PYTHON PROGRAMMING FUNDAMENTALS

Universal Reference Cheat Sheet

PART 1: VARIABLES AND DATA TYPES

What is a Variable?

A variable is a named storage location that holds data. Variables are created through assignment.

```python
# Variable assignment
name = "Alice"        # String variable
age = 25              # Integer variable
height = 5.9          # Float variable
isStudent = True      # Boolean variable
```

Variable Naming Rules

- Start with letter (a-z, A-Z) or underscore (_)
- Can contain letters, numbers, underscores
- Case sensitive (name ≠ Name)
- Cannot use Python keywords (if, for, while, etc.)
- Use descriptive names

Good: studentName, totalScore, maxValue

Bad: x, data1, temp

Data Types

Integer (int) - Whole Numbers

```python
count = 42
negative = -15
zero = 0
```

Float - Decimal Numbers

```python
price = 19.99
temperature = 98.6
pi = 3.14159
```

String (str) - Text Data

```python
message = "Hello World"
```

```
singleQuotes = 'Also valid'
emptyString = ""
```

Boolean (bool) - True/False

```
isValid = True
isEmpty = False
```

Type Checking and Conversion

```
# Check data type
print(type(42))          # <class 'int'>
print(type("hello"))     # <class 'str'>

# Convert between types
ageStr = "25"
ageInt = int(ageStr)     # Convert string to integer
ageFloat = float(ageStr) # Convert string to float
ageBack = str(ageInt)    # Convert integer to string
```

PART 2: INPUT AND OUTPUT

Getting User Input

```
# Basic input (always returns string)
name = input("Enter your name: ")

# Input with type conversion
age = int(input("Enter your age: "))
height = float(input("Enter your height: "))
```

Displaying Output

```
# Basic print statements
print("Hello World")
print(42)

# Multiple values
name = "Alice"
age = 25
print("Name:", name, "Age:", age)

# String concatenation
print("Hello " + name)
print("Age: " + str(age))  # Must convert numbers to strings
```

String Formatting

```
name = "Bob"
score = 87.5


# f-string formatting (recommended)
print(f"Student: {name}")
print(f"Score: {score:.2f}")          # Two decimal places
print(f"Percentage: {score:.1f}%")   # One decimal place


# .format() method
print("Name: {}, Score: {:.2f}".format(name, score))


# Older % formatting
print("Name: %s, Score: %.2f" % (name, score))
```

PART 3: MATHEMATICAL OPERATIONS

Arithmetic Operators

```
# Basic operations
addition = 10 + 5        # 15
subtraction = 10 - 5     # 5
multiplication = 10 * 5  # 50
division = 10 / 3        # 3.333... (always returns float)
floorDivision = 10 // 3  # 3 (integer division)
modulus = 10 % 3         # 1 (remainder)
exponentiation = 2 ** 3  # 8 (2 to the power of 3)
```

Order of Operations (PEMDAS)

1. Parentheses ()
2. Exponents **
3. Multiplication *, Division /, Modulus %
4. Addition +, Subtraction -

```
result = 2 + 3 * 4       # 14 (not 20)
result = (2 + 3) * 4     # 20
```

Assignment Operators

```
x = 10
x += 5    # Same as: x = x + 5
x -= 3    # Same as: x = x - 3
x *= 2    # Same as: x = x * 2
```

```
x /= 4      # Same as: x = x / 4
```

PART 4: FUNCTIONS

Function Structure (Modular Design)

```python
def addNumbers(num1, num2):
    """Add two numbers together"""
    result = num1 + num2
    return result


def getUserInput():
    """Get a number from user"""
    return int(input("Enter a number: "))


def displayResult(result):
    """Display formatted output"""
    print(f"Result: {result}")


def main():
    """Main function orchestrates program flow"""
    # Get input
    value1 = getUserInput()
    value2 = getUserInput()

    # Process data
    result = addNumbers(value1, value2)

    # Display output
    displayResult(result)

# Run the program
main()
```

Grade Calculator Example (Modular)

```python
def calculateAverage(scores):
    """Calculate average of test scores"""
    total = sum(scores)
    return total / len(scores)


def getTestScores():
    """Get test scores from user"""
    scores = []
    count = int(input("How many test scores? "))

    for i in range(count):
```

```python
        score = float(input(f"Enter score {i+1}: "))
        scores.append(score)

    return scores


def getLetterGrade(average):
    """Convert average to letter grade"""
    if average >= 90:
        return "A"
    elif average >= 80:
        return "B"
    elif average >= 70:
        return "C"
    elif average >= 60:
        return "D"
    else:
        return "F"


def displayResults(average, letterGrade):
    """Show the final results"""
    print(f"Average: {average:.2f}")
    print(f"Grade: {letterGrade}")


def main():
    """Run the grade calculator"""
    # Get input
    scores = getTestScores()

    # Process
    average = calculateAverage(scores)
    letterGrade = getLetterGrade(average)

    # Output
    displayResults(average, letterGrade)


# Start the program
main()
```

