

# Python List Comprehensions

## Basic List Comprehensions

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

# List comprehension (preferred)
squares = [x**2 for x in range(10)]
print("Squares:", squares)

# Another example
cubes = [x**3 for x in range(1, 6)]
print("Cubes:", cubes)

# String operations
words = ['hello', 'world', 'python']
uppercase = [word.upper() for word in words]
print("\nUppercase:", uppercase)

lengths = [len(word) for word in words]
print("Lengths:", lengths)
```

## List Comprehensions with Conditions

```
# Filter even numbers
evens = [x for x in range(20) if x % 2 == 0]
print("Even numbers:", evens)

# Get even squares
even_squares = [x**2 for x in range(10) if x % 2 == 0]
print("Even squares:", even_squares)

# Multiple conditions
filtered = [x for x in range(20) if x % 2 == 0 and x > 10]
print("\nEven and > 10:", filtered)

# If-else in comprehension
labels = ['even' if x % 2 == 0 else 'odd' for x in range(10)]
print("\nLabels:", labels)

# Filter strings
words = ['apple', 'banana', 'cherry', 'date']
long_words = [w for w in words if len(w) > 5]
print("\nLong words:", long_words)
```

## Nested List Comprehensions

```
# Flatten 2D list
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
flattened = [num for row in matrix for num in row]
print("Flattened:", flattened)

# Create multiplication table
table = [[i * j for j in range(1, 6)] for i in range(1, 6)]
print("\nMultiplication table:")
for row in table:
    print(row)
```

```

# Combine two lists
colors = ['red', 'blue']
objects = ['ball', 'car']
combinations = [f"{color} {obj}" for color in colors for obj in objects]
print("\nCombinations:", combinations)

```

## Advanced Patterns

```

# Dictionary to list transformation
person = {'name': 'Alice', 'age': 25, 'city': 'NYC'}
pairs = [f'{k}: {v}' for k, v in person.items()]
print("Key-value pairs:", pairs)

# Process with enumerate
words = ['apple', 'banana', 'cherry']
indexed = [f'{i}: {word}' for i, word in enumerate(words)]
print("\nIndexed:", indexed)

# Nested conditions
numbers = range(1, 21)
special = [x for x in numbers
           if x % 2 == 0
           if x % 3 == 0]
print("\nDivisible by 2 and 3:", special)

# Multiple operations
data = [1, 2, 3, 4, 5]
transformed = [x**2 if x % 2 == 0 else x**3 for x in data]
print("\nSquare evens, cube odds:", transformed)

```

## Practical Applications

```

# Extract specific data
students = [
    {'name': 'Alice', 'grade': 85},
    {'name': 'Bob', 'grade': 92},
    {'name': 'Charlie', 'grade': 78}
]

names = [s['name'] for s in students if s['grade'] > 80]
print("High scorers:", names)

# Parse strings
text = "1,2,3,4,5"
numbers = [int(x) for x in text.split(',')]
print("\nParsed numbers:", numbers)

# Remove duplicates (preserving order)
items = [1, 2, 2, 3, 4, 3, 5, 1]
seen = set()
unique = [x for x in items if not (x in seen or seen.add(x))]
print("\nUnique (ordered):", unique)

# Create coordinate pairs
x_coords = [1, 2, 3]
y_coords = [4, 5, 6]
points = [(x, y) for x, y in zip(x_coords, y_coords)]
print("\nPoints:", points)

```