

# Key Concepts on Control Flow in Python

## What is Control Flow?

- Refers to the order in which a program's instructions are executed.
- Enables decision-making, leading to different program actions or directions.
- In Python, control flow is managed through:
  - **Conditional Statements:** `if`, `else`, `elif`.
  - **Loops:** `for` and `while`.

## Conditional Statements Overview

### 1. `if` Statement:

- Executes a block of code if the condition is true.
- Example:

```
if bill_total > 100:  
    print("Bill is greater than 100")
```

### 2. `else` Statement:

- Executes when the preceding `if` condition is false.
- Example:

```
else:  
    print("Bill is less than or equal to 100")
```

### 3. `elif` Statement:

- Short for "else if".
- Checks another condition if the preceding `if` or `elif` conditions are false.
- Example:

```
elif bill_total > 200:  
    print("Bill is greater than 200")
```

## Practical Example: Restaurant Discount Application

### 1. Setting Up Variables:

- Define `bill_total` and discount variables (`discount1`, `discount2`).

- Example:

```
bill_total = 210
discount1 = 10
discount2 = 20
```

## 2. Adding Conditions:

- Apply a discount if the bill is over a specific amount:
  - If `bill_total > 100` and `bill_total < 200`, subtract `discount1`.
  - If `bill_total > 200`, subtract `discount2`.

## 3. Final Code:

```
bill_total = 210
discount1 = 10
discount2 = 20

if bill_total > 100 and bill_total < 200:
    print("Bill is greater than 100")
    bill_total = bill_total - discount1
elif bill_total > 200:
    print("Bill is greater than 200")
    bill_total = bill_total - discount2
else:
    print("Bill is less than 100")

print("Total bill is " + str(bill_total))
```

## 4. Outputs:

- For `bill_total = 210`:

```
Bill is greater than 200
Total bill is 190
```

## Important Points

- **Order Matters:**
  - Conditions are evaluated in the sequence they appear.
  - Only the first true condition executes, skipping others.
- **else Acts as a Catch-All:**
  - Executes only if none of the preceding `if` or `elif` conditions are true.

- **Typecasting:**
  - Convert data types (e.g., integer to string) when combining text and numbers in outputs.

## Loops in Python (Brief Introduction)

1. **for Loop:**
  - Repeats code for a known number of iterations.
2. **while Loop:**
  - Repeats code while a condition is true.

## Key Takeaways

- Mastering control flow is essential for writing effective programs.
- Conditional statements enable decision-making in code.
- Practice by applying these concepts to solve real-world problems.