Loops

Recap of Decision Making

Decision making helps the computer decide what to do based on certain conditions.

Condition:

Give chocolate to friend, if available in Plate

Yayyy, It's Your Birthday

Introducing Loops as Repeated Decisions:

Now there are 100 Friends.....

Condition: Until chocolates are available in the plate, Distribute them

Loops

Loops are used to execute a block of code repeatedly as long as a certain condition is true or for a specific number of iterations

Types

- while
- for

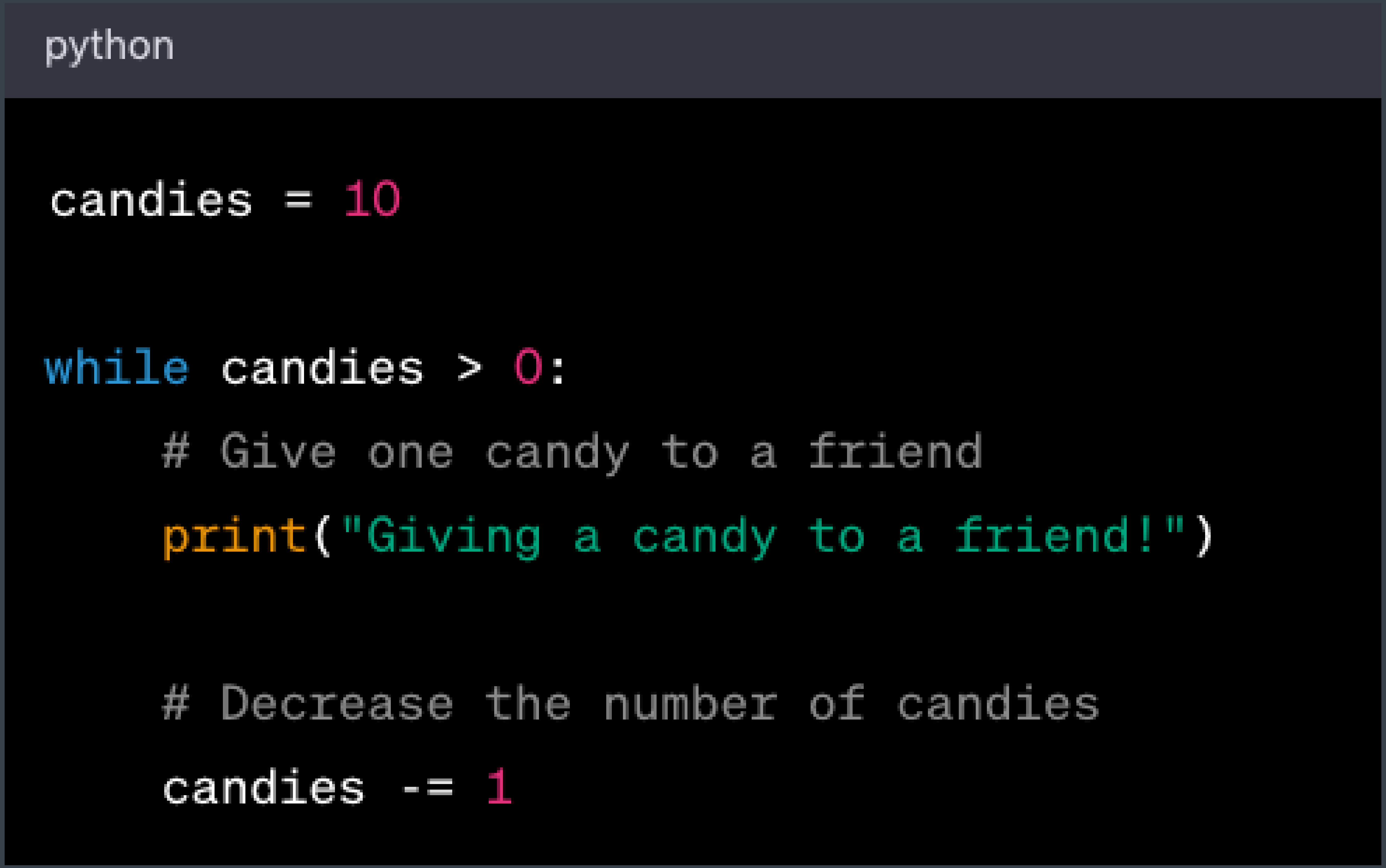
while loop

The while loop executes a block of code as long as a specified condition is true. It continuously checks the condition before each iteration and stops when the condition becomes false.

