

## 1. First 10 Square Numbers

Using a loop:

```
squares = []
for i in range(1, 11):
    squares.append(i**2)
print(squares)  # [1, 4, 9, ..., 100]
```

Using list comprehension:

```
squares = [i**2 for i in range(1, 11)]
print(squares)
```

## 2. Remove Duplicates (Preserve Order)

```
def remove_duplicates(lst):
    seen = set()
    result = []
    for item in lst:
        if item not in seen:
            seen.add(item)
            result.append(item)
    return result

print(remove_duplicates([1, 2, 2, 3, 1, 4]))  # [1, 2, 3, 4]
```

## 3. Slicing Logic

- **Reverse a list:**

```
my_list = [1, 2, 3, 4, 5]
reversed_list = my_list[::-1]
print(reversed_list)  # [5, 4, 3, 2, 1]
```

- **Extract every third element:**

```
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]
third_elements = my_list[::3]
print(third_elements)  # [1, 4, 7]
```

## 4. Matrix Representation

```
matrix = [
```

```
[1, 2, 3],  
[4, 5, 6],  
[7, 8, 9]  
]  
diagonal = [matrix[i][i] for i in range(3)]  
print(diagonal)  # [1, 5, 9]
```

## 5. Sorting Challenge

```
names = ["Alice", "Bob", "Charlie"]  
scores = [85, 90, 80]  
  
paired = list(zip(names, scores))  
paired.sort(key=lambda x: x[1], reverse=True)  
  
for name, score in paired:  
    print(name, score)
```

## 6. Unpacking

```
data = [10, 20, 30, 40]  
first, *rest = data  
print(first)  # 10  
print(rest)   # [20, 30, 40]
```

**Why useful?**

In data processing, the first element could be a **timestamp, header, or identifier**, and the rest could be **data values**.

## 7. Flatten a Nested List

```
nested = [[1, 2], [3, 4]]  
flat = [item for sublist in nested for item in sublist]  
print(flat)  # [1, 2, 3, 4]
```

## 8. List vs Set

```
my_list = [1, 2, 2, 3]  
unique_set = set(my_list)  
print(unique_set)  # {1, 2, 3} – duplicates removed, but  
order is lost  
Lost: Order of elements.
```

## 9. Memory Management

```
list1 = [1, 2, 3]
list2 = list1          # Same memory reference
list3 = list1.copy()   # New memory reference

list2.append(4)
print(list1)  # [1, 2, 3, 4] – affects list1
print(list3)  # [1, 2, 3] – unaffected
```

## 10. AI Applications – Tokenization & Stopword Removal

```
sentence = "This is a simple example for NLP"
stopwords = {"is", "a", "for"}

# Tokenization
tokens = sentence.lower().split()

# Remove stopwords
filtered = [word for word in tokens if word not in stopwords]
print(filtered)  # ['this', 'simple', 'example', 'nlp']
```

If you like, I can make **all 10 solutions into one Python file** so you can run them together as a single reference script. That way you have them in one place.