

CS-121:

Week 4



Review



Today's Topics

- Expressions and operators
- Conditionals (if statements)
- Defining functions
- Puzzles!

Expressions

- An expression is a combination of values, variables, and operators that computes to a value.
- Examples:
 - `2 + 3`
 - `x * y`
 - `len(name) > 5`

Operators Review

- **Arithmetic Operators:**
 - + Addition
 - – Subtraction
 - * Multiplication
 - / Division
 - // Floor Division
 - % Modulus (Remainder)
 - ** Exponentiation

Comparison Operators

- == Equal to
- != Not equal to
- > Greater than
- < Less than
- >= Greater than or equal to
- <= Less than or equal to

Logical Operators

- and Returns True if both statements are true
- or Returns True if one of the statements is true
- not Reverse the result, returns False if the result is true

Example Expressions

```
x = 10
```

```
y = 5
```

```
print(x > y)           # True
```

```
print(x == y)          # False
```

```
print((x > y) and (y > 0)) # True
```

```
print(not(x < y))       # True
```


Conditionals: if Statements

- Control the flow of your program based on conditions.
- **Syntax:**

```
if condition:
```

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

Example if Statement

```
age = int(input("Enter your age: "))
```

```
if age >= 18:  
    print("You are an adult.")
```

if...else Statements

- Provide alternative execution when the condition is False.

```
if condition:  
    # Code if True  
else:  
    # Code if False
```

Example if...else

```
number = int(input("Enter a number: "))
```

```
if number % 2 == 0:  
    print("Even number.")  
else:  
    print("Odd number.")
```

if...elif...else Statements

- Check multiple conditions.

```
if condition1:  
    # Code if condition1 is True  
elif condition2:  
    # Code if condition2 is True  
else:  
    # Code if neither condition is True
```

Example

if...elif...else

```
score = int(input("Enter your score: "))
```

```
if score >= 90:  
    grade = 'A'  
elif score >= 80:  
    grade = 'B'  
elif score >= 70:  
    grade = 'C'  
elif score >= 60:  
    grade = 'D'  
else:  
    grade = 'F'
```

```
print(f"Your grade is {grade}.")
```

Nested Conditionals

- Conditionals inside conditionals.

```
age = int(input("Enter your age: "))
```

```
if age >= 0:
    if age < 18:
        print("You are a minor.")
    else:
        print("You are an adult.")
else:
    print("Invalid age entered.")
```

Defining Functions

- A function is a reusable block of code that performs a specific task.
- **Syntax:**

```
def function_name(parameters):  
    # Code block  
    return value
```


Example Function

```
def greet(name):  
    return f"Hello, {name}!"
```

```
message = greet("Alice")  
print(message)
```

Functions with Multiple Parameters

```
def add_numbers(a, b):  
    return a + b
```

```
result = add_numbers(5, 7)  
print(f"The sum is {result}.")
```

Why Use Functions?

- **Reusability:** Write once, use multiple times.
- **Modularity:** Break down complex problems.
- **Readability:** Easier to understand and maintain.

Puzzle 1: Operator Precedence and Expressions

What is the output of the following code snippet?

```
result = 5 + 2 * (10 // 4) - 3 ** 2  
print(result)
```

Answer:

Let's break down the expression step by step, following operator precedence:

1. Parentheses:

- $10 // 4$ performs floor division: $10 // 4 = 2$

2. Multiplication:

- $2 * 2 = 4$

3. Exponentiation:

- $3 ** 2 = 9$

4. Addition and Subtraction:

- $5 + 4 = 9$

Puzzle 2: Logical Operators and Short-Circuit Evaluation

Consider the following code:

```
x = 10
y = 0

if x > 5 or (10 / y) > 1:
    print("Condition met")
else:
    print("Condition not met")
```

What happens when this code is executed?

Answer:

- The condition uses `or`, which short-circuits if the first condition is `True`.
- `x > 5` is `True` (since `10 > 5`).
- Because of short-circuiting, `(10 / y)` is **not evaluated**, avoiding a `ZeroDivisionError`.
- Therefore, the code prints `"Condition met"` without error.

Output:

```
Condition met
```

Puzzle 3: Understanding Conditionals with Elif

What will be printed when the following code runs with `score = 75`?

```
score = 75

if score >= 90:
    print("Grade: A")
elif score >= 80:
    print("Grade: B")
elif score >= 70:
    print("Grade: C")
elif score >= 60:
    print("Grade: D")
else:
    print("Grade: F")
```


Answer:

- `score >= 90` is `False`.
- `score >= 80` is `False`.
- `score >= 70` is `True`.
- Therefore, it prints "Grade: C".

Output:

Grade: C

Puzzle 4: Functions and Return Values

Given the following function definitions, what is the output of the code?

```
def square(n):  
    return n * n
```

```
def subtract(a, b):  
    return a - b
```

```
result = subtract(square(5), square(3))  
print(result)
```

Answer:

Compute `square(5)`:

- `square(5)` returns $5 * 5 = 25$

Compute `square(3)`:

- `square(3)` returns $3 * 3 = 9$

Compute `subtract(25, 9)`:

- `subtract(25, 9)` returns $25 - 9 = 16$

Output:

Puzzle 5: Conditional Expressions and Truthy Values

What will be printed when the following code is executed?

```
value = "Hello"

if value:
    print("Value exists")
else:
    print("No value")
```

Answer:

In Python, non-empty strings are considered True in a boolean context.

- Since `value` is "Hello" (a non-empty string), the condition `if value` is True.
- Therefore, it prints "Value exists".

Output:

Value exists

Recap

- **Expressions** compute values.
- **Conditionals** control the flow based on conditions.
- **Functions** encapsulate reusable code.



Let's Code!

- Open your code notebook.
- Experiment with conditionals and functions.