

# Jump Statements

- 
- JUMP STATEMENTS ARE USED TO TRANSFER THE PROGRAM'S CONTROL FROM ONE LOCATION TO ANOTHER.
  - MEANS THESE ARE USED TO ALTER THE FLOW OF A LOOP LIKE - TO SKIP A PART OF A LOOP OR TERMINATE A LOOP

# Types of Jump Statements

---

There are three types of jump statements used in python.

1.break

2.continue

3.pass

# break statement

---

It is used to terminate the loop

e.g.

```
for val in "string":  
    if val == "i":  
        break  
    print(val)  
print("The end")
```

Output

s

t

r

The end

# continue statement

It is used to skip all the remaining statements in the loop and move controls back to the top of the loop.

e.g.

```
for val in "string":  
    if val == "i":  
        continue  
    print(val)  
print("The end")
```

Output

s  
t  
r  
n  
g  
The end

# pass statement

- This statement does nothing.
  - It can be used when a statement is required syntactically but the program requires no action.
- 

e.g.

```
for val in "string":  
    if val == "i":  
        pass  
    print(val)  
print("The end")
```

Output

s  
t  
r  
i  
n  
g  
The end

# pass statement

e.g.

```
for i in 'initial':  
    if(i == 'i'):  
        continue  
    else:  
        print(i)
```

OUTPUT  
n  
t  
a  
l

e.g.

```
for i in 'initial':  
    if(i == 'i'):  
        pass  
    else:  
        print(i)
```

OUTPUT  
i  
n  
i  
t  
i  
a  
l

**NOTE :** continue forces the loop to start at the next iteration while pass means "there is no code to execute here" and will continue through the remainder or the loop body.

# pass statement

---

## Use in loop

while True:

```
    pass                # Busy-wait for keyboard interrupt (Ctrl+C)
```

## In function

It makes a controller to pass by without executing any code.

e.g.

```
def myfun():
```

```
    pass    #if we don't use pass here then error message will be shown
```

```
print('my program')
```