PART 5: CONDITIONAL STATEMENTS

Basic If Statement

```python
age = 18
if age >= 18:
    print("You can vote")
```

If-Else Statement

```python
score = 75
if score >= 70:
    print("Pass")
else:
    print("Fail")
```

Multiple Conditions (elif)

```python
grade = 85

if grade >= 90:
    letter = "A"
elif grade >= 80:
    letter = "B"
elif grade >= 70:
    letter = "C"
elif grade >= 60:
    letter = "D"
else:
    letter = "F"

print(f"Grade: {letter}")
```

Comparison Operators

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

Logical Operators

```python
and    # Both conditions must be true
or     # At least one condition must be true
not    # Reverses the condition

# Examples
if age >= 18 and age <= 65:
    print("Working age")

if grade == "A" or grade == "B":
    print("Good grade")

if not isRaining:
    print("No umbrella needed")
```

PART 6: LOOPS

For Loops

```
# Basic for loop with range
for i in range(5):
    print(i)  # Prints 0, 1, 2, 3, 4

# Range with start and stop
for i in range(1, 6):
    print(i)  # Prints 1, 2, 3, 4, 5

# Range with step
for i in range(0, 10, 2):
    print(i)  # Prints 0, 2, 4, 6, 8

# Counting backwards
for i in range(10, 0, -1):
    print(i)  # Prints 10, 9, 8, ..., 1
```

While Loops

```
# Basic while loop
count = 1
while count <= 5:
    print(count)
    count += 1  # Important: update the condition variable

# Input validation with while loop
age = -1
while age < 0 or age > 120:
    age = int(input("Enter valid age (0-120): "))
```

Nested Loops

```
# Loop inside another loop
for row in range(3):
    for col in range(3):
        print(f"({row}, {col})")
```

PART 7: INPUT VALIDATION

Basic Error Handling

```python
# Handle invalid input
try:
    age = int(input("Enter age: "))
    print(f"Your age is {age}")
except ValueError:
    print("Please enter a valid number")
```

Input Validation Example (Modular)

```python
def getValidNumber(prompt):
    """Get valid number from user"""
    while True:
        try:
            value = int(input(prompt))
            return value
        except ValueError:
            print("Please enter a valid number")


def getValidAge():
    """Get valid age between 0 and 120"""
    while True:
        age = getValidNumber("Enter your age: ")
        if 0 <= age <= 120:
            return age
        print("Age must be between 0 and 120")


def getYesOrNo(question):
    """Get yes/no answer from user"""
    while True:
        answer = input(question + " (y/n): ").lower()
        if answer in ['y', 'yes']:
            return True
        elif answer in ['n', 'no']:
            return False
        print("Please enter 'y' or 'n'")


def showPersonInfo(age, isStudent):
    """Display person information"""
    print(f"Age: {age}")
    print(f"Student: {'Yes' if isStudent else 'No'}")


def main():
    """Run the person information program"""
    # Get input with validation
    age = getValidAge()
    isStudent = getYesOrNo("Are you a student?")

    # Show results
    showPersonInfo(age, isStudent)
```

```
# Run the program
main()
```

PART 8: COMMON PROGRAMMING PATTERNS

Basic Calculator (Modular)

```python
def getNumbers():
    """Get two numbers from user"""
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    return num1, num2


def addNumbers(num1, num2):
    """Add two numbers"""
    return num1 + num2


def showResult(num1, num2, result):
    """Display the calculation result"""
    print(f"{num1} + {num2} = {result:.2f}")


def main():
    """Run the calculator program"""
    # Input
    number1, number2 = getNumbers()

    # Process
    result = addNumbers(number1, number2)

    # Output
    showResult(number1, number2, result)

# Run the program
main()
```

