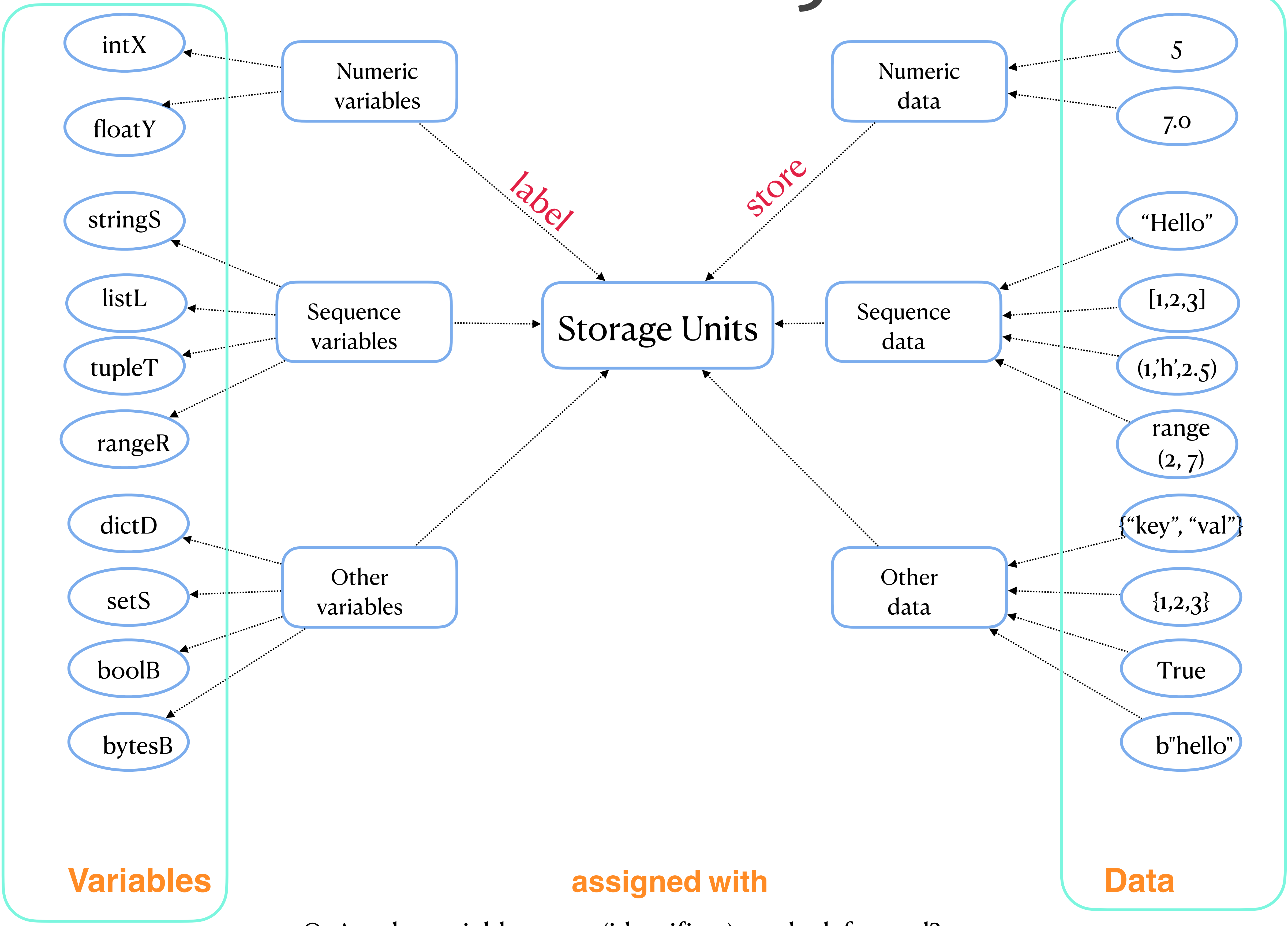
Ch2: Variables & Expressions

by Na Liu

Review of Key Topics

- Variables and assignments
- Identifiers
- Objects
- Numeric types: Floating-point
- Arithmetic expressions
- python expressions
- Division and modulo
- Module basics
- Math module