Syntax:

```
while condition:
    # Code block to be executed repeatedly
```

while loop



python

```python
candies = 10

while candies > 0:
    # Give one candy to a friend
    print("Giving a candy to a friend!")

    # Decrease the number of candies
    candies -= 1
```

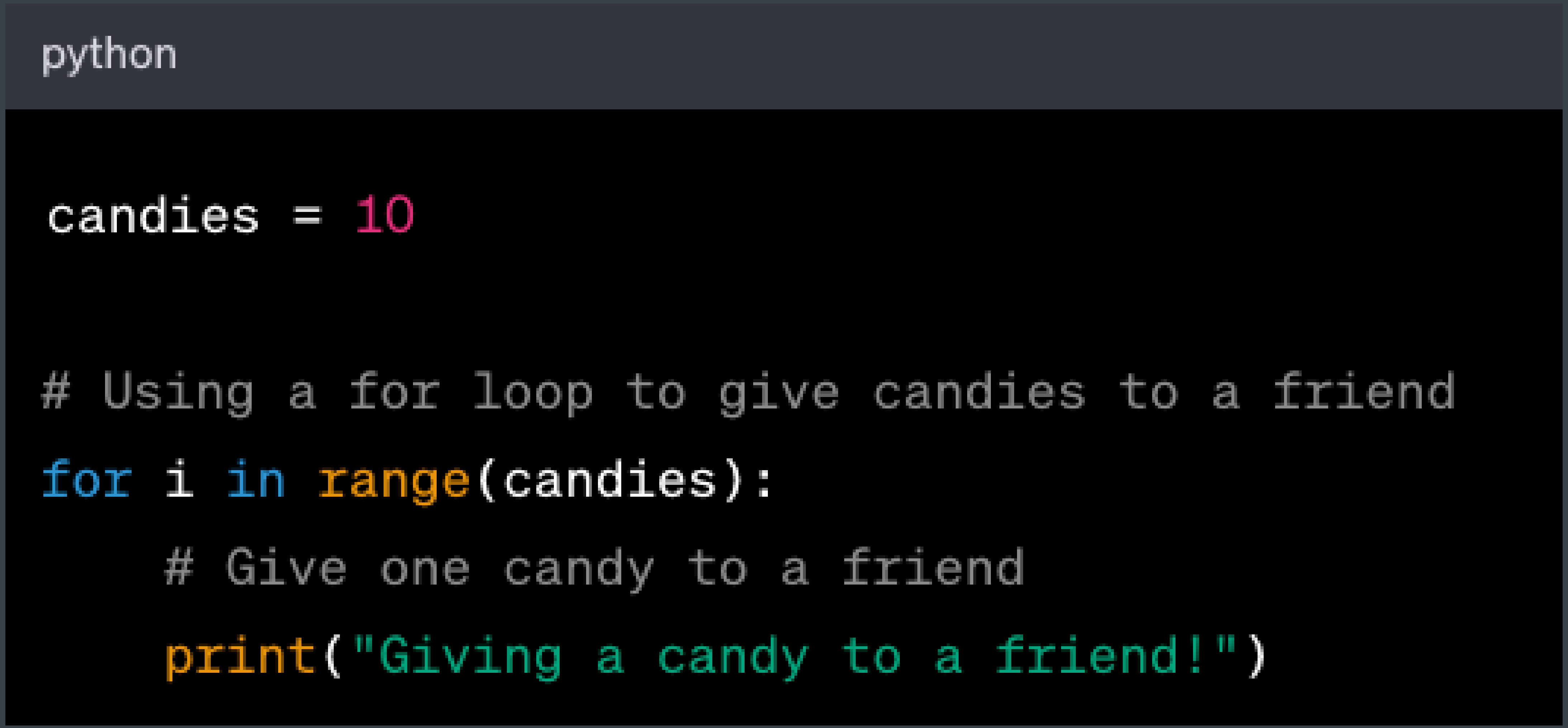
for loop

A for loop is a way to repeat a block of code for each item in a collection (like a list) or for a specific range of numbers.

