# Demonstrates multiple (identical) function calls

```python
print("meow")
print("meow")
print("meow")
```
# Demonstrates a while loop, counting down

```python
temp = 3
while temp != 0:
    print("meow")
    temp = temp - 1
```
# Demonstrates a while loop, counting up from 1

```python
i = 1
while i <= 3:
    print("meow")
    i = i + 1
```
# Demonstrates a while loop, counting up from 0

```python
i = 0
while i < 3:
    print("meow")
    i = i + 1
```
# Demonstrates (more succinct) incrementation

```python
i = 0
while i < 3:
    print("meow")
    i += 1
```
# Demonstrates a for loop, using a list

```python
for i in [0, 1, 2):
    print("meow")
```
# Demonstrates a for loop, using range

```python
for i in range(3):
    print("meow")
```
# Demonstrates a for loop, with _ as a variable

```python
for _ in range(3):
    print("meow")
```
# Demonstrates str multiplication

print("meow\n" * 3, end="")
# Introduces continue, break

```python
while True:
    n = int(input("What's n? "))
    if n <= 0:
        continue
    else:
        break

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

```python
while True:
    n = int(input("What's n? "))
    if n > 0:
        break

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

# 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()
# Demonstrates indexing into a list

```python
students = ["Hermione", "Harry", "Ron"]

print(students[0])
print(students[1])
print(students[2])
```
# Demonstrates iterating over a list

```python
students = ['Hermione', 'Harry', 'Ron']

for student in students:
    print(student)
```
# Demonstrates iterating over and indexing into a list

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

for i in range(len(students)):
    print(i + 1, students[i])
# Demonstrates indexing into a dict

```python
students = {
    "Hermione": "Gryffindor",
    "Harry": "Gryffindor",
    "Ron": "Gryffindor",
    "Draco": "Slytherin",
}

print(students["Hermione"])  
print(students["Harry"])  
print(students["Ron"])  
print(students["Draco"])  
```
# Demonstrates iterating over and index into a dict

```python
students = {
    "Hermione": "Gryffindor",
    "Harry": "Gryffindor",
    "Ron": "Gryffindor",
    "Draco": "Slytherin",
}

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

# Demonstrates iterating over a list of dict objects

```python
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=",")
```
# Prints a column of bricks

print("#")
print("#")
print("#")
# Prints column of bricks using a loop

```python
for _ in range(3):
    print("#")
```
def main():
    print_column(3)

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

main()
# Prints column of bricks using a function with str multiplication

def main():
    print_column(3)

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

main()
# Prints row of coins using a function with str multiplication

def main():
    print_row(4)

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

main()
# 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()
# 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()
# 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()