Variables and assignments



intX

floatY

stringS

listL

tupleT

rangeR

dictD

setS

boolB

bytesB

Numeric variables

Sequence variables

Other variables

label

store

Storage Units

Numeric data

Sequence data

Other data

5

7.0

“Hello”

[1,2,3]

(1,’h’,2.5)

range (2, 7)

{“key”, “val”}

{1,2,3}

True

b"hello"

Variables

assigned with

Data

Q: Are the variable names(identifiers) on the left good?

Identifiers

- **Valid Characters:**
  - Letters (a-z, A-Z)
  - Digits (0-9)
  - Underscore (_)
- **Must Start With:**
  - Letter (a-z, A-Z)
  - Underscore (_)
- **Case-Sensitive:**
  - Cat and cat are different variables
- **Avoid Double Underscores:**
  - (e.g., __init__) have special meanings in Python.
- **Best Practices:**
  - Use lowercase letters
  - Use underscores to separate words (e.g., temperature_celsius)
  - Choose descriptive names (e.g., age instead of a)
  - Balance meaning with brevity

Object Types

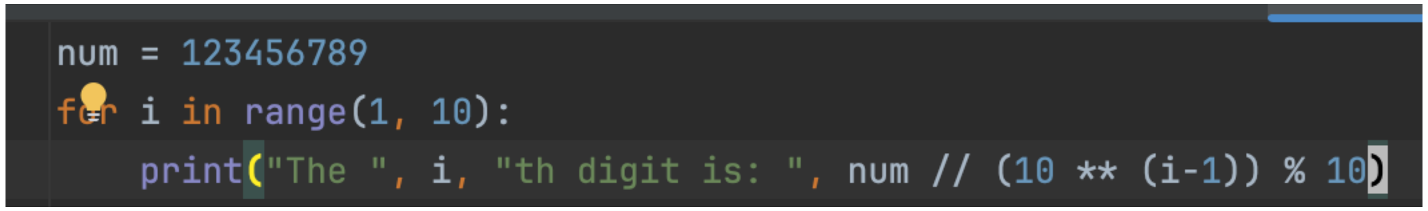
- **type() function**
- **Immutable Types:**
  - Numbers (integers, floats, booleans)
  - Strings
  - Tuples
- **Mutable Types:**
  - List
  - Dictionary
  - Sets
  - Byte
- **id() function**

Expressions

- Use correct arithmetic operator
- Assignment are not expressions
- Following precedence rules
- Single space around operators
- Compound operators