Syntax:

```
for variable in range(start, stop, step):
    # Code block to be executed for each variable
```

for loop



python

```python
candies = 10

# Using a for loop to give candies to a friend
for i in range(candies):
    # Give one candy to a friend
    print("Giving a candy to a friend!")
```

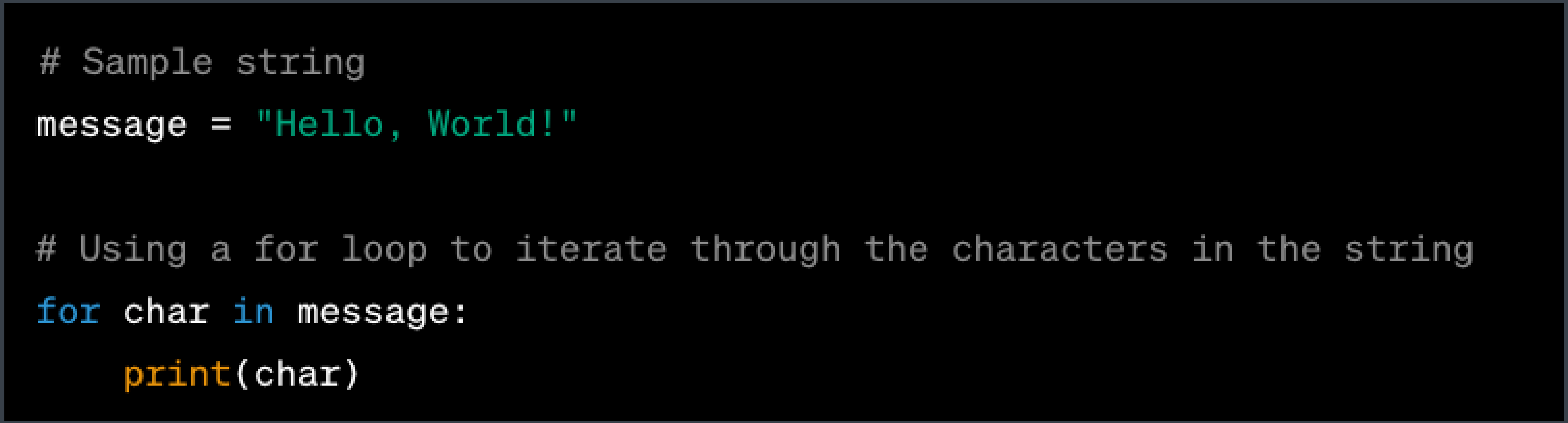
for loop for Sequence

The for loop is used to iterate over a sequence (such as a list, tuple, string, or dictionary) and execute a block of code for each item in the sequence.

Syntax:

```
for item in sequence:
    # Code block to be executed for each item
```

Example:



```python
# Sample string
message = "Hello, World!"

# Using a for loop to iterate through the characters in the string
for char in message:
    print(char)
```

