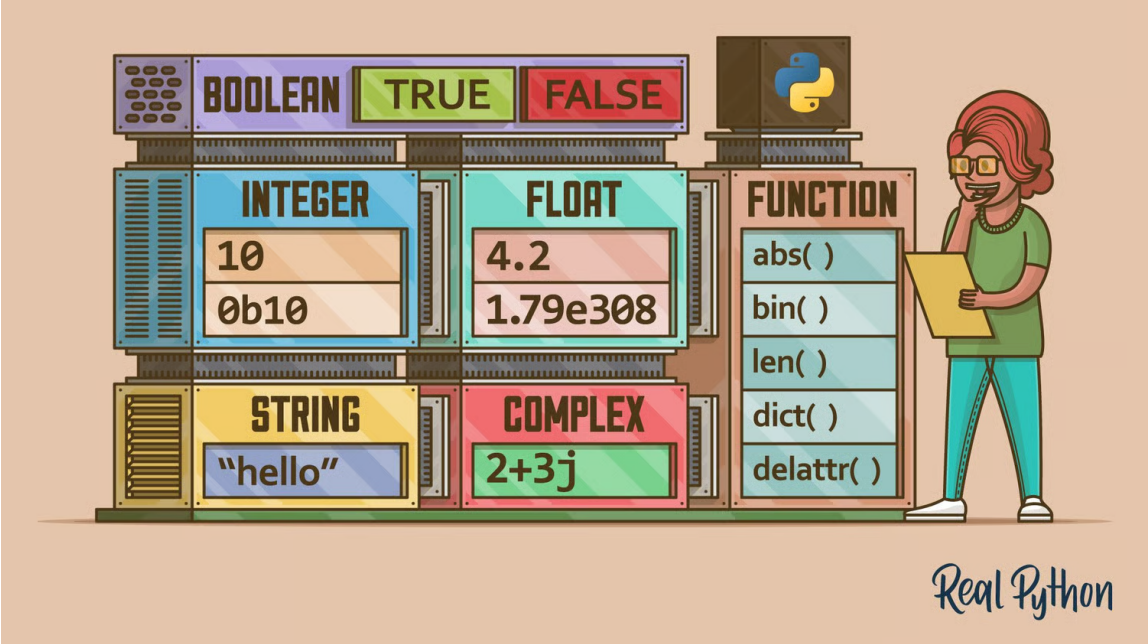
> created by Luis S. Flores for ITESM Contact: Luis_SFU@tec.mx
> August/2025

Data types, operator hierarchy, and mathematical expressions

Identifiers

- First character must be a letter or underscore.
- Other characters can be letters or digits.
- Examples: aux, sum, accum, a1, variable, variable1, _age.



Data types

- int: whole numbers (no decimals). Example: a = 5
- float: numbers with decimals. Example: b = 7.5
- str: text in " " or ' '. Example: "Street X #33 corner with Street Y"
- bool: True or False.

In [6]:
```
# Quick type demos
a = 5          # int
b = 7.5        # float
address = "Street X #33 corner with Street Y"  # str
logic1 = True  # bool
```

```
print(type(a), a)
print(type(b), b)
print(type(address), address)
print(type(logic1), logic1)
```

```
<class 'int'> 5
<class 'float'> 7.5
<class 'str'> Street X #33 corner with Street Y
<class 'bool'> True
```

Data types: str

To store text, use the string (str) type. The value must be enclosed in double quotes or in single quotes.

Examples:

In [14]:
```
address = "Street X #33 corner with Street Y"
message = 'Python is "the best" language'

print(address)
print(message)
```

```
Street X #33 corner with Street Y
Python is "the best" language
```

In [20]:
```
name = "Luis"
age = 29

print(name, "is",age)
```

```
Luis is 29
```

Arithmetic operator hierarchy

1. Parentheses `()`
2. Exponentiation `**`
3. Multiplication / Division / Floor / Modulo
4. Addition / Subtraction

Example:

5 + 9 * 2 → 5 + 18 → 15.0

In [24]:
```
expr = 5 + 9 * 2 - 40/5
print(expr)
```

```
15.0
```

Relational operators

- Compare values; result is True or False.

Example:

(7**2) > (25*2) → 49 > 50 → False

In [10]:
```
print((7**2) > (25*2))
```

```
False
```

Logical operators

- Combine conditions using: and, or, not

In [18]:
```
x=10
y=5
print("(x > 0) and (y > 0)  :", (x > 0) and (y > 0))
print("(x < 0) or (y > 0)   :", (x < 0) or (y > 0))
print("not (x > y)          :", not (x > y))
```

```
(x > 0) and (y > 0)  : True
(x < 0) or (y > 0)   : True
not (x > y)          : False
```

int and float operations in Python

• If either number is a `float`, the result is a `float`.

• In Python 3, division `/` always returns a `float` — even if both operands are `int`.

• The variable type can change automatically depending on the operation.

In [5]:
```
# Initial integers
a = 5
b = 2
print("a =", a, "| type:", type(a))
print("b =", b, "| type:", type(b))

# Integer division
result1 = a // b
print("\nInteger division (//):", result1, "| type:", type(result1))

# Regular division with integers
result2 = a / b
print("Regular division (/):", result2, "| type:", type(result2))

# Mixing int and float
c = 5.0
result3 = a + c
print("\nAddition with float:", result3, "| type:", type(result3))

# Changing value of int variable with float operation
a = a / b
print("\nAfter a = a / b -> a =", a, "| type:", type(a))
```

```
a = 5 | type: <class 'int'>
b = 2 | type: <class 'int'>

Integer division (//): 2 | type: <class 'int'>
Regular division (/): 2.5 | type: <class 'float'>

Addition with float: 10.0 | type: <class 'float'>

After a = a / b -> a = 2.5 | type: <class 'float'>
```

Compound interest with regular deposits

$$A = P\left(1 + \frac{r}{n}\right)^{nt} + \frac{d\left[\left(1 + \frac{r}{n}\right)^{nt} - 1\right]}{\frac{r}{n}}$$

In [22]:
```python
# Compound interest with regular deposits
P = float(input("input value for P (principal amount): "))
r = float(input("input value for r (annual interest rate in decimal, e.g. 0.
n = float(input("input value for n (compounding periods per year): "))
t = float(input("input value for t (number of years): "))
d = float(input("input value for d (deposit per period): "))

A = P * (1 + r/n)**(n*t) + d * ((1 + r/n)**(n*t) - 1) / (r/n)

print(f"A = {A}")
```

```
A = 3.562500000000001
```

Volume of a frustum of a cone

$$V = \frac{\pi h}{3}\left(R^2 + Rr + r^2\right)$$

In [23]:
```python
from math import pi

R = float(input("input value for R (base radius): "))
r = float(input("input value for r (top radius): "))
h = float(input("input value for h (height): "))

V = (pi * h / 3) * (R**2 + R*r + r**2)

print(f"V = {V}")
```

```
V = 29.059732045705587
```

In [ ]:
```python

```