



Lecture 3: Loops

CSE 160

Summer 2026

Questions?



Announcements

- [Homework Assignment 0](#) is due tonight at 11:59pm
- [Written Check-in 1](#) is due tonight at 11:59pm
- [Homework Assignment 1](#) is due Monday, July 6 at 11:59pm

Today's Roadmap

1. Homework 1 Preview
2. Expressions and Variables Recap
3. Loops
4. `range()`
5. More Loops!

Homework 1 Preview

Homework 1 is due Monday, July 6 at 11:59 PM

cs.uw.edu/160

Today's Roadmap

1. Homework 1 Preview
- 2. Expressions and Variables Recap**
3. Loops
4. `range()`
5. More Loops!

Expressions and Variables (Recap)

Expressions are *evaluated* by Python to produce a **value**

- Order of operations matters!
- Not just arithmetic (e.g. "1" + "2" → "12")
- Every **value** in Python has a type (e.g. `int`, `str`, `bool`)

```
(1014 - 1) * 2
```

Variables *hold* **values**

- Use assignment statements to assign a **value** to a **variable**
- We can reference the value using the variable name
- Variable names are case sensitive!

```
my_num = 42
```

try this!

```
print("Favorite Num =", my_num)
```

Today's Roadmap

1. Homework 1 Preview
2. Expressions and Variables Recap
- 3. Loops**
4. `range()`
5. More Loops!

Loops

What if we want to repeat the same code multiple times?

Two Options:

“Hard Coding”

- Type out/copy all code
- Very slow to code
- What if we make a mistake?

Loops

- Write code for only one iteration
- Faster!
- Easier to spot/correct mistakes

Loops

What if we want to repeat the same code multiple times?

Two Options:

~~“Hard Coding”~~

- Type out/copy all code
- Very slow
- What if we make a mistake?

Loops

- Write code for only one iteration
- Faster!
- Easier to spot/correct mistakes

Anatomy of a For Loop

```
for i in range(10):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

Anatomy of a For Loop

for loop keyword

```
for i in range(10):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

Anatomy of a For Loop

loop variable

```
for i in range(10):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

Anatomy of a For Loop

controls loop

```
for i in range(10):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

Anatomy of a For Loop

```
for i in range(10):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

loop body

Anatomy of a For Loop

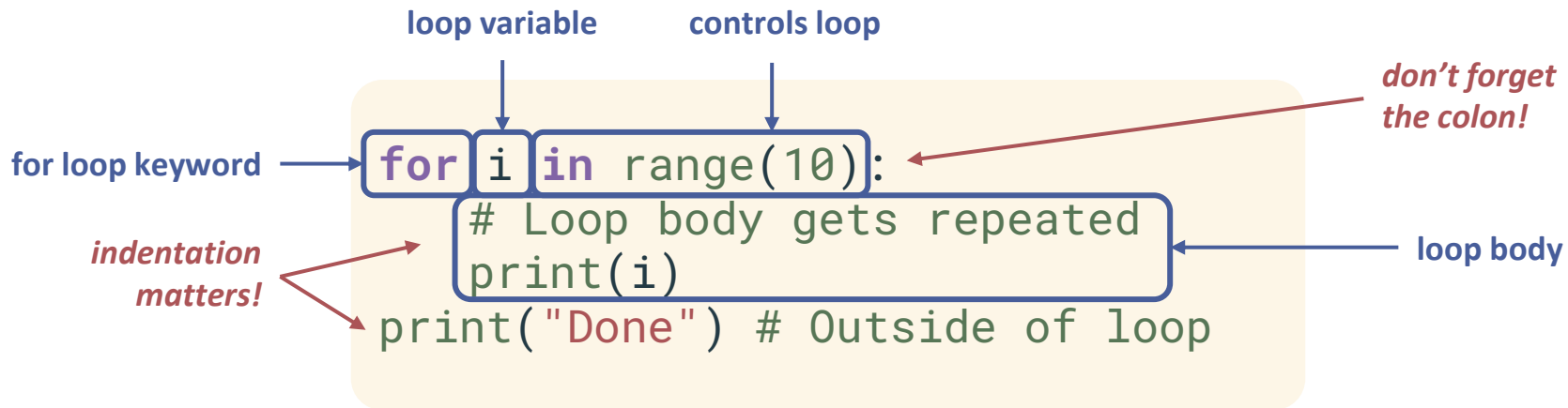
The diagram illustrates the components of a Python for loop. A yellow rounded rectangle contains the following code:

```
for i in range(10):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

Annotations with red arrows point to specific parts of the code:

- don't forget the colon!* points to the colon at the end of the first line.
- indentation matters!* points to the indentation of the second line.

Anatomy of a For Loop



What Happens Inside a Loop?

The **value** of the **loop variable** is updated each time through the loop:

```
for i in range(5):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

Python Tutor is a very helpful website for visualizing how Python code is executed!

Python Tutor

Today's Roadmap

1. Homework 1 Preview
2. Expressions and Variables Recap
3. Loops
- 4. range()**
5. More Loops!

range Function

range is a function (like **print**) that produces a sequence of **integers**.

- **range(i)** produces a sequence of integers from **0** (inclusive) to less than i
 - `range(5)` → 0, 1, 2, 3, 4
- **range(i, j)** produces a sequence of integers from **i** (inclusive) to less than j
 - `range(2, 5)` → 2, 3, 4
- **range(i, j, k)** produces a sequence of integers from **i** (inclusive) to less than j in increments of **k**
 - `range(1, 11, 2)` → 1, 3, 5, 7, 9

Think Pair Share

Which use of the **range** function will produce the following numbers?

10, 20, 30, 40, 50, 60, 70, 80, 90, 100



Sli.do

#cse160

A

`range(100)`

B

`range(10, 100)`

C

`range(10, 110)`

D

`range(10, 100, 10)`

E

`range(10, 110, 10)`

Today's Roadmap

1. Homework 1 Preview
2. Expressions and Variables Recap
3. Loops
4. `range()`
5. **More Loops!**

Accumulator Pattern

We can *accumulate* (sum, multiply) values while going through a for loop:

```
result = initial value
for i in range(5):
    result = updated result
use result
```

More Loops?

If range produces a sequence of numbers, can we just specify the numbers ourselves?

0, 1, 2, 3, 4

```
for i in range(5):  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```

Yes!

More Loops!

A **list** allows us to specify a sequence of **values**

A **list** is specified with square brackets:

```
for i in [10, 13, 7]:  
    # Loop body gets repeated  
    print(i)  
print("Done") # Outside of loop
```



10
13
7

Even More Loops!

It turns out we can *iterate* over a **lot** of things!

Lists

```
print("Some CSE 160 TAs:")  
for name in ["Kellen", "Dani", "Guy"]:  
    print(name)
```

Some CSE 160 TAs:
Kellen
Dani
Guy

Strings

```
for letter in "Adrian":  
    print(letter)
```

A
d
r
i
a
n

And more!