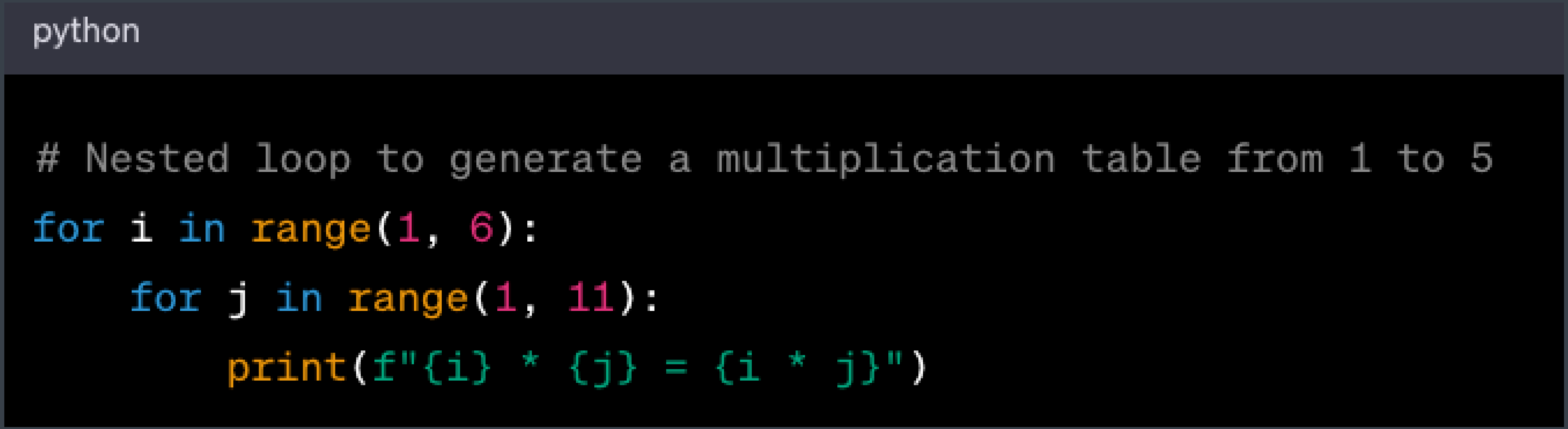
Nested loops

Nested loops refer to the situation where one loop is placed inside another loop. This allows you to execute a set of instructions repeatedly

Syntax:

```
for outer_var in outer_sequence:
    # Code block of the outer loop
    for inner_var in inner_sequence:
        # Code block of the inner loop
```

Nested loops



python

```python
# Nested loop to generate a multiplication table from 1 to 5
for i in range(1, 6):
    for j in range(1, 11):
        print(f"{i} * {j} = {i * j}")
```

