🐍 Python Numbers Mastery Guide

Welcome to the ultimate guide on **Python Numbers**! This guide covers everything from core concepts to advanced number manipulations, internal working, and useful libraries like `math`, `random`, `decimal`, and more! Let's go! 🚀

🔢 1. Number Basics in Python

✅ Immutable Nature

- Numbers in Python are **immutable**.
- When we change a number, Python **creates a new object**.

```
x = 5
print(id(x))  # Unique ID in memory
x = x + 1
print(id(x))  # Different ID → New object
```

🧠 Type & Identity

```
x = 10
print(type(x))  # <class 'int'>
print(isinstance(x, int))  # True
```

🧪 Typecasting

```
int("10")       # ➡️ 10
float(5)         # ➡️ 5.0
str(20.5)        # ➡️ '20.5'
bool(1)          # ➡️ True
```

🔍 2. Truthy and Falsy Numbers

- `0`, `0.0`, `0j` → **Falsy**
- Any non-zero value → **Truthy**

```
if 5:
    print("True")  # ✅ True
```

```
if 0:
    print("False")  # ❌ Won’t execute
```

📏 3. Number Precision and Float Problems

❌ Issue:

```
0.1 + 0.1 + 0.1 == 0.3  # False ❗
```

✅ Solution: `decimal`

```
from decimal import Decimal
print(Decimal("0.1") + Decimal("0.1") + Decimal("0.1") == Decimal("0.3"))  # ✅
True
```

🧮 4. Math Operations and Libraries

📦 `math` Module

```
import math

math.sqrt(16)         # ➡️ 4.0
math.pow(2, 3)        # ➡️ 8.0
math.floor(3.7)       # ➡️ 3
math.ceil(3.2)        # ➡️ 4
math.trunc(-3.9)      # ➡️ -3 (towards 0)
```

💡 Floor vs Trunc

- `floor`: Goes down to nearest integer.
- `trunc`: Trims decimal (towards 0).

🎲 5. Random Numbers

📦 `random` Module

```
import random

random.randint(1, 10)        # Random int between 1-10
random.choice([1,2,3])       # Random element from list
```

```
tea_list = ["Lemon", "Masala", "Mint"]
random.shuffle(tea_list)     # Shuffles list in-place
```

🧠 6. Comparisons and Booleans

🔁 Operators

```
x = 5.0
y = 5.0

x == y        # True
x != y        # False
x < y         # False
x >= y        # True
```

🧠 Chained Comparison

```
x = 2
y = 3
z = 4
print(x < y < z)  # ✅ True
```

🔀 Logical Operators

```
x = 2
y = 3
z = 4

print(x < y and y < z)  # ✅ True
print(True and False)   # ❌ False
```

🧠 True vs False Internals

- `True == 1` → ✅
- `False == 0` → ✅
- But `True is 1` → ❌

```
print(True == 1)       # True
print(True is 1)       # False
```

🔣 7. Number Systems (Base Conversion)

🔢 Binary, Octal, Hexadecimal

```
bin(64)      # '0b1000000'
oct(64)      # '0o100'
hex(255)     # '0xff'
```

🎯 Convert from literals

```
int('0b1010', 2)  # 10
int('0xff', 16)   # 255
```

🧮 8. Fractions and Precise Arithmetic

📦 `fractions` Module

```
from fractions import Fraction
f = Fraction(2, 7)
print(f + Fraction(1, 7))  # 3/7
```

🌈 9. Complex & Imaginary Numbers

```
x = 2 + 3j
print(x.real)  # ➡️ 2.0
print(x.imag)  # ➡️ 3.0
```

🧺 10. Sets (Mathematical Set Type)

🎯 Creating Sets

```
s1 = {1, 2, 3}
s2 = {2, 3, 4}
```

🧠 Set Operations

```
s1 & s2    # ➡️ {2, 3} Intersection
s1 | s2    # ➡️ {1, 2, 3, 4} Union
s1 - s2    # ➡️ {1} Difference
```

⚠️ Empty Set

```
empty = set()  # Not {}
```

📊 11. Bitwise Operators

```
x = 1
print(x << 2)  # ➡️ 4 (Shift left by 2 bits)

x = 5
y = 3
print(x & y)   # ➡️ 1 (Bitwise AND)
```

✅ Final Thoughts

Python handles numbers efficiently and provides libraries for:

- **Precision** (`decimal`, `fractions`)
- **Math utilities** (`math`, `cmath`)
- **Randomization** (`random`)
- **Set operations**
- **Base conversions**

🎯 Once you're confident here, you can confidently work on **scientific computing, data science, and algorithm development**!
