

① Syntax of the break statement and continue statement

(A) Both break and continue are keywords as well as statements.

(B) Both of these statements must be used in the body of iterative statements.

If you try to use these statements outside the body of for/while then the Python will emit a Syntax error.

Experiment:

```
>>> break
```

SyntaxError.

```
>>> continue
```

SyntaxError.

(C) Even inside the body of iterative statements, 'break' and 'continue' are not used unconditionally.

<pre>i=0 while i<5: break i=i+1</pre> <p>X</p>	<pre>i=0 while i<5: continue. i=i+1</pre> <p>X</p>
<pre>L=[10,20,30,40] for x in L: break</pre> <p>X</p>	<pre>L=[10,20,30,40] for x in L: continue</pre> <p>X</p>

2) Semantics of break and continue statements:

(i) Use-case (ii) Run-time behaviour.

Destined Number of Iterations.

(= Number of repetitions a while/for statement is supposed to make.

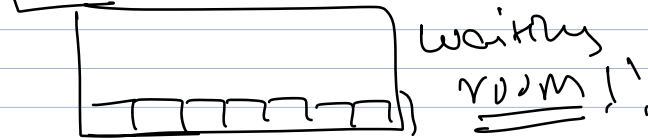
while \rightarrow initialization / condition / modification

for \rightarrow length of iterable in the header

).

exam room

Analogy



for patient on chairs:
examine (patient)

for/while :

— (break)

$i = 0$

Exp-1.

while $i < 5$:

print('START')

break

print('END')

$i = i + 1$

Exp - II.

L = [10, 20, 30, 40]

for x in L:

print('START', x)

break

print('END')

for/while:

Evaluate Cond

break