Titolo della Slide

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing

- Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
- Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
- Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Why Python?

- Easy to learn and read
- Versatile for web, data science, AI
- Large community and ecosystem
- Cross-platform compatibility
- Excellent documentation

Core Features

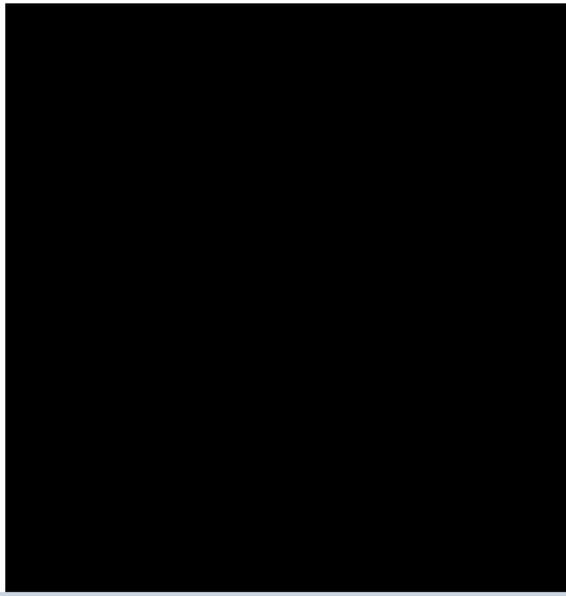
- Dynamic typing for flexibility
- Object-oriented programming support
- Functional programming capabilities
- Extensive standard library
- Third-party package ecosystem

Data Types

- Integers and floating-point numbers
- Strings for text manipulation
- Lists for ordered collections
- Dictionaries for key-value pairs
- Sets for unique elements
- Tuples for immutable sequences

Control Flow

- if/elif/else statements
- for loops for iteration
- while loops for conditions
- break and continue keywords
- List comprehensions
- Generator expressions



List Comprehension Example

```
# Traditional approach
squares = []
for i in range(10):
    squares.append(i ** 2)

# List comprehension
squares = [i ** 2 for i in range(10)]

# With condition
even_squares = [i ** 2 for i in range(10) if i % 2 == 0]

# Nested comprehension
matrix = [[i * j for j in range(5)] for i in range(5)]
```

Mathematical Foundations

The time complexity of common algorithms:

Linear search

$$O(n) = n$$

Binary search

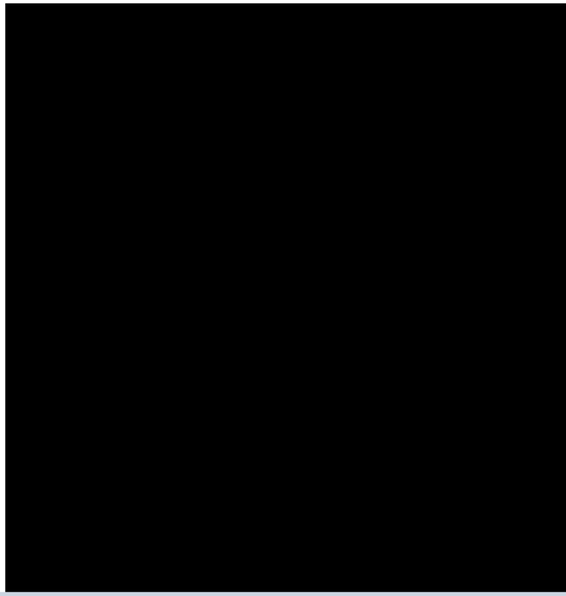
$$O(\log n) = \log_2 n$$

Quadratic algorithms

$$O(n^2) = n^2$$

General polynomial

$$T(n) = \sum_{i=0}^{k} a_i n^i$$



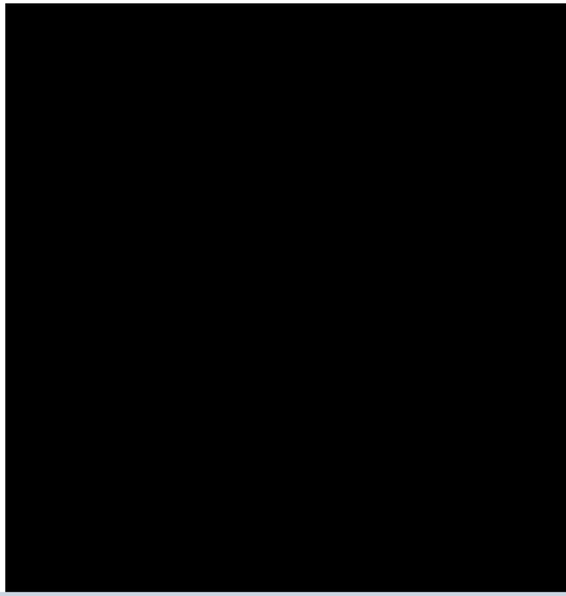
Functions

- Function definition with def
- Parameters and arguments
- Return values
- Default parameters
- *args and **kwargs
- Lambda functions

