, 2, 3, 4, 5]]
```

[13]:
```
for i in gatos:
    print(i, type(i))
```

```
Persa <class 'str'>
Sphynx <class 'str'>
Ragdoll <class 'str'>
Siamés <class 'str'>
[1, 2, 3, 4, 5] <class 'list'>
```

Funcion range

[14]:
```
range(1,20,2) # (inicio, final, paso)
```

[14]:
```
range(1, 20, 2)
```

[15]:
```
for i in range(1,20):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

[16]:
```
r = range(1,100,10)
```

[17]:
```
print(r, type(r))
```

```
range(1, 100, 10) <class 'range'>
```

[18]:
```
for i in r:
    print(i, end=" ")
```

```
1 11 21 31 41 51 61 71 81 91
```

[19]:
```
r = list(r)
print(r, type(r))
```

```
[1, 11, 21, 31, 41, 51, 61, 71, 81, 91] <class 'list'>
```

[20]:
```
for i in range(100,1,-10):
    print(i, end=" ")
```

```
100 90 80 70 60 50 40 30 20 10
```

break, continue, else, pass

[21]:
```
for letra in "Pythonico":
    if letra == "h":
        continue
    print ("Letra actual : " + letra)
```

```
Letra actual : P
Letra actual : y
Letra actual : t
Letra actual : o
Letra actual : n
Letra actual : i
Letra actual : c
Letra actual : o
```

[22]:
```
for letra in "Pythonico":
    if letra == "h":
        break
    print ("Letra actual : " + letra)
```

```
Letra actual : P
Letra actual : y
Letra actual : t
```

[23]:
```
for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            print(n, 'igual a ', x, '*', n//x)
            break
    else:
        print(n, 'es un número primo')
```

```
2 es un número primo
3 es un número primo
4 igual a  2 * 2
5 es un número primo
6 igual a  2 * 3
7 es un número primo
```

```
8 igual a  2 * 4
9 igual a  3 * 3
```

[24]:
```
for num in range(2, 10):
    if num % 2 == 0:
        print("Número par ", num)
        continue
    print("Número impar", num)
```

```
Número par  2
Número impar 3
Número par  4
Número impar 5
Número par  6
Número impar 7
Número par  8
Número impar 9
```

[25]:
```
suma = 0
while suma < 3:
    entrada = input("Clave:")
    if entrada == "despedida":
        break
    suma = suma + 1
    print("Intento %d. \n " % suma)
print("Tuviste %d intentos fallidos." % suma)
```

```
Clave: hola
Intento 1.


Clave: mundo
Intento 2.


Clave: despedida
Tuviste 2 intentos fallidos.
```

pass Esta declaración no hace nada. Se usa principalmente para cuestiones de desarrollo de código a un nivel abstracto.

[26]:
```
i = 0
while i > 10:
    pass
```

[27]:
```
# La siguiente función calcula la secuencia de Fibonacci
def fib(n):
```

```
#    print(i, end=" ")
    pass

# En este punto del programa requiero el uso de la función fib(n):

fib(100000) #
```

[ ]:

[ ]: