



Control Flow in Programming: If-Else, Loops, and Control Statements

This document provides an overview of fundamental control flow structures in programming, specifically focusing on if-else statements, for and while loops, as well as break and continue statements. These constructs are essential for creating dynamic and responsive code, allowing developers to implement decision-making processes and repetitive tasks efficiently.

If-Else Statements

The if-else statement is a conditional control structure that allows a program to execute certain blocks of code based on whether a specified condition evaluates to true or false. The basic syntax is as follows:

```
if condition:
    # code to execute if condition is true
else:
    # code to execute if condition is false
```

Example:

```
age = 18
if age >= 18:
    print("You are an adult.")
else:
    print("You are a minor.")
```

In this example, the program checks if the variable **age** is greater than or equal to 18. If true, it prints "You are an adult." Otherwise, it prints "You are a minor."

For Loops

A for loop is used to iterate over a sequence (like a list, tuple, or string) or a range of numbers. It allows you to execute a block of code multiple times. The syntax is:

```
for variable in sequence:
    # code to execute for each item in the sequence
```

Example:

```
for i in range(5):  
    print(i)
```

This loop will print the numbers 0 through 4, as it iterates over the range generated by **range(5)**.

While Loops

A while loop repeatedly executes a block of code as long as a specified condition is true. The syntax is:

```
while condition:  
    # code to execute while condition is true
```

Example:

```
count = 0  
while count < 5:  
    print(count)  
    count += 1
```

In this example, the loop continues to print the value of **count** until it reaches 5, incrementing **count** by 1 on each iteration.

Break and Continue Statements

The **break** and **continue** statements are used to alter the flow of loops.

Break Statement

The **break** statement is used to exit a loop prematurely. It can be used in both for and while loops.

Example:

```
for i in range(10):  
    if i == 5:  
        break  
    print(i)
```

This loop will print numbers 0 through 4 and then exit when **i** equals 5.

Continue Statement

The **continue** statement skips the current iteration of a loop and proceeds to the next iteration.

Example:

```
for i in range(5):  
    if i == 2:  
        continue  
    print(i)
```

In this case, the loop will print 0, 1, 3, and 4, skipping the number 2.

Conclusion

Understanding control flow structures such as if-else statements, loops, and control statements like break and continue is crucial for effective programming. These constructs enable developers to create flexible and efficient code that can handle various conditions and repetitive tasks seamlessly. Mastery of these concepts lays the foundation for more advanced programming techniques and problem-solving skills.