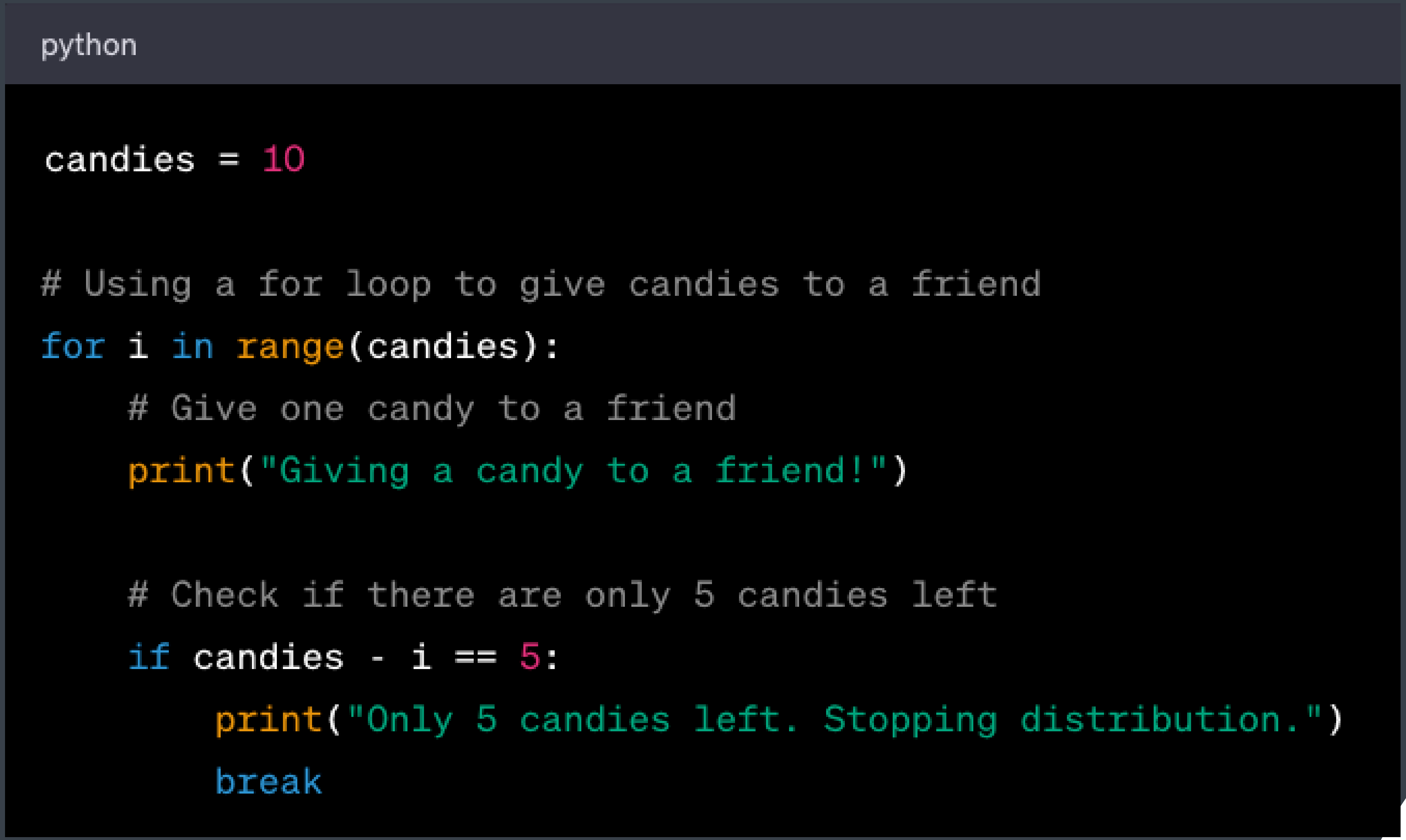
Break

If during the execution of the loop Python interpreter encounters break, it immediately stops the loop execution and exits out of it.

Syntax:

```
while condition:
    # Code block inside the loop
    if some_condition:
        break  # Exit the loop if the condition is met
```

Break



python

```python
candies = 10

# Using a for loop to give candies to a friend
for i in range(candies):
    # Give one candy to a friend
    print("Giving a candy to a friend!")

    # Check if there are only 5 candies left
    if candies - i == 5:
        print("Only 5 candies left. Stopping distribution.")
        break
```

