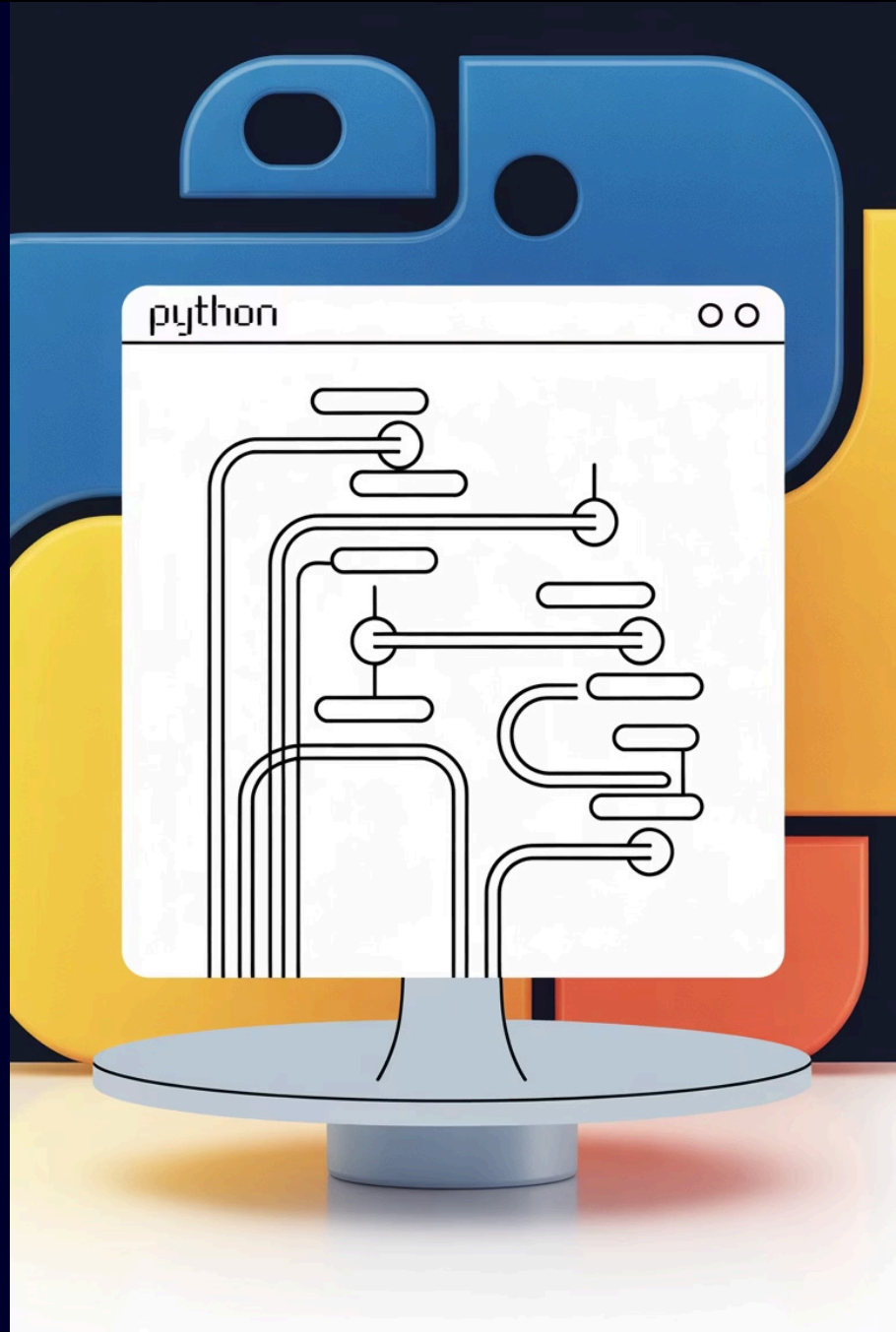
Constraints and Conditional Statements in Python

A complete guide to understanding how constraints work in Python and the various conditional statements you can use to enforce them.



Constraints First!

What are Constraints?

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

- In real life: "You must be at least 16 to ride this roller coaster."
- In math: "x must be between 1 and 100."
- In programming: "Password must be at least 8 characters."

They describe the boundaries within which everything works correctly.

Why are Constraints Important?

Safety

They don't allow wrong inputs that could cause errors or security issues.

Correctness

Answers follow the rules and produce expected results.

Fairness

Everyone plays by the same rules, making programs predictable.

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?"

```
age = int(input("Enter your age: "))
if age < 0:
    print("Age cannot be negative.")  # Enforcing constraint
elif 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.

Types of Conditional Statements in Python:

Single if

Optional action

if / else

Exactly one of two paths

if / elif / else

Choose one of many

match

Pattern/structural match

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 test
if 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 = True
if raining:
    print("Take umbrella")
print("Leave house")

battery_low = False
if 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.")
```

if / else (Exactly One of Two Paths)

One question. ALWAYS take exactly one branch.

Pattern:

```
if condition:
    # then branch
else:
    # 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")
```

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")
```

More if/elif/else Examples

Time Management

```
time = input("Time (homework/brush/bed): ")
if time == "homework":
    print("Finish your assignments")
elif time == "brush":
    print("Brush your teeth")
elif time == "bed":
    print("Lights out. Sleep well")
else:
    print("Not in the plan")
```

Meal Selection

```
meal = input("Meal (sandwich/pasta/salad): ")
if meal == "sandwich":
    print("Packing sandwich")
elif meal == "pasta":
    print("Serving pasta")
elif meal == "salad":
    print("Fresh salad ready")
else:
    print("Pick something from the list")
```

Practice Ideas – Elif Chains

After-School Plan: homework/snack/play -> custom line; else -> Pick one plan.

Recess Activity: tag/swings/read -> action line; else -> Ask a teacher.

Favorite Subject: math/art/science -> fun title; else -> Great subject!

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 / sun
match 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")
```

Real-Life Rules vs Python Constraints

Using constraints in Python is similar to following rules in real life.

Real-Life Rule	Python Constraint Example

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

Only drink water if bottle is full	`if bottle_full:`
	`  print("Drink water")`
	`else:`
	`  print("Fill the bottle first")`

Ride allowed only if height ⩾ 120 cm	`if height >= 120:`
	`  print("You can ride!")`
	`else:`
	`  print("Too short for this ride")`

More Real-Life Rules vs Python Constraints

Real-Life Rule	Python Constraint Example

Password must be at least 8 characters	`if len(password) >= 8:`
	`  print("Password accepted")`
	`else:`
	`  print("Password too short")`

Game score cannot be negative	`if score >= 0:`
	`  print("Score recorded")`
	`else:`
	`  print("Invalid score")`

In both real life and programming, constraints help maintain order and ensure things work as expected.

Quick Decision Pattern Cheat Sheet

Enforce or check a constraint

Use: if or if / else

Maybe do something

Use: single if

Exactly one of two

Use: if / else

One of many (first true)

Use: if / elif / else

One of many (same subject)

Use: match