COMP110

Relative Reassignment Operators and Practice with while Loops

Announcements

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

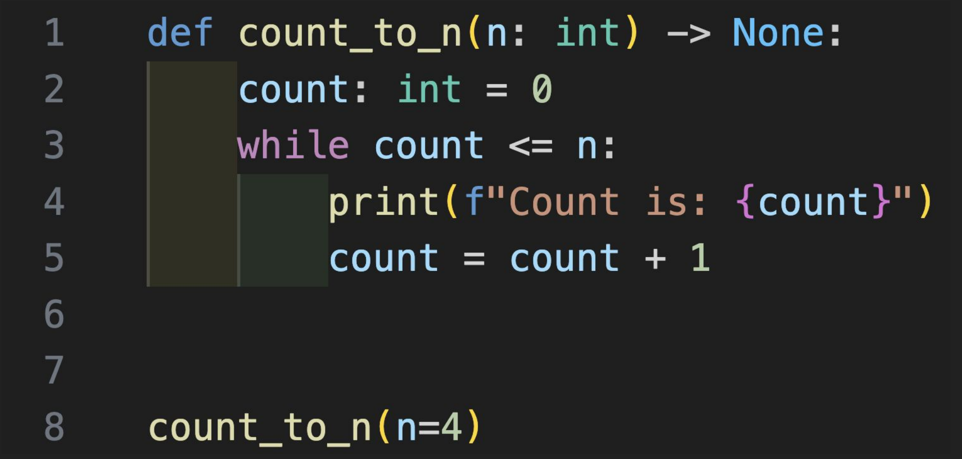
- Need help? Please visit Office Hours or Tutoring!

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!



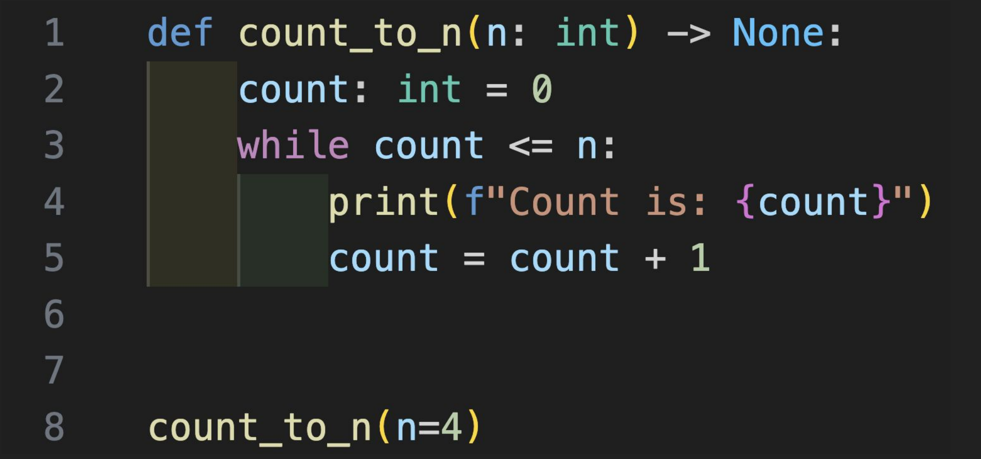
```python
def count_to_n(n: int) -> None:
    count: int = 0
    while count <= n:
