```python
# Demonstrates multiple (identical) function calls

print("meow")
print("meow")
print("meow")
```

```python
# Demonstrates a while loop, counting down

i = 3
while i != 0:
    print("meow")
    i = i - 1
```

```
# Demonstrates a while loop, counting up from 1

i = 1
while i <= 3:
    print("meow")
    i = i + 1
```

```
# Demonstrates a while loop, counting up from 0

i = 0
while i < 3:
    print("meow")
    i = i + 1
```

```python
# Demonstrates (more succinct) incrementation

i = 0
while i < 3:
    print("meow")
    i += 1
```

```python
# Demonstrates a for loop, using a list

for i in [0, 1, 2]:
    print("meow")
```

```python
# Demonstrates a for loop, using range

for i in range(3):
    print("meow")
```

```python
# Demonstrates a for loop, with _ as a variable

for _ in range(3):
    print("meow")
```

```
# Demonstrates str multiplication

print("meow\n" * 3, end="")
```

```python
# Demonstrates validation of user's input

while True:
    n = int(input("What's n? "))
    if n > 1:
        break

for _ in range(n):
    print("meow")
```

```python
# Demonstrates defining functions


def main():
    meow(get_number())


def get_number():
    while True:
        n = int(input("What's n? "))
        if n > 1:
            return n


def meow(n):
    for _ in range(n):
        print("meow")


main()
```

```python
# Demonstrates indexing into a list

students = ["Hermione", "Harry", "Rob"]

print(students[0])
print(students[1])
print(students[2])
```

```python
# Demonstrates iterating over a list

students = ["Hermione", "Harry", "Rob"]

for student in students:
    print(student)
```

```python
# Demonstrates iterating over and indexing into a list

students = ["Hermione", "Harry", "Rob"]

for i in range(len(students)):
    print(i + 1, students[i])
```

```python
# Demonstrates indexing into a dict

students = {
    "Hermione": "Gryffindor",
    "Harry": "Gryffindor",
    "Rob": "Gryffindor",
    "Draco": "Slytherin",
}

print(students["Hermione"])
print(students["Harry"])
print(students["Rob"])
print(students["Draco"])
```

```python
# Demonstrates iterating over and index into a dict

students = {
    "Hermione": "Gryffindor",
    "Harry": "Gryffindor",
    "Rob": "Gryffindor",
    "Draco": "Slytherin",
}

for student in students:
    print(student, students[student], sep=", ")
```

```python
# Demonstrates iterating over a list of dict objects

students = [
    {"name": "Hermione", "house": "Gryffindor", "patronus": "Otter"},
    {"name": "Harry", "house": "Gryffindor", "patronus": "Stag"},
    {"name": "Ron", "house": "Gryffindor", "patronus": "Jack Russell terrier"},
    {"name": "Draco", "house": "Slytherin", "patronus": None},
]

for student in students:
    print(student["name"], student["house"], student["patronus"], sep=", ")
```

```python
# Prints a column of bricks

print("#")
print("#")
print("#")
```

```python
# Prints column of bricks using a loop

for _ in range(3):
    print("#")
```

```python
# Prints column of bricks using a function with a loop


def main():
    print_column(3)


def print_column(height):
    for _ in range(height):
        print("#")


main()
```

```python
# Prints column of bricks using a function with str multiplication


def main():
    print_column(3)


def print_column(height):
    print("#\n" * height, end="")


main()
```

```python
# Prints row of coins using a function with str multiplication


def main():
    print_row(4)


def print_row(width):
    print("?" * width)


main()
```

```python
# Prints square of bricks using a function with nested loops


def main():
    print_square(3)


def print_square(size):
    for i in range(size):
        for j in range(size):
            print("#", end="")
        print()


main()
```

```python
# Prints square of bricks using a function with a loop and str multiplication


def main():
    print_square(3)


def print_square(size):
    for _ in range(size):
        print("#" * size)


main()
```

```python
# Prints square of bricks using a function with a loop and str multiplication


def main():
    print_square(3)


def print_square(size):
    for _ in range(size):
        print_row(size)


def print_row(width):
    print("#" * width)


main()
```