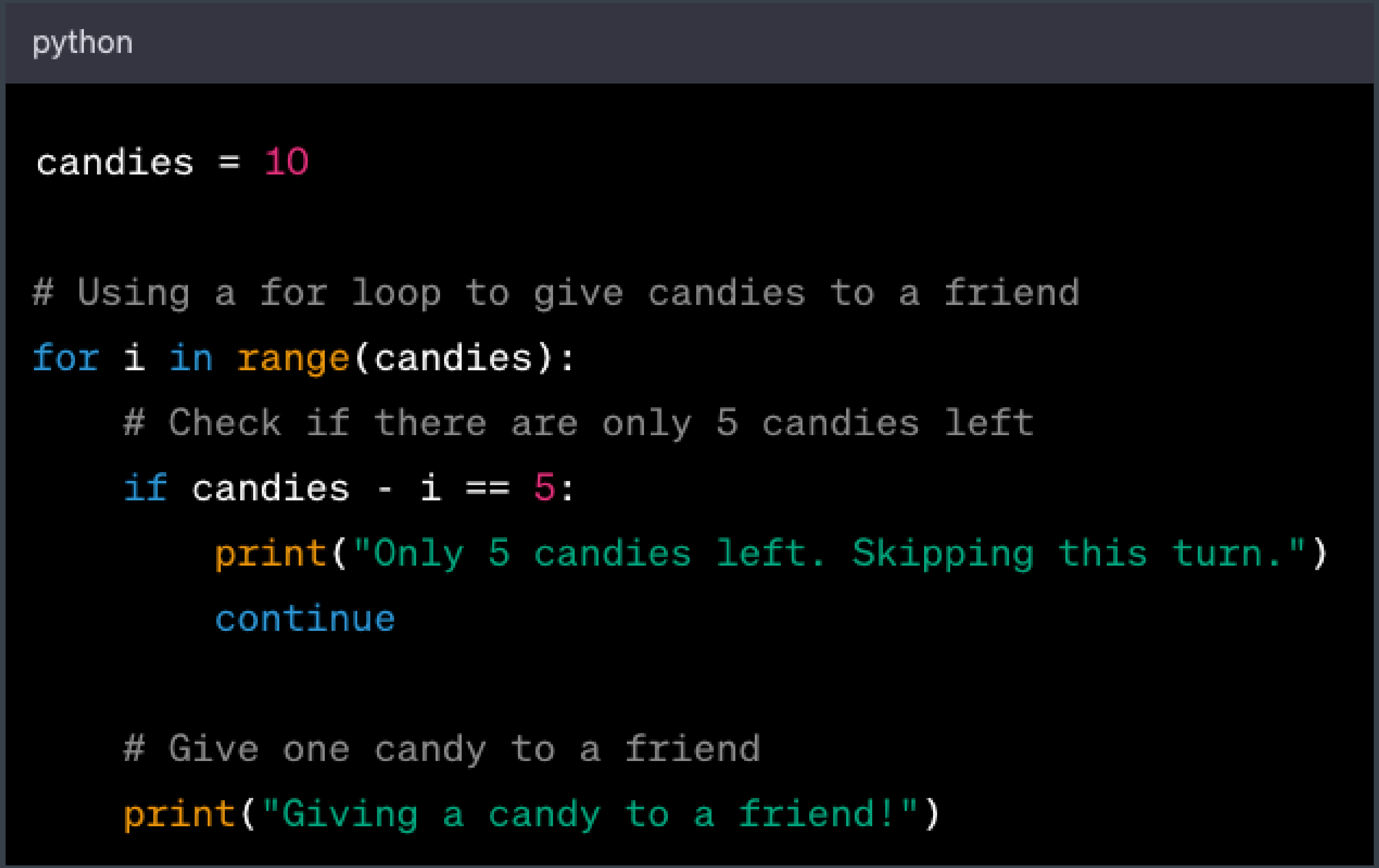
Continue

Continue statement is used to skip the rest of the current iteration in a loop and move to the next iteration immediately.

Syntax:

```
while condition: # Code block inside the loop
    if some_condition:
        continue    #skip this iteration
```

Continue



python

```python
candies = 10

# Using a for loop to give candies to a friend
for i in range(candies):
    # Check if there are only 5 candies left
    if candies - i == 5:
        print("Only 5 candies left. Skipping this turn.")
        continue

    # Give one candy to a friend
    print("Giving a candy to a friend!")
```

Problems On

Loops + Strings + Numbers + Decision Making

Print numbers from 1 to N

Take a positive integer N as input and print all the numbers from 1 to N.

Sample Input: N = 5

Sample Output:

```
1
2
3
4
5
```

Calculate the sum of N natural numbers

Take a positive integer N as input and calculate the sum of the first N natural numbers.

Sample Input: N = 5

Sample Output: Sum of first 5 natural numbers: 15

Print even numbers from 1 to N

Take a positive integer N as input and print all the even numbers from 1 to N.

Sample Input: N = 10

Sample Output:

```
2
4
6
8
10
```

Print odd numbers from 1 t num

Take a positive integer N as input and print all the odd numbers from 1 to N.

Sample Input: N = 10

Sample Output:

```
1
3
5
7
9
```

Multiplication table of a number

Take a positive integer N as input and print the multiplication table of N from 1 to 10.

Sample Input:

N = 3

Sample Output:

```
Multiplication table of 3:
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

Calculate the factorial of a number

Take a positive integer N as input and calculate its factorial (N!).

Sample Input:	N = 5
Sample Output:	Factorial of 5: 120