

Copilot

Creating an exhaustive list of iteration examples can be quite extensive, but here's a diverse set of examples that demonstrate different iteration scenarios in the context of prompts for an AI copilot:

1. Basic Loop Iteration

```
# Prompt: "Print numbers from 1 to 5."
for i in range(1, 6):
    print(i)
```

2. While Loop

```
# Prompt: "Continue prompting the user until they type 'exit'."
user_input = ''
while user_input.lower() != 'exit':
    user_input = input("Enter text (type 'exit' to quit): ")
```

3. Iterating Over a List

```
# Prompt: "Convert all strings in the list to uppercase."
strings = ['hello', 'world', 'copilot']
uppercase_strings = [s.upper() for s in strings]
```

4. Nested Loops

```
# Prompt: "Create a multiplication table from 1 to 3."
for i in range(1, 4):
    for j in range(1, 4):
        print(f"{i} * {j} = {i*j}")
```

5. Iterating with Conditions

```
# Prompt: "Print only even numbers from 1 to 10."
for i in range(1, 11):
    if i % 2 == 0:
        print(i)
```

6. Dictionary Iteration

```
# Prompt: "Print each key-value pair in the dictionary."
my_dict = {'a': 1, 'b': 2, 'c': 3}
for key, value in my_dict.items():
    print(f"Key: {key}, Value: {value}")
```

7. File Iteration

```
# Prompt: "Read a file line by line and print each line."
with open('myfile.txt', 'r') as file:
    for line in file:
        print(line.strip())
```

8. Function Iteration

```
# Prompt: "Apply a function to all elements in a list."
def square(x):
    return x * x
numbers = [1, 2, 3, 4, 5]
squared_numbers = [square(x) for x in numbers]
```

9. Generator Iteration

```
# Prompt: "Create a generator that yields numbers from 1 to 3."
def number_generator():
    for i in range(1, 4):
        yield i
for number in number_generator():
    print(number)
```

10. Iterating Over Multiple Collections Simultaneously

```
# Prompt: "Zip two lists and print pairs."
list1 = [1, 2, 3]
list2 = ['a', 'b', 'c']
for item1, item2 in zip(list1, list2):
    print(f"Pair: {item1}, {item2}")
```

These examples cover a range of iteration techniques, from simple loops to more complex structures involving conditions, nested loops, and generators. They can serve as a starting point for various tasks that might be prompted to an AI copilot.