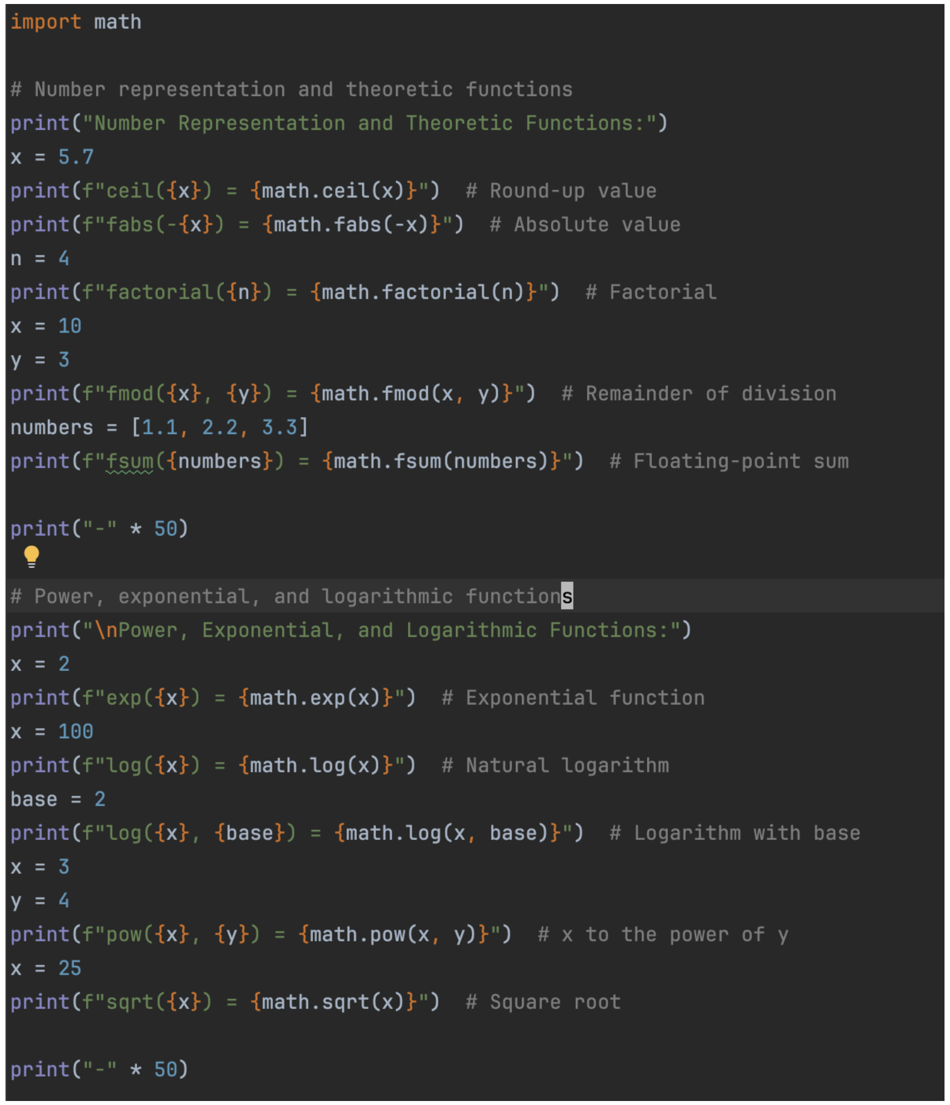
Special Operators

- Division operator(/) vs floor division operator(//)
- Modulo(%)
- example:



```python
num = 123456789
for i in range(1, 10):
    print("The ", i, "th digit is: ", num // (10 ** (i-1)) % 10)
```

Math Module —import math



```
import math

# Number representation and theoretic functions
print("Number Representation and Theoretic Functions:")
x = 5.7
print(f"ceil({x}) = {math.ceil(x)}")  # Round-up value
print(f"fabs(-{x}) = {math.fabs(-x)}")  # Absolute value
n = 4
print(f"factorial({n}) = {math.factorial(n)}")  # Factorial
x = 10
y = 3
print(f"fmod({x}, {y}) = {math.fmod(x, y)}")  # Remainder of division
numbers = [1.1, 2.2, 3.3]
print(f"fsum({numbers}) = {math.fsum(numbers)}")  # Floating-point sum

print("-" * 50)

# Power, exponential, and logarithmic functions
print("\nPower, Exponential, and Logarithmic Functions:")
x = 2
print(f"exp({x}) = {math.exp(x)}")  # Exponential function
x = 100
print(f"log({x}) = {math.log(x)}")  # Natural logarithm
base = 2
print(f"log({x}, {base}) = {math.log(x, base)}")  # Logarithm with base
x = 3
y = 4
print(f"pow({x}, {y}) = {math.pow(x, y)}")  # x to the power of y
x = 25
print(f"sqrt({x}) = {math.sqrt(x)}")  # Square root

print("-" * 50)
```

Random Module —import random

- 0.0 <= random.random() < 1.0
- min <= random.randrange(min, max) < max
- min <= random.randint(min, max) <= max
- random.seed()

Representing text

- Escape sequences
- ord(char)
- chr(int)