Data Types Covered

We started by looking at examples of different data types:

- **Integers:** `2, 5, -3, 4, 31241251, 153, 15174351463, -1234561341`
- **Floats:** `2.1, 3.0, 0.00000, 3.141592, 2.71828, -4.2`
- **Strings:** `'a', 'cat', "cats", "dogs", "cars", "houses", 'planes'`
- **Booleans:** `True`, `False`

Each of these data types serves different purposes in Python.

Arithmetic Operations

Multiplication and Exponentiation

```
print(2 * 3)
print(2 * 3.5)
print(2 ** 3)
```

Explanation:

- `2 * 3` multiplies two integers (result is 6).
- `2 * 3.5` multiplies an integer by a float (result is 7.0).
- `2 ** 3` is exponentiation (2 raised to 3, result is 8).

Order of Operations (Operator Precedence)

```
print(1 + 2 * 3)
```

Explanation:

- Multiplication has higher precedence than addition, so `2 * 3` is computed first.
- Then we add 1, producing 7.

Combination of Operations

```
x = 5
y = 2
print(x * y - x - y)
```

Explanation:

- `x * y` multiplies 5 by 2, giving 10.
- Then we subtract `x` (5), resulting in 5.
- Finally we subtract `y` (2), leaving 3.

Strings

Basic String Printing

```
print('fat cat')
```

Explanation: Simply prints the string `'fat cat'` to the console.

String Variables and Concatenation

```
first_name = "Richie"
last_name = "Pasumarty"
full_name = first_name + " " + last_name
print(full_name)
```

Explanation:

- `first_name` and `last_name` are both strings.
- Adding them with a space in between creates `"Richie Pasumarty"`.

Adding a Number to a String

```
print(first_name + "3")
```

Explanation: Python allows concatenation of two strings. Here, `"3"` is also a string, so it can be appended to `first_name` without error.

Type Conversion and Multiplying Strings

```
print(int(3.0) * "HI")
```

Explanation:

- `int(3.0)` converts the float 3.0 to an integer 3.
- Multiplying a string by an integer repeats the string that many times, resulting in `"HIHIHI"`.

Booleans

```
print(True)
print(False)
```

Explanation:

- `True` and `False` are boolean values.
- Booleans often come into play in conditionals, but here we just print them.

Reading Input from the User

Reading Strings

```
name = input("Please enter your name: ")
print(f"Your name is {name}")
```

Explanation:

- `input()` reads a string from the user.
- We then store that string in `name` and print it out with an f-string.

Reading and Converting to Integer

```
class_number = '2001'
print("The original type of class_number is", type(class_number))
class_number = int(class_number)
print("The new type of class_number is", type(class_number))

class_number = int(input("Please enter the class number: "))
print(type(class_number))
```

Explanation:

- A string that contains digits can be converted to an integer with `int()`.
- `type()` shows the data type of the variable.

Reading and Converting to Float

```
class_number = float(input("Please enter the class number: "))
print(type(class_number))
```

Explanation:

- We can convert user input to a floating-point number with `float()`.
- Useful for cases when the input might be a decimal value.