

Conditional control Statements:

In C-Language: goto | break | continue

In Java : x | break | continue

In Python: x | break | continue | pass

Pass is keyword it used to represent an empty block | scope | suite

- As the name suggests pass statement simply does nothing. The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute. It is like `null` operation, as nothing will happen is it is executed. `Pass` statement can also be used for writing empty loops. `Pass` is also used for empty control statement, function and classes.

In Python:

```
a,b=10,20
```

```
if a>b:
```

```
    pass
```

```
else:
```

```
    pass
```

```
def add(x,y):
```

```
    pass
```

```
def sub(x,y):
```

```
    pass
```

```
    print("Chinni")
```

Ex 2:

```
import time
```

```
for i in range(1,11):  
    time.sleep(.2)  
    if i==3 or i==7:  
        continue  
    print(i)
```

Ex 3:

```
import time
```

```
s="education"  
for i in s:  
    time.sleep(.1)  
    if i=='a' or i=='e' or i=='i' or i=='o' or i=='u':  
        continue  
    print(i)
```

Break

- The `break` statement is used to terminate the loop or statement in which it is present. After that, the control will pass to the statements that are present after the `break` statement, if available. If the `break` statement is present in the nested loop, then it terminates only those loops which contains `break` statement.
- `is` keyword it is used to comes out from the control statement [`while` | `for ...`] based certain condition

```
import time
```

```
for i in range(1,11):  
    time.sleep(.5)  
    if i==5:  
  
        break
```

```
print(i)
```

Continue: it is used to skip the current iteration

- Continue is also a loop control statement just like the break statement. continue statement is opposite to that of break statement, instead of terminating the loop, it forces to execute the next iteration of the loop.

- As the name suggests the continue statement forces the loop to continue or execute the next iteration. When the continue statement is executed in the loop, the code inside the loop following the continue statement will be skipped and the next iteration of the loop will begin.

```
import time
```

```
for i in range(1,11):  
    time.sleep(.2)  
    print("Hello")  
    print(i)  
    continue
```