Function Example

```
def timer_decorator(func):
    def wrapper(*args, **kwargs):
        import time
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print(f"{func.__name__} took {end - start:.4f}s")
        return result
    return wrapper

@timer_decorator
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n-1) + fibonacci(n-2)
```



Statistics & Probability

Key statistical measures used in data science:

Mean

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Variance

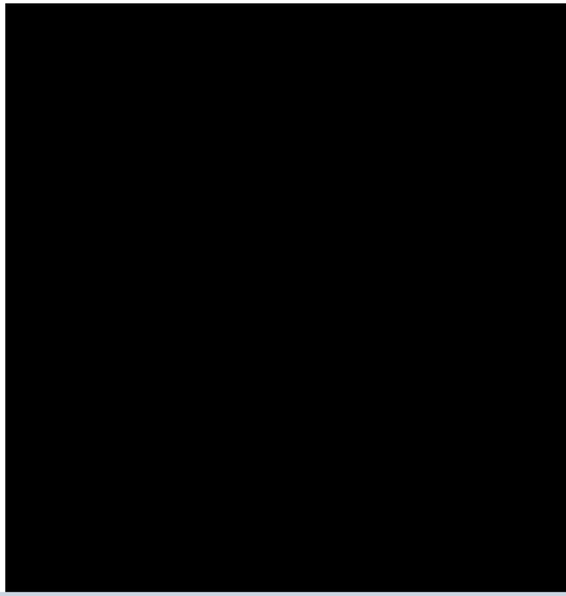
$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$$

Standard Deviation

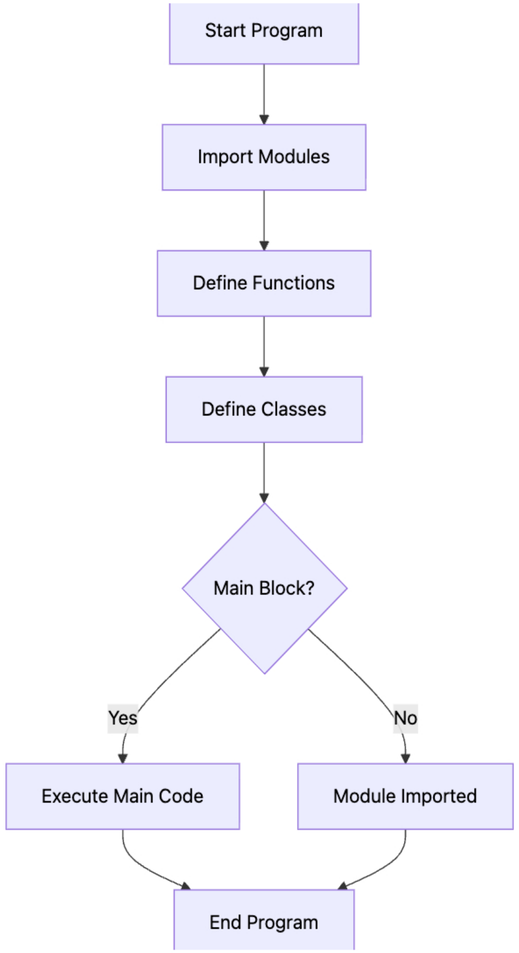
$$\sigma = \sqrt{\sigma^2}$$

Normal Distribution

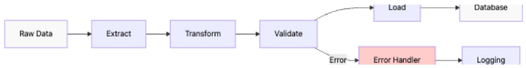
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



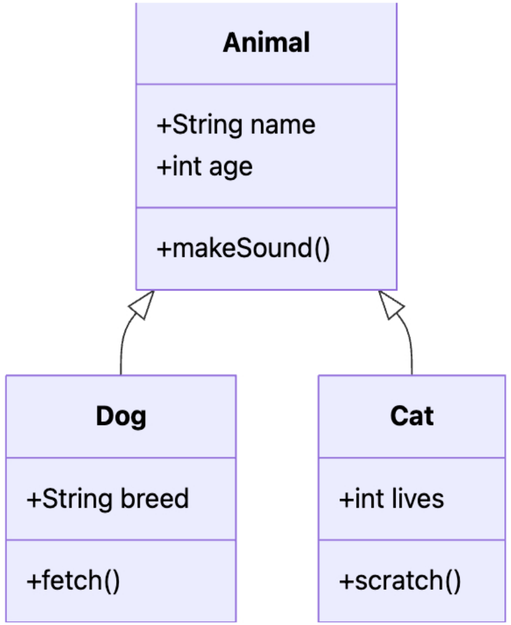
Python Program Flow



Data Processing Pipeline



Object-Oriented Hierarchy



Thank You

- Practice regularly
- Build real projects
- Join the community
- Never stop learning
- Share your knowledge