        print(f"Count is: {count}")
        count = count + 1


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:

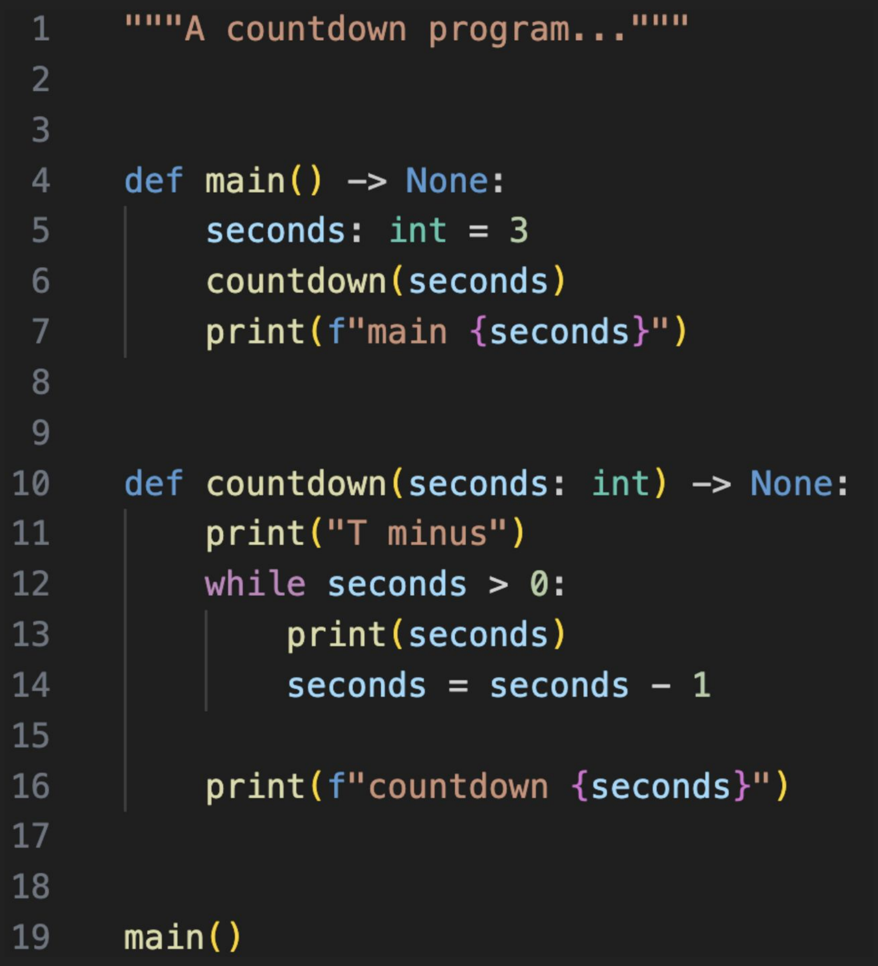
- Ensure that your loop's condition will eventually be False!



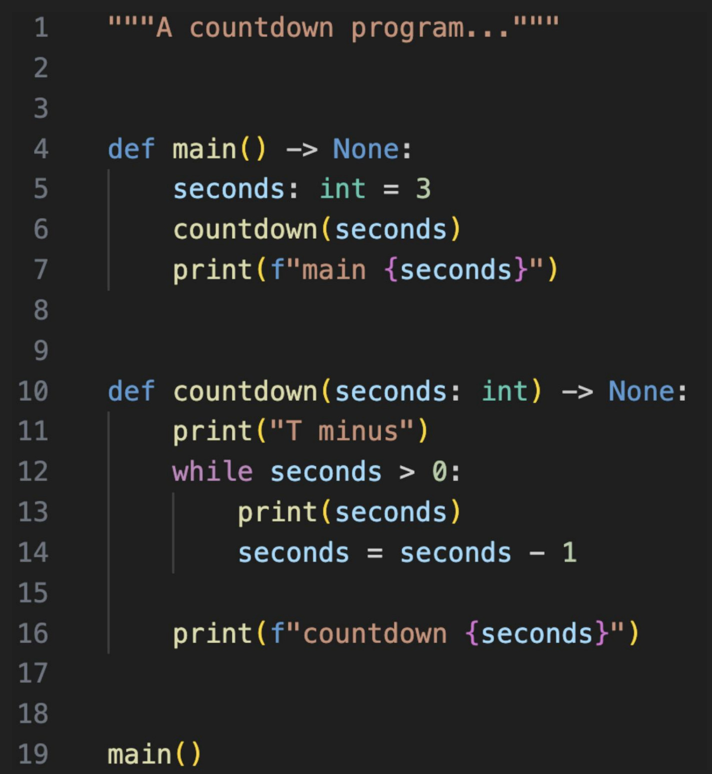
```python
def count_to_n(n: int) -> None:
    count: int = 0
    while count <= n:
        print(f"Count is: {count}")
        count = count + 1


