Lecture 6: While Loops

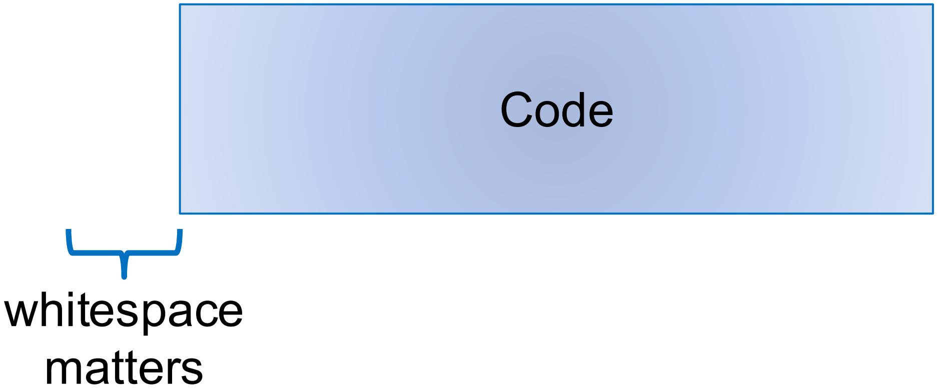
CS 50

February 5, 2026

Review: for loops

- When you want some set of statements to execute repeatedly . . . once for each element in a sequence.

```
for <var> in <sequence>:
```



Code

whitespace
matters

What about…

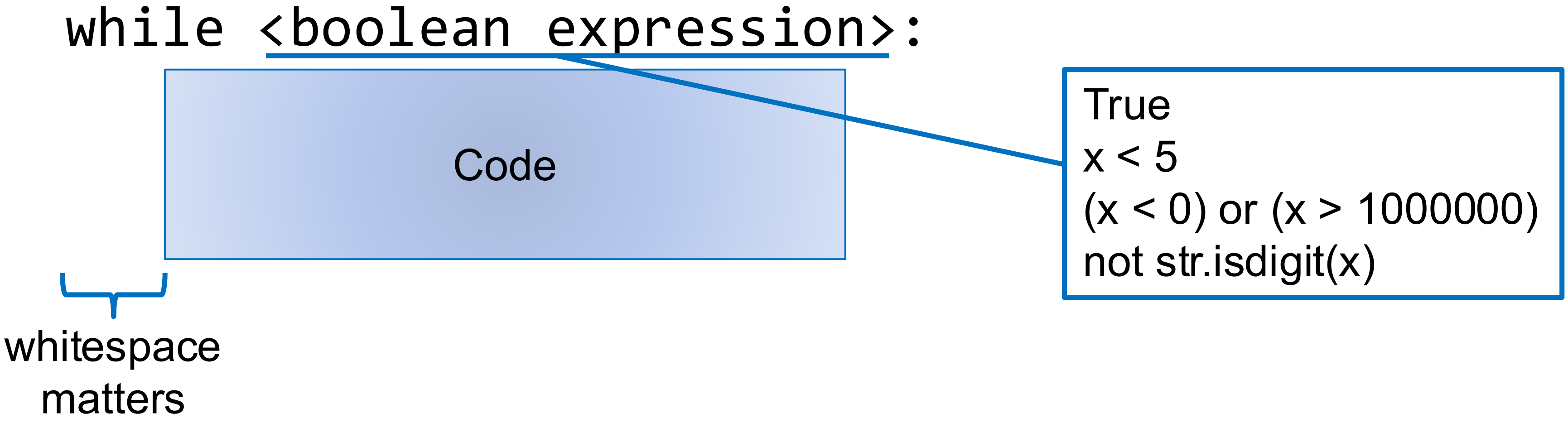
- What if we wanted a program that asks the user for a number between 1 and 10 and keeps asking until the user enters a number in the range [1, 10]?

- Example run:

```
Pick a number between 1 and 10:
   -10
Pick a number between 1 and 10:
   50
Pick a number between 1 and 10:
   7
Thanks!
```

while loops

- When you want some set of statements to execute repeatedly . . . until some stopping criteria is met.



```
while <boolean expression>:
```

Code

True
x < 5
(x < 0) or (x > 1000000)
not str.isdigit(x)

whitespace matters

Example 1: Using a while loop

- Define a function `get_num` that asks the user for a number between 1 and 10 and keeps asking until the user enters a number in the range [1, 10], then prints "Thanks!" and returns the requested number

```
x = get_num()
print(x)
```

```
Pick a number between 1 and 10:
   -10
Pick a number between 1 and 10:
   50
Pick a number between 1 and 10:
   7
Thanks!
7
