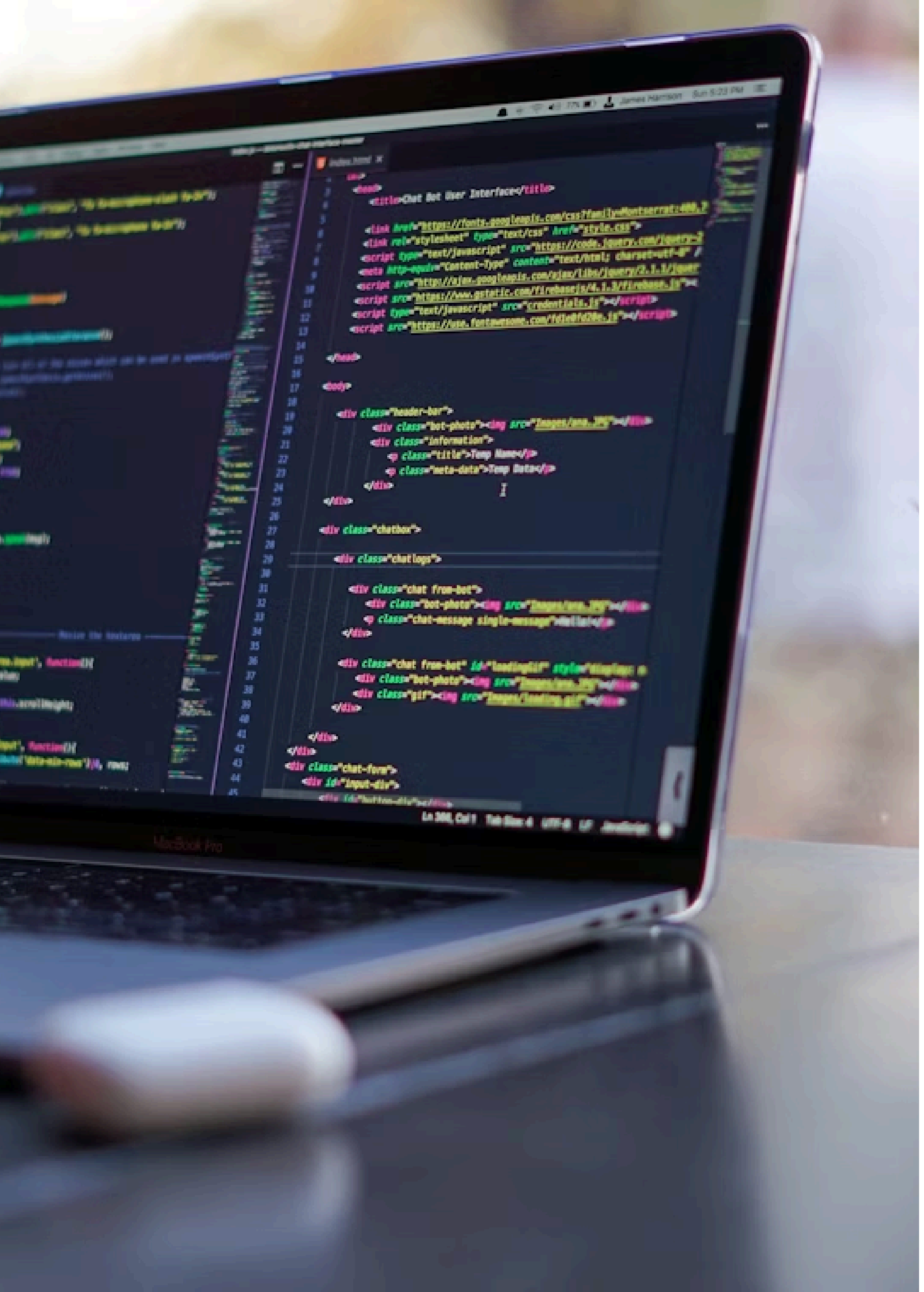

# Python Control Flow:

## Mastering If Statements

by Xintong

## What is an If Statement?

- Fundamental building block of programming logic 🔨
- Allows conditional execution of code blocks
- Evaluates boolean expressions (True/False)
- Controls program flow based on conditions

**Real World Analogy:**

> "If it rains, take an umbrella ☔; else, wear sunglasses 😎"

## Basic Syntax Structure

```
if condition:
    # code to execute
    # if condition is True
```

- **Colon (:)**: Required at end of condition
- **Indentation**: 4 spaces for code block
- **Condition**: Any expression that returns boolean

## Simple Example: Age Check

```python
age = 18

if age >= 18:
    print("You are an adult")
    print("You can vote!")

print("This always executes")
```

**Key Points:**

✅ Code block executes only if condition is True

✅ Multiple lines allowed in block

✅ Code after block always runs

## Adding Alternatives

### The else Clause

```python
if condition:
    # True block
else:
    # False block
```

### Example:

```python
temperature = 30

if temperature > 25:
    print("It's hot! 🥵")
else:
    print("It's cool 😎")
```

## Handling Multiple Conditions

### elif (Else If) Ladder

```
grade = 85

if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Needs improvement")
```

- Evaluates top to bottom
- Stops at first true condition
- else is optional

## Nested If Statements

```python
account_active = True
balance = 150

if account_active:
    if balance >= 100:
        print("Withdrawal allowed")
    else:
        print("Insufficient funds")
else:
    print("Account disabled")
```

**Best Practice:**

❗ Avoid deep nesting (hard to read)

✅ Use logical operators (and/or) when possible

## Common Errors & Pitfalls

### 1. Missing Colon

```
if x > 5  # SyntaxError
```

### 2. Incorrect Indentation

```
if correct:
print("Wrong")  # IndentationError
```

### 3. Assignment vs Comparison

```
if x = 5:  # SyntaxError
```

## Other Truthy and Falsy Values

| Truthy | Falsy |
|---|---|
| Non-zero numbers | 0 |
| Non-empty strings | "" |
| Non-empty collections | None |

### Example:

```python
name = ""

if name:
    print(f"Hello {name}")
else:
    print("Anonymous user")
```

## Best Practices

1. Keep conditions simple
2. Avoid complex nested structures
3. Use parentheses for clarity in complex conditions

```python
if (x > 5) and (y < 10):
    # ...
```

4. Use meaningful variable names
5. Comment complex logic

## Summary

- `if` for basic conditions
- `elif` for multiple branches
- `else` for final alternative
- Indentation defines code blocks
- Use boolean logic effectively
- Watch for common syntax errors

## Practice Exercises

1. Basic: Check if number is positive/negative/zero
2. Intermediate: Create grade classifier (A-F)
3. Advanced: Implement login system with:
   - Username check
   - Password length
   - Admin privileges