

Python Mania

Artículos (RSS) Comentarios (RSS)

control de bucles, break, continue y pass

Posted by: Daniel Fuentes B. on: abril 5, 2013

In: python Dejar un comentario

voy a explicar como se usan y las diferencias entre break y continue para terminar/salir de un bucle y de paso para que sirve pass. Así que vamos con un ejemplo

Break

Este funciona tal como el break de C. Break se puede usar en bucles *for* y *while* y simplemente termina el bucle actual y continua con la ejecución de la siguiente instrucción, por ejemplo:

```
1  #!/usr/bin/python
2
3  # Primer ejemplo
4  for letra in "Python":
5      if letra == "h":
6          break
7      print "Letra actual :", letra
8
9  # Segundo ejemplo
10 var = 10
11 while var > 0:
12     var = var -1
13     if var == 5:
14         break
15     print "Valor actual de la variable :", var
16
17 print "fin del script"
```

que da como resultado:

```
1  >>>
2  Letra actual : P
3  Letra actual : y
4  Letra actual : t
5  Valor actual de la variable : 9
6  Valor actual de la variable : 8
7  Valor actual de la variable : 7
8  Valor actual de la variable : 6
```

9 | fin `del` script

Si se fijan en el primer ejemplo al llegar a la letra "h" simplemente se termina (rompe) el ciclo (bucle) y se sigue con el segundo ejemplo. En el segundo ejemplo la variable va disminuyendo su valor hasta que llega a 5, en donde se termina (rompe) el ciclo (bucle), mostrando el print final.

Continue

Al aparecer un Continue en Python, este regresa al comienzo del bucle, ignorando todos los estamentos que quedan en la iteración actual del bucle e inicia la siguiente iteración. Queda más claro con un ejemplo:

Nota: al igual que break, se puede usar en los bucles *for* y *while*

```
1  #!/usr/bin/python
2
3  # Primer ejemplo
4  for letra in "Python":
5      if letra == "h":
6          continue
7      print "Letra actual :", letra
8
9  # Segundo ejemplo
10 var = 10
11 while var > 0:
12     var = var -1
13     if var == 5:
14         continue
15     print "Valor actual de la variable :", var
16
17 print "fin del script"
```

que da como resultado:

```
1  >>>
2  Letra actual : P
3  Letra actual : y
4  Letra actual : t
5  Letra actual : o
6  Letra actual : n
7  Valor actual de la variable : 9
8  Valor actual de la variable : 8
9  Valor actual de la variable : 7
10 Valor actual de la variable : 6
11 Valor actual de la variable : 4
12 Valor actual de la variable : 3
13 Valor actual de la variable : 2
14 Valor actual de la variable : 1
15 Valor actual de la variable : 0
16 fin del script
```

En el primer ejemplo al llegar a la letra "h" simplemente termina esa iteración (ignorando al print que sigue en la línea 7) y continua con las siguientes iteraciones (letras o y n) hasta que se termina el ciclo (bucle).

En el segundo ejemplo la variable va disminuyendo su valor hasta que llega a 5, en donde se termina esa iteración del ciclo (bucle) y se continua con las iteraciones que siguen hasta que se termina el bucle y se llega al print final.

Pass

Por ultimo tenemos a pass, que tal como su nombre lo indica es una operación nula, o sea que no pasa nada cuando se ejecuta. Se utiliza cuando se requiere por sintaxis una declaración pero no se quiere ejecutar ningún comando o código. También se utiliza en lugares donde donde el código irá finalmente, pero no ha sido escrita todavía (utilizándolo como un relleno temporal, hasta que se escriba el código final).

Vamos con el ejemplo:

```
1  #!/usr/bin/python
2
3  # Primer ejemplo
4  for letra in "Python":
5      if letra == "h":
6          pass
7      print "Letra actual :", letra
8
9  # Segundo ejemplo
10 var = 10
11 while var > 0:
12     var = var -1
13     if var == 5:
14         pass
15     print "Valor actual de la variable :", var
16
17 print "fin del script"
```

que da como resultado:

```
1  >>>
2  Letra actual : P
3  Letra actual : y
4  Letra actual : t
5  Letra actual : h
6  Letra actual : o
7  Letra actual : n
8  Valor actual de la variable : 9
9  Valor actual de la variable : 8
10 Valor actual de la variable : 7
11 Valor actual de la variable : 6
12 Valor actual de la variable : 5
13 Valor actual de la variable : 4
14 Valor actual de la variable : 3
15 Valor actual de la variable : 2
16 Valor actual de la variable : 1
17 Valor actual de la variable : 0
18 fin del script
```

En el primer ejemplo al llegar a la letra h no se ejecuta nada (se pasa), siguiendo con la ejecución de la siguiente línea (la línea 7). Lo mismo ocurre con el 5 en el segundo ejemplo

Nota: pass es útil cuando se ha creado un bloque de código, pero ya no es necesario, por lo que se pueden eliminar las instrucciones dentro del bloque remplazandolas a todas con una única sentencia pass, de modo que se mantiene el bloque dentro del programa, pero como no hace nada, no interfiere con otras partes del código.

Diferencia entre continue y pass

La diferencia es que tal como lo indica su nombre continue termina la iteración actual, pero continua con el ciclo, volviendo al inicio del bucle en la siguiente iteración. En cambio pass simplemente no hace nada y pasa a la siguiente instrucción.

Por ejemplo si abrimos el interprete:

```
1  >>> for x in (1, 2, 3):
2      print x
3      continue
4      print x, "nuevamente"
5
6  1
7  2
8  3
9  >>> for x in (1, 2, 3):
10     print x
11     pass
12     print x, "nuevamente"
13
14  1
15  1 nuevamente
16  2
17  2 nuevamente
18  3
19  3 nuevamente
```

En el primer caso (el del continue): imprime (print) el valor de x que es 1, llega al continue, actualiza x al valor de 2, imprime el valor de x que que es 2, llega al continue, actualiza x al valor de 3, imprime el valor de x que que es 3, llega al continue, y termina el ciclo ya que no hay más iteraciones (valores) para x.

En el segundo caso (el del pass): imprime (print) el valor de x que es 1, llega al pass (y pasa a la siguiente línea), imprime 1 nuevamente, actualiza x al valor de 2, imprime el valor de x que que es 2, llega al pass (y pasa a la siguiente línea), imprime 2 nuevamente, , actualiza x al valor de 3, imprime el valor de x que que es 3, llega al pass (y pasa a la siguiente línea), imprime 3 nuevamente, y termina el ciclo ya que no hay más iteraciones (valores) para x.

Buscar

[Crea un blog o un sitio web gratuitos con WordPress.com.](#)

[El tema Albeo.](#)