Advanced Calculator (Modular)

```python
def getOperation():
    """Get math operation from user"""
    while True:
        op = input("Enter operation (+, -, *, /): ")
        if op in ['+', '-', '*', '/']:
            return op
        print("Please enter +, -, *, or /")


def getNumber(message):
```

```python
    """Get valid number from user"""
    while True:
        try:
            return float(input(message))
        except ValueError:
            print("Please enter a valid number")


def doMath(num1, num2, operation):
    """Perform the calculation"""
    if operation == '+':
        return num1 + num2
    elif operation == '-':
        return num1 - num2
    elif operation == '*':
        return num1 * num2
    elif operation == '/':
        if num2 != 0:
            return num1 / num2
        else:
            return None


def showAnswer(num1, num2, operation, result):
    """Display the calculation result"""
    if result is not None:
        print(f"{num1} {operation} {num2} = {result:.2f}")
    else:
        print("Error: Cannot divide by zero")


def main():
    """Run the calculator"""
    print("Calculator Program")

    # Get input
    num1 = getNumber("First number: ")
    operation = getOperation()
    num2 = getNumber("Second number: ")

    # Calculate
    result = doMath(num1, num2, operation)

    # Show result
    showAnswer(num1, num2, operation, result)

# Start program
main()
```

Menu Program (Modular)

```python
def showMenu():
```

```python
    """Show the menu options"""
    print("\n=== MENU ===")
    print("1. Add")
    print("2. Subtract")
    print("3. Multiply")
    print("4. Divide")
    print("5. Quit")


def getChoice():
    """Get menu choice from user"""
    while True:
        choice = input("Pick option 1-5: ")
        if choice in ['1', '2', '3', '4', '5']:
            return choice
        print("Enter 1, 2, 3, 4, or 5")


def getTwoNumbers():
    """Get two numbers for math"""
    num1 = float(input("First number: "))
    num2 = float(input("Second number: "))
    return num1, num2


def doOperation(choice, num1, num2):
    """Do the math operation"""
    if choice == '1':
        return num1 + num2
    elif choice == '2':
        return num1 - num2
    elif choice == '3':
        return num1 * num2
    elif choice == '4':
        if num2 != 0:
            return num1 / num2
        else:
            return None


def showResult(num1, num2, choice, result):
    """Show the math result"""
    operations = ['', 'Addition', 'Subtraction', 'Multiplication', 'Division']

    if result is not None:
        print(f"{operations[int(choice)]}: {num1} and {num2} = {result:.2f}")
    else:
        print("Cannot divide by zero")


def main():
    """Main program"""
    print("Calculator Program")
```

```
    while True:
        showMenu()
        choice = getChoice()

        if choice == '5':
            print("Goodbye!")
            break

        num1, num2 = getTwoNumbers()
        result = doOperation(choice, num1, num2)
        showResult(num1, num2, choice, result)


# Start the program
main()
```

PART 9: ESSENTIAL FUNCTIONS AND METHODS

Built-in Functions

```
abs(-5)              # Absolute value: 5
round(3.7)           # Round to nearest integer: 4
round(3.14159, 2) # Round to 2 decimals: 3.14
max(1, 5, 3)         # Maximum: 5
min(1, 5, 3)         # Minimum: 1
len("hello")         # String length: 5
```

String Methods

```
text = "Hello World"
text.upper()       # "HELLO WORLD"
text.lower()       # "hello world"
text.strip()       # Remove whitespace from ends
text.replace("o", "0")  # "Hell0 W0rld"
text.split()       # ["Hello", "World"]
```

Math Operations

```
import math
math.sqrt(16)      # Square root: 4.0
math.ceil(3.2)     # Round up: 4
math.floor(3.8)    # Round down: 3
```

PART 10: DEBUGGING CHECKLIST

Common Syntax Errors

- Missing colons after if, for, while, def
- Incorrect indentation (use 4 spaces consistently)
- Unmatched parentheses or quotes
- Misspelled variable names
- Using = instead of == in conditions

Common Logic Errors

- Forgetting to convert input() to numbers
- Off-by-one errors in loops
- Infinite loops (forgetting to update loop variable)
- Using wrong comparison operators

Testing Strategy

1. Test with known, simple values
2. Test edge cases (0, negative numbers, empty strings)
3. Test invalid input
4. Trace through code step by step

QUICK REFERENCE

Format Specifiers

```
score = 87.56789
print(f"{score:.2f}")    # 87.57 (2 decimal places)
print(f"{score:.1f}")    # 87.6 (1 decimal place)
print(f"{score:>10}")    # Right align in 10 spaces
print(f"{score:<10}")    # Left align in 10 spaces
```

Common Patterns

```
# Check if number is even
if num % 2 == 0:
    print("Even")


# Count iterations
for i in range(len(items)):
    print(f"Item {i}: {items[i]}")


# Accumulator pattern
```

```
total = 0
for num in numbers:
    total += num
```

Remember:

- Read problems carefully
- Plan before coding
- Use descriptive variable names
- Test your code
- Comment complex logic

Good luck on your exam!