count_to_n(n=4)
```

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

Warm-Up: Memory Diagram



```python
"""A countdown program..."""


def main() -> None:
    seconds: int = 3
    countdown(seconds)
    print(f"main {seconds}")


def countdown(seconds: int) -> None:
    print("T minus")
    while seconds > 0:
        print(seconds)
        seconds = seconds - 1

    print(f"countdown {seconds}")


main()
```



```python
"""A countdown program..."""


def main() -> None:
    seconds: int = 3
    countdown(seconds)
    print(f"main {seconds}")


def countdown(seconds: int) -> None:
    print("T minus")
    while seconds > 0:
        print(seconds)
        seconds = seconds - 1

    print(f"countdown {seconds}")


main()
```

Relative Reassignment Operators

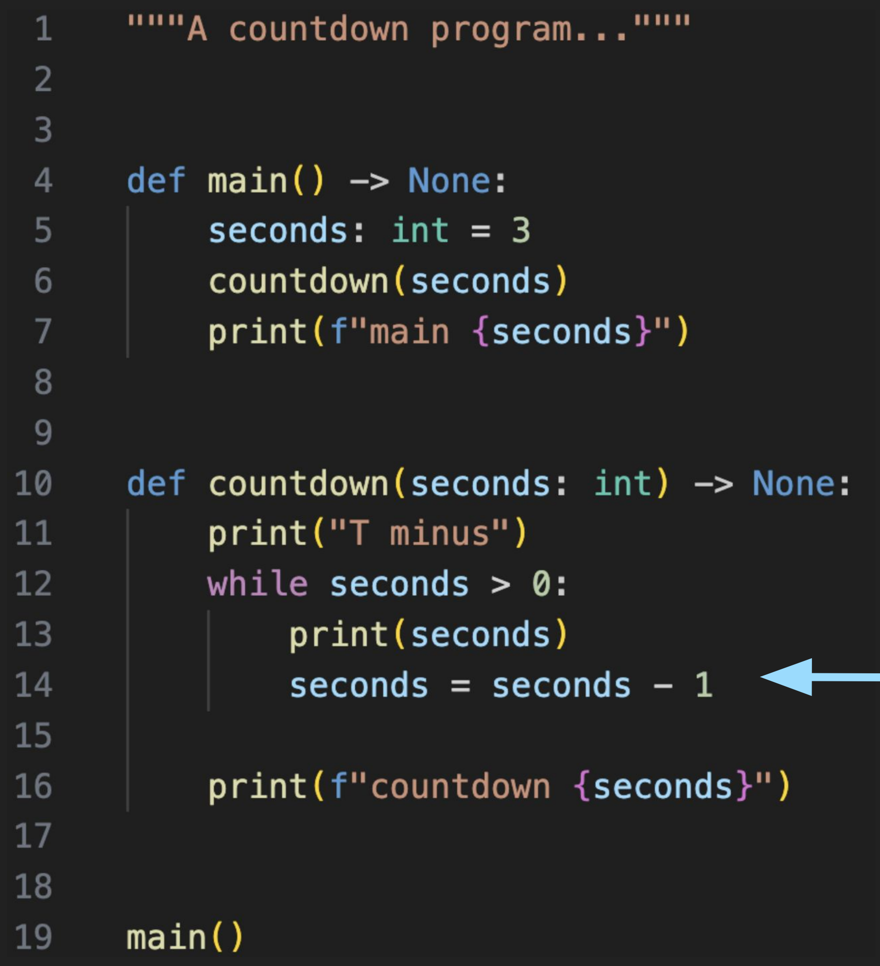
It’s *very* common to need to update the value of a variable, relative to its current value, e.g.:

```
count: int = 1
count = count + 1
```

Relative reassignment operators offer a shorthand way of doing this!

```
count += 1
```

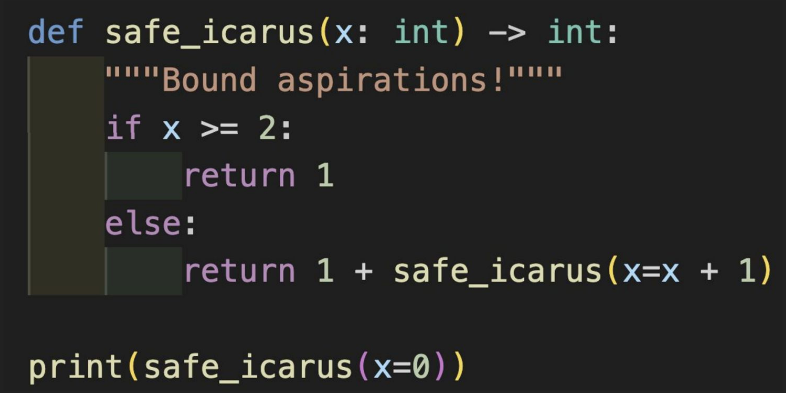
Relative Reassignment Operators



```
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()
```

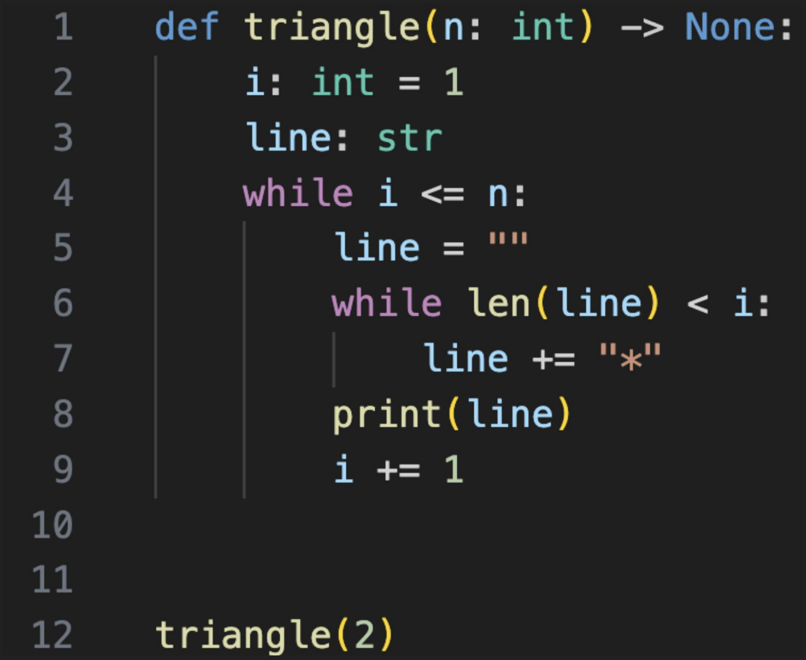
Try writing line 14 using a relative reassignment operator!

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



```python
def safe_icarus(x: int) -> int:
    """Bound aspirations!"""
    if x >= 2:
        return 1
    else:
        return 1 + safe_icarus(x=x + 1)

print(safe_icarus(x=0))
```

A nested while loop!



```python
def triangle(n: int) -> None:
    i: int = 1
    line: str
    while i <= n:
        line = ""
        while len(line) < i:
            line += "*"
        print(line)
        i += 1


triangle(2)
```