Tutorial 2: Python Control Flow - Making Decisions in Code! 🎮

Hey future programmers! Ready to make your programs smarter? Just like you make decisions in your daily life (like what to wear or what to eat), your code needs to make decisions too! In this tutorial, you'll learn how to make your programs think and act based on different situations.

🎯 What You'll Learn

- How to make your code make decisions using if-statements
- How to repeat actions using loops (like in games where you have multiple lives!)
- How to compare values (like checking if you have enough coins to buy something)
- How to write simple functions (like creating your own commands)
- How to build small games and programs!

1. Making Decisions with If-Statements 🤔

Think of if-statements like questions you ask yourself:

- IF it's raining, take an umbrella
- IF you have enough money, buy ice cream
- ELSE save the money

Basic If-Statements

```
# Check if a player has enough coins to buy an item
player_coins = 100
item_cost = 50

# This is like asking: "Do I have enough money?"
if player_coins >= item_cost:
    print("You can buy this item!")
    player_coins = player_coins - item_cost  # Subtract the cost
    print(f"You have {player_coins} coins left")
```

If-Else Statements

When you need to do something in both cases:

```
# Check if a student passed or failed
score = 75
passing_score = 70

# This is like saying: "If you got 70 or higher you pass,
# otherwise you need to try again"
if score >= passing_score:
    print("You passed! 🎉")
```

```
    print(f"Your score was {score - passing_score} points above passing!")
else:
    print("Try again next time! 📚")
    print(f"You need {passing_score - score} more points to pass")
```

If-Elif-Else Statements

```
# Grade calculator
grade = 85

if grade >= 90:
    print("A - Excellent! 🌟")
elif grade >= 80:
    print("B - Good job! ⭐")
elif grade >= 70:
    print("C - Keep practicing! 📝")
else:
    print("Need more practice! 💪")
```

Mini-Project 1: Simple Game Choice

```
print("You find a mysterious door. What do you do?")
print("1. Open the door")
print("2. Knock first")
print("3. Walk away")

choice = input("Enter your choice (1/2/3): ")

if choice == "1":
    print("The door creaks open...")
elif choice == "2":
    print("A friendly voice answers...")
elif choice == "3":
    print("You decide to play it safe...")
else:
    print("Invalid choice!")
```

2. Comparison Operators 🔍

Let's learn how to compare things:

```
# Equal to: ==
# Not equal to: !=
# Greater than: >
# Less than: <
# Greater than or equal to: >=
# Less than or equal to: <=
```

```
age = 15
if age >= 13:
    print("You can play teen-rated games!")
```

3. Repeating Actions with While Loops 🔄

Basic While Loop

```
# Countdown timer
countdown = 5

while countdown > 0:
    print(f"{countdown}...")
    countdown = countdown - 1
print("Blast off! 🚀")
```

While Loop with Break

```
# Guess the number game
secret_number = 7
attempts = 0

while True:
    guess = int(input("Guess the number (1-10): "))
    attempts = attempts + 1

    if guess == secret_number:
        print(f"You got it in {attempts} tries! 🎉")
        break
    elif guess > secret_number:
        print("Too high! Try again!")
    else:
        print("Too low! Try again!")
```

4. For Loops - Repeating with a Pattern 🔁

Basic For Loop

```
# Print multiplication table
number = 5
for i in range(1, 11):
    print(f"{number} x {i} = {number * i}")
```

Looping Through Lists

```
# Check your inventory
inventory = ["sword", "shield", "potion", "map"]
for item in inventory:
    print(f"You have a {item}")
```

Using Range

```
# Different ways to use range
for i in range(5):  # 0 to 4
    print(i)

for i in range(2, 5):  # 2 to 4
    print(i)

for i in range(0, 10, 2):  # Even numbers 0 to 8
    print(i)
```

5. Break and Continue 🚦

Break - Stop the Loop

```
# Find the first multiple of 7
for i in range(1, 101):
    if i % 7 == 0:
        print(f"First multiple of 7 is: {i}")
        break
```

Continue - Skip to Next Iteration

```
# Print only even numbers
for i in range(1, 6):
    if i % 2 != 0:
        continue
    print(f"{i} is even")
```

6. Introduction to Functions 🔧

Basic Functions

```
def greet_player(name):
    print(f"Welcome to the game, {name}! 🎮")
```

```
# Use the function
greet_player("Alex")
```

Functions with Return Values

```
def calculate_score(hits, misses):
    score = (hits * 100) - (misses * 50)
    return score

# Use the function
player_score = calculate_score(5, 2)
print(f"Your score: {player_score}")
```

🎮 Fun Projects to Try!

