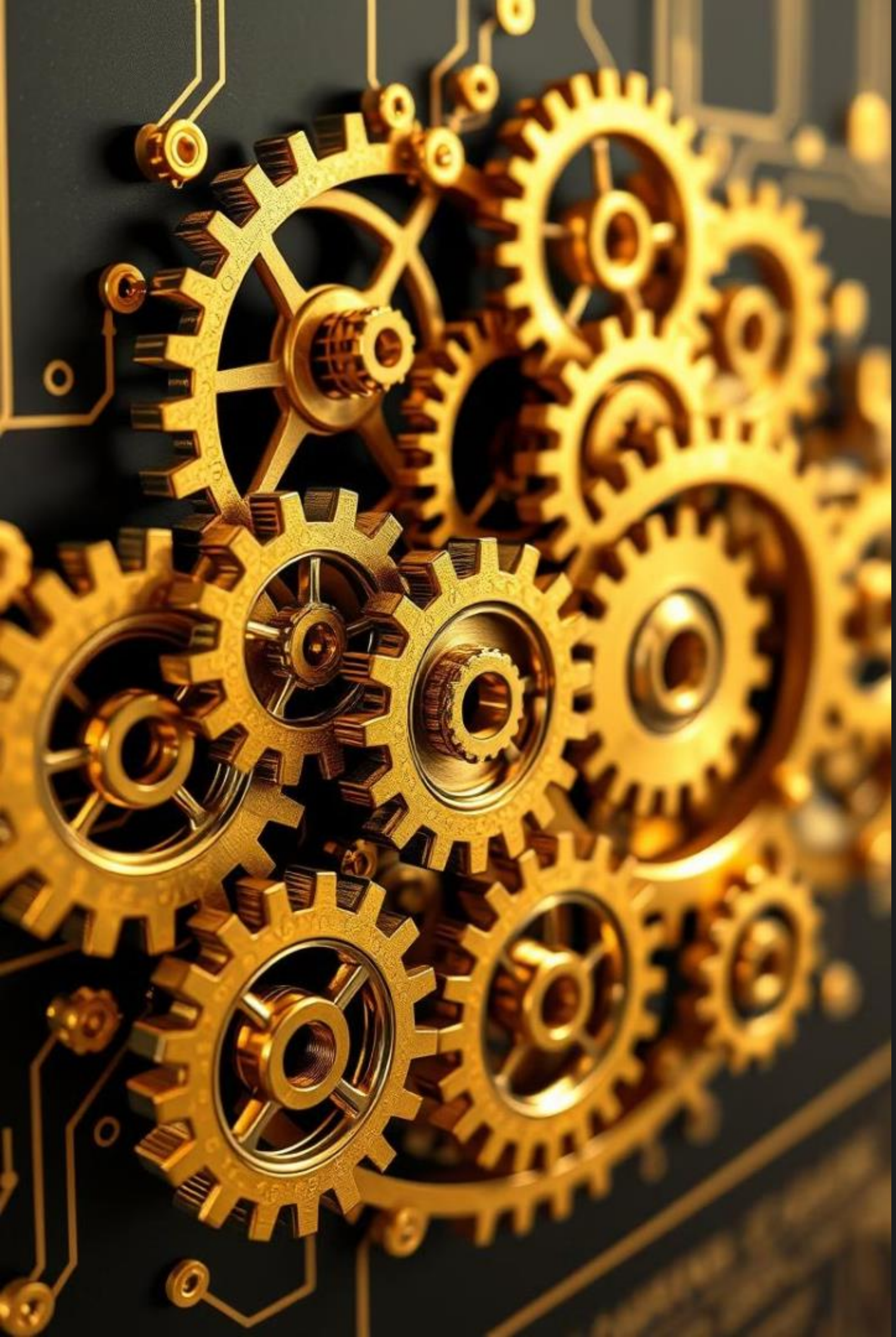
Conditional Statements in Python

This presentation explores the fundamental conditional statements in Python, which allow your code to make decisions based on various conditions.



Key Takeaways

1 Conditional Logic

Conditional statements are essential for creating dynamic and responsive programs.

2 Code Flow Control

They allow you to control the flow of your code based on specific conditions.

3 Decision-Making

They enable your programs to make decisions and respond to different inputs.

The if Statement

Syntax

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

Example

```
number = 10
if number > 5:
  print("The number is greater than 5")
```

The if-else Statement

Syntax

```
if condition:
  # Code to execute if condition is True
else:
  # Code to execute if condition is False
```

Example

```
number = 3
if number > 5:
  print("The number is greater than 5")
else:
  print("The number is less than or equal to 5")
```

The elif Statement

Syntax

```
if condition1:
  # Code to execute if condition1 is True
elif condition2:
  # Code to execute if condition2 is True
else:
  # Code to execute if all conditions are False
```

Example

```
number = 15
if number < 10:
  print("The number is less than 10")
elif number < 20:
  print("The number is between 10 and 20")
else:
  print("The number is 20 or greater")
```

Nested if-else Statements

Syntax

```
if condition1:
  if condition2:
    # Code to execute if condition1 and
condition2 are True
  else:
    # Code to execute if condition1 is True but
condition2 is False
else:
  # Code to execute if condition1 is False
```

Example

```
number = 7
if number > 5:
  if number % 2 == 0:
    print("The number is greater than 5 and even")
  else:
    print("The number is greater than 5 and odd")
else:
  print("The number is 5 or less")
```