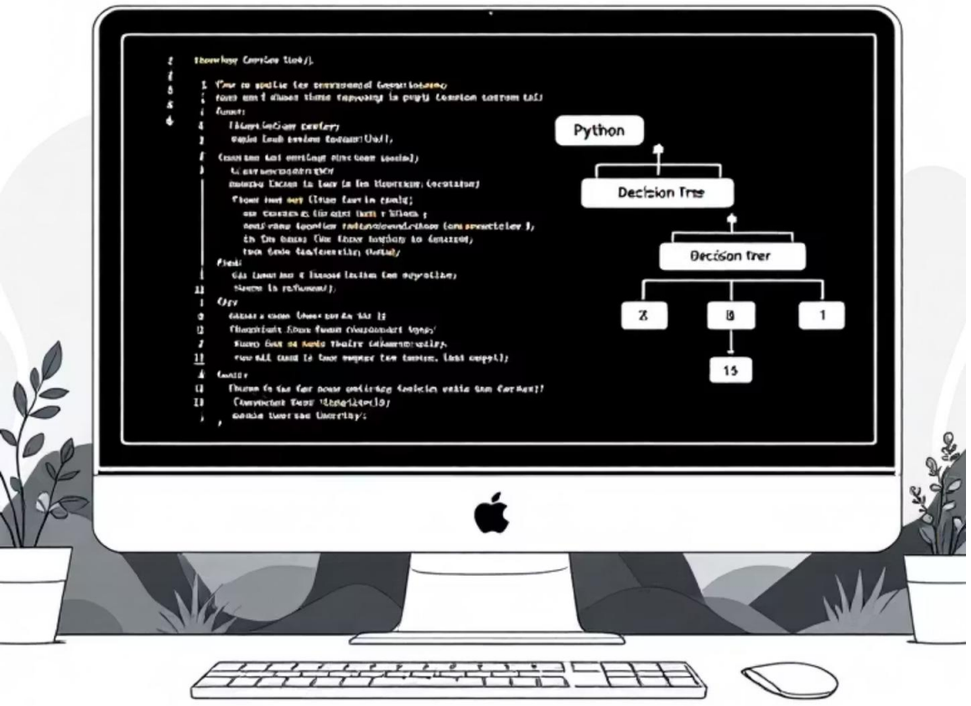

Constraints and Conditional Statements in Python – Complete Guide

A comprehensive guide to understanding how constraints work as rules in programming and how conditional statements enforce them to create safe, correct, and predictable programs.

MATH CODE LAB

Constraints First!

What are Constraints?

*Constraints are **rules or limits** that tell us what values are acceptable in a system.*

Real Life

"You must be at least 16 to ride this roller coaster."

Math

"x must be between 1 and 100."

Programming

"Password must be at least 8 characters."

They describe the boundaries within which everything works correctly.

Why are Constraints Important?

Constraints help keep our programs:

Safe

They don't allow wrong inputs

Correct

Answers follow the rules

Fair and Predictable

Everyone plays by the same rules

Without constraints:

- *An age of -5 years doesn't make sense.*
- *A score of 200 in a game with max 100 would break fairness.*
- *An empty password would not keep accounts safe.*

Just like classroom rules keep order, ***constraints keep programs running smoothly****.*

How Do We Enforce Constraints?

*This is where **Conditional Statements** step in. They are like **gatekeepers** who check:*

- *"Does this input follow the rules?"*
- *"If yes, what should we do? If no, what's the correct response?"*

Example:

```
age = int(input("Enter your age: "))if age < 0:    print("Age cannot be negative.")  # Enforcing constraintelif age < 18:    print("You are not eligible to vote yet.")else:    print("You can vote!")
```

Here:

- ***Constraint:** age >= 0*
- ***Conditional Statement:** if age < 0 checks and blocks invalid input*
- ***Flow:** Program continues only when the value follows the rules*

> 👉 *In short: Constraints are the rules. Conditional statements are the tools to enforce them.*

Conditional Statements – The Enforcers

Computers are patient helpers waiting for instructions. Conditional statements are their "decision goggles" that enforce your constraints by asking questions and choosing the correct path.

Single if (Optional Action)

Ask ONE question; maybe run ONE block.

Real life: *"Is it raining?" If yes → grab umbrella; otherwise continue.*

Example: Helmet Safety

```
going_to_ride_bike = True  # Change to False to testif going_to_ride_bike:    print("Put on
helmet")print("Ready to go")
```

If condition is False the inside block is skipped.

More Quick Examples

```
raining = Trueif raining:    print("Take
umbrella")print("Leave house")battery_low =
Falseif battery_low:    print("Plug in
charger")print("Continue using device")
```

Input Version

```
answer = input("Did you finish your homework?
(yes/no): ")if answer == "yes":    print("Great
job! Free time now.")print("Homework check
complete.")
```

Try editing the prompt for other habits.

Practice – Single If

1. *Ask: "Did you sleep well?" If yes print "Energy high!".*
2. *Ask: "Hungry?" If yes print a snack idea.*
3. *Ask: "Water bottle full?" If no (answer != "yes") print "Fill it".*
4. *Ask: "Backpack packed?" If no print "Pack it now".*

if / else (Exactly One of Two Paths)

One question. ALWAYS take exactly one branch.

Pattern:

```
if condition:    # then branchelse:    # else branch
```

Example: Traffic Light

```
color = input("Light color (red/green): ")if color == "green":    print("Go")else:    print("Stop")
```

More Examples

```
done = input("Chores finished? (yes/no): ")if done == "yes":    print("Snack time!")else:
print("Do chores first")
```

```
cold = input("Is it cold? (yes/no): ")if cold == "yes":    print("Wear jacket")else:
print("No jacket needed")
```

Practice – If / Else

1. *Battery low? yes -> Charge now / else -> Keep using.*
2. *Friend online? yes -> Start chat / else -> Wait.*
3. *Shoes on? no -> Put on shoes / else -> Ready.*

if / elif / else (Choose ONE of Many)

Top-to-bottom scan. First True branch runs; rest are skipped. Optional final else for "none matched".

Shape:

```
if cond1:    ...elif cond2:    ...elif cond3:    ...else:    ...  # (only if nothing above matched)
```

Example: Weather Outfit

```
weather = input("Weather (sunny/cloudy/rainy): ")if weather == "sunny": print("Wear sunglasses")elif weather == "cloudy": print("Take a light jacket")elif weather == "rainy": print("Grab an umbrella")else: print("Check the forecast")
```

match (Structural / Exact Pattern Match) – Python 3.10+

Modern alternative for checking ONE value against many patterns. Cleaner than large if/elif chains when comparing the same subject.

Shape:

```
match value:    case pattern1:        ...    case pattern2:        ...    case _:        ...  # fallback
```

_ is a wildcard (anything else). You can group with | and add guards if ... for ranges.

Example: Day Greeting

```
day = "mon" # try: mon / tue / wed / fri / sat / sunmatch day: case "mon": print("Monday boost: You got this!") case "tue": print("Tuesday: Keep rolling") case "wed": print("Mid-week high five") case "fri": print("Friday: Almost weekend!") case "sat" | "sun": print("Weekend mode: Relax") case _: print("Use a short day code (mon/tue/wed/fri/sat/sun)")
```

Quick Decision Pattern Cheat Sheet

Need	*Use*
Enforce or check a constraint	*if or if / else*
Maybe do something	*single if*
Exactly one of two	*if / else*
One of many (first true)	*if / elif / else*
One of many (same subject)	*match*

Real-Life Rule

Finish homework before play

Python Constraint

```
if homework_done:    print("Play time!")else:
print("Finish homework first")
```