Project 1: Number Guessing Game 🎲

This game uses random numbers and loops to create a fun guessing game. It's like "Hot and Cold" but with numbers!

```
import random  # This lets us create random numbers

def play_guess_game():
    # Create a random number between 1 and 100
    secret = random.randint(1, 100)
    attempts = 0
    max_attempts = 10  # Let's give the player 10 tries

    print("🎲 Welcome to the Number Guessing Game! 🎲")
    print("I'm thinking of a number between 1 and 100...")
    print(f"Can you guess it in {max_attempts} attempts?\n")

    while attempts < max_attempts:
        # Get the player's guess
        try:
            guess = int(input("Your guess: "))
            attempts += 1

            # Check if the guess is correct
            if guess == secret:
                print(f"🎉 You won in {attempts} tries! 🎉")
                if attempts == 1:
                    print("Wow! First try! You're amazing! 🌟")
                elif attempts <= 3:
                    print("Great job! You're really good at this! ⭐")
                else:
                    print("Well done! Keep practicing to get better! 👍")
                break
            # Give hints if the guess is wrong
            elif guess > secret:
```

```
                print("Too high! Try a lower number 👇")
            else:
                print("Too low! Try a higher number 👆")

            # Tell player how many attempts are left
            attempts_left = max_attempts - attempts
            print(f"Attempts left: {attempts_left}\n")

        except ValueError:
            print("Please enter a valid number! 🔢")

    # If player runs out of attempts
    if attempts >= max_attempts and guess != secret:
        print(f"Game Over! The number was {secret}")
        print("Better luck next time! 🎮")

# Play the game
play_guess_game()
```

Project 2: Dragon's Cave Adventure Game 🐲

This is a text-based adventure game where every choice matters! It's like creating your own story where the player decides what happens next.

```
def dragons_cave():
    # Initialize player stats
    health = 100
    coins = 0
    has_sword = False
    has_key = False

    print("🐲 Welcome to the Dragon's Cave! 🐲")
    print("Your mission: Find the treasure and escape alive!\n")

    while health > 0:
        # Show player status
        print("\n" + "="*30)
        print(f"Health: {'❤️' * (health // 20)}")  # Visual health bar
        print(f"Coins: {coins} 💰")
        if has_sword:
            print("Items: Sword ⚔️")
        if has_key:
            print("Items: Key 🗝️")
        print("="*30 + "\n")

        print("What would you like to do?")
        print("1. Explore deeper into the cave")
        print("2. Rest by the campfire")
        print("3. Look for items")
        print("4. Try to exit")

        choice = input("Enter your choice (1-4): ")
```

```
        if choice == "1":
            # Random encounter system
            import random
            encounter = random.randint(1, 3)

            if encounter == 1:
                print("\nYou found a treasure chest! 🎁")
                if has_key:
                    coins += 50
                    print("You used the key to open it and found 50
coins!")
                else:
                    print("You need a key to open it...")
            elif encounter == 2:
                print("\nA small dragon appears! 👹")
                if has_sword:
                    print("You use your sword to scare it away!")
                else:
                    damage = 20
                    health -= damage
                    print(f"The dragon breathes fire! You lose {damage}
health!")
            else:
                print("\nThe path is quiet... for now...")

        elif choice == "2":
            if health < 100:
                health += 20
                if health > 100:
                    health = 100
                print("\nYou rest by the fire and recover some health!
❤️")
            else:
                print("\nYou're already at full health!")

        elif choice == "3":
            if not has_sword or not has_key:
                found_item = random.randint(1, 3)
                if found_item == 1 and not has_sword:
                    has_sword = True
                    print("\nYou found a sword! ⚔️")
                elif found_item == 2 and not has_key:
                    has_key = True
                    print("\nYou found a key! 🗝️")
                else:
                    print("\nYou found nothing useful...")
            else:
                print("\nYou already have all the items!")

        elif choice == "4":
            if coins >= 100:
                print("\n🎉 Congratulations! You escape with the treasure!
🎉")
                print(f"You collected {coins} coins!")
```

```
                break
            else:
                print("\nYou need at least 100 coins to win!")
                print("Keep exploring!")

        # Check if player died
        if health <= 0:
            print("\n💀 Game Over! The dragon was too powerful! 💀")
            break

# Start the adventure
dragons_cave()
```

🎯 Practice Exercises - Level 1: Loops and Logic

Exercise 1: Countdown Timer ⏰

Create a countdown program that:

- Asks the user for a number
- Counts down from that number to 1
- Prints "Blast off! 🚀 " at the end

```
# Example output for input 5:
# 5...
# 4...
# 3...
# 2...
# 1...
# Blast off! 🚀
```

Exercise 2: Number Pattern 📊

Print this pattern of numbers:

```
# 1
# 2 2
# 3 3 3
# 4 4 4 4
# 5 5 5 5 5
```

Hint: Use nested loops!

