



Relative Reassignment Operators and Practice with while Loops

Announcements

EX02 (Wordle) due *tonight* at 11:59pm

- Need help? Please visit Office Hours or Tutoring!

Review of a common problem: the dreaded *infinite loop*

If a condition in a `while` loop never becomes False, the loop will continue indefinitely.

To prevent this:

- Ensure that your loop's condition will eventually be False!

```
1 def count_to_n(n: int) -> None:
2     count: int = 0
3     while count <= n:
4         print(f"Count is: {count}")
5         count = count + 1
6
7
8 count_to_n(n=4)
```

Review of a common problem: the dreaded *infinite loop*

If a condition in a `while` loop never becomes False, the loop will continue indefinitely.

To prevent this:

- Ensure that your loop's condition will eventually be False!

Which line of code in the repeat block prevents an *infinite loop* from occurring?
Keep this in mind as you finish **Wordle**!

```
1 def count_to_n(n: int) -> None:
2     count: int = 0
3     while count <= n:
4         print(f"Count is: {count}")
5         count = count + 1
6
7
8 count_to_n(n=4)
```

Warm-Up: Memory Diagram

```
1  """A countdown program..."""
2
3
4  def main() -> None:
5      seconds: int = 3
6      countdown(seconds)
7      print(f"main {seconds}")
8
9
10 def countdown(seconds: int) -> None:
11     print("T minus")
12     while seconds > 0:
13         print(seconds)
14         seconds = seconds - 1
15
16     print(f"countdown {seconds}")
17
18
19 main()
```

```
1     """A countdown program..."""
2
3
4 def main() -> None:
5     seconds: int = 3
6     countdown(seconds)
7     print(f"main {seconds}")
8
9
10    def countdown(seconds: int) -> None:
11        print("T minus")
12        while seconds > 0:
13            print(seconds)
14            seconds = seconds - 1
15
16        print(f"countdown {seconds}")
17
18
19    main()
```

Relative Reassignment Operators

It's *very* common to need to update the value of a variable, relative to its current value, e.g.:

```
count: int = 1
```

```
count = count + 1
```

Relative reassignment operators offer a shorthand way of doing this!

```
count += 1
```

Relative Reassignment Operators

```
1  """A countdown program..."""
2
3
4  def main() -> None:
5      seconds: int = 3
6      countdown(seconds)
7      print(f"main {seconds}")
8
9
10 def countdown(seconds: int) -> None:
11     print("T minus")
12     while seconds > 0:
13         print(seconds)
14         seconds = seconds - 1
15
16     print(f"countdown {seconds}")
17
18
19 main()
```

Try writing line 14 using a relative reassignment operator!

Your task: Convert this recursive function to one that uses a while loop!

```
def countdown_rec(n: int) -> None:
    if n < 0:
        print("Too late!")
    elif n == 0:
        print("Blast off!")
    else:
        print(n)
        countdown_rec(n - 1)
```

A nested while loop!

```
1  def triangle(n: int) -> None:
2      i: int = 1
3      line: str
4      while i <= n:
5          line = ""
6          while len(line) < i:
7              line += "*"
8          print(line)
9          i += 1
10
11
12 triangle(2)
```

More practice: Convert this recursive function to one that uses a while loop!

```
def rec_icarus(x: int) -> int:
    if x < 0:
        return -1
    elif x < 2:
        print(f"x is {x}")
        return rec_icarus(x + 1)
    else:
        return x
```