

Python Revisit Lesson for Algebra Simulator

This lesson revisits **only the Python concepts required** for building the **Elementary and Intermediate Algebra Simulator**. All examples are directly related to algebraic operations, equation solving, and graphing.

Key Concepts

- Sympy basics (symbols, expressions, equations)
 - Simplification, solving, factoring
 - Conditionals & loops for algebra-related tasks
 - Functions for reusable algebra operations
 - Error handling for invalid algebra input
 - Graphing algebraic functions with Matplotlib
-

1. Python Basics for Algebra

1.1 Variables and Symbols

```
import sympy as sp
x, y = sp.symbols('x y')
expr = x + y
print(expr)
```

1.2 Input and Output

```
expr_input = input("Enter an expression in x: ")
expr = sp.sympify(expr_input)
print("You entered:", expr)
```

1.3 Comments

```
# Comments help explain algebra code
```

2. Operators in Algebra

2.1 Arithmetic Operators

```
x = sp.symbols('x')
expr1 = x + 2
expr2 = x**2
expr3 = expr1 * expr2
print(expr3)
```

2.2 Comparison Operators

```
x = sp.symbols('x')
print(sp.Eq(x+2, 5))    # equation: x+2 = 5
```

2.3 Logical Operators (used in conditions)

```
num = 5
print(num > 0 and num < 10)    # algebraic check conditions
```

3. Conditionals (Decision Making)

```
expr = sp.sympify("x**2 - 4")
if expr.is_polynomial():
    print("This is a polynomial.")
else:
    print("Not a polynomial.")
```

4. Loops (Useful for Algebra Tables and Graphs)

4.1 For Loop Example (multiplication table)

```
x = sp.symbols('x')
for i in range(1, 6):
    print(f"x * {i} =", x * i)
```

4.2 While Loop Example

```
count = 1
while count <= 5:
    print("x^", count, "=", x**count)
    count += 1
```

5. Functions (Reusable Algebra Tasks)

```
def simplify_expression(expr_str):
    expr = sp.sympify(expr_str)
    return sp.simplify(expr)

print(simplify_expression("x^2 + 2*x + 1"))
```

6. Error Handling (Safe Algebra Operations)

```
try:
    expr = sp.sympify("x/0")
except Exception as e:
    print("Error:", e)
```

7. Sympy for Algebra

7.1 Simplifying Expressions

```
expr = sp.sympify("x^2 + 2*x + 1")
print(sp.simplify(expr))
```

7.2 Solving Equations

```
solution = sp.solve(sp.Eq(x**2 - 4, 0), x)
print(solution)
```

7.3 Factoring Expressions

```
factored = sp.factor(x**2 - 4)
print(factored)
```

8. Graphing Functions

```
import matplotlib.pyplot as plt
import numpy as np

f = sp.sympify("x**2 - 3*x + 2")
f_lambdified = sp.lambdify(x, f, "numpy")

x_vals = np.linspace(-10, 10, 400)
y_vals = f_lambdified(x_vals)

plt.plot(x_vals, y_vals)
plt.axhline(0, color='black')
plt.axvline(0, color='black')
plt.title("Graph of y = x^2 - 3x + 2")
plt.grid(True)
plt.show()
```