Exercise 3: Even Sum Calculator ➗

Create a program that:

- Asks for a maximum number
- Calculates the sum of all even numbers from 2 to that number

- Prints the total

```
# Example: If user enters 8
# Even numbers: 2, 4, 6, 8
# Output: The sum is 20
```

Exercise 4: Password Checker 🔑

Make a simple password checker that:

- Gives the user 3 attempts to enter the correct password
- Password should be "python123"
- Tells them how many attempts they have left
- Prints "Access granted! 🎉 " or "Access denied! 🔒 "

Exercise 5: Number Guesser with Range 🎲

Create a number guessing game that:

- Asks for a number between 1-50
- Tells if the guess is "getting warmer" or "getting colder" compared to the previous guess
- Counts the number of guesses
- Has a maximum of 7 guesses

🎯 Practice Exercises - Level 2: Functions with Loops and Logic

Exercise 1: Temperature Converter 🌡️

Write a function that:

```
def convert_temperatures(temperatures, convert_to="F"):
    """
    Convert a list of temperatures between Celsius and Fahrenheit.
    Example: convert_temperatures([0, 100], "F") should return [32, 212]
    """
    # Your code here!
```

Exercise 2: Grade Calculator 📚

Create a function that:

```
def calculate_final_grade(assignments, final_exam):
    """
    Calculate final grade where:
    - assignments are worth 60% (can have variable number of assignments)
    - final_exam is worth 40%
    Return letter grade (A, B, C, D, or F) and percentage
    """
```

```
    # Your code here!

# Example usage:
grades = [85, 90, 88, 95]
final = 83
letter, percentage = calculate_final_grade(grades, final)
```

Exercise 3: Word Counter 📝

Write a function that:

```
def analyze_text(text):
    """
    Return a dictionary containing:
    - Number of words
    - Number of sentences
    - Number of uppercase letters
    - Number of lowercase letters
    """
    # Your code here!

# Example:
stats = analyze_text("Hello, World! This is Python.")
```

Exercise 4: Game Score Tracker 🎮

Create a function that:

```
def track_high_scores(new_score, high_scores):
    """
    Keep track of top 5 high scores.
    - Add new score if it qualifies
    - Keep list sorted (highest first)
    - Maximum 5 scores in list
    """
    # Your code here!

# Example:
scores = [100, 95, 90, 85, 80]
updated_scores = track_high_scores(92, scores)
# Should return: [100, 95, 92, 90, 85]
```

Exercise 5: Shopping Cart Calculator 🛒

Write a function that:

```
def calculate_shopping_cart(items, discounts):
    """
```

```
    Calculate total price where:
    - items is a dictionary of item:price pairs
    - discounts is a dictionary of item:discount_percentage pairs
    - Apply discounts when available
    - Add 8% tax at the end
    Return total price and savings
    """
    # Your code here!

# Example:
cart = {"apple": 0.50, "bread": 2.25, "milk": 3.50}
discounts = {"bread": 20, "milk": 10}
total, savings = calculate_shopping_cart(cart, discounts)
```

💡 Tips for Solving the Exercises

1. **Break it Down:**
    - Write down what the program needs to do
    - Break big problems into smaller steps
    - Test each step before moving to the next
2. **Start Simple:**
    - Get the basic version working first
    - Add extra features later
    - Test with simple inputs first
3. **Debug Strategy:**
    - Add print statements to check values
    - Test with different inputs
    - Check indentation
    - Read error messages carefully
4. **Challenge Yourself:**
    - After solving an exercise, try to:
        - Add extra features
        - Make it more user-friendly
        - Make the code shorter or more efficient
        - Add error handling

🌟 Extra Challenges

Challenge 1: Adventure Game Upgrade

Add to the Dragon's Cave game:

- More items to find
- Different types of monsters
- A shop system

- Multiple endings

Challenge 2: Advanced Number Guessing

Improve the Number Guessing game:

- Add different difficulty levels
- Keep track of high scores
- Add hints system
- Add multiplayer mode

Challenge 3: Create Your Own!

Design your own game using:

- At least 3 if-elif-else statements
- At least 2 different types of loops
- At least 2 functions
- A scoring system

🎯 Key Takeaways

1. **Control Flow Basics:**
    - If-statements help make decisions
    - Loops repeat actions
    - Functions organize code
    - Break and continue give more control
2. **Best Practices:**
    - Keep your code organized
    - Test frequently
    - Start simple, then add features
    - Comment your code
3. **Problem-Solving Steps:**
    - Understand the problem
    - Plan your solution
    - Write small pieces at a time
    - Test and fix errors
    - Improve your code

🚀 Next Steps

After completing this tutorial, you can:

1. Try all the practice exercises
2. Modify the sample projects
3. Create your own games
4. Share your code with friends

5. Help others learn!

🤔 Common Questions

1. **"My loop never ends!"**
    - Check your loop condition
    - Make sure values change inside the loop
    - Ensure you have a way to break the loop
2. **"My if-statement isn't working!"**
    - Check your indentation
    - Verify your comparison operators
    - Print values to debug
3. **"My function returns the wrong value!"**
    - Test with simple inputs
    - Add print statements
    - Check your calculations

Remember: Programming is like learning a new language - practice makes perfect! Don't be afraid to experiment and make mistakes. That's how we learn!

Need help? Ask your teacher or try explaining your code to a friend - sometimes just talking about it helps you find the solution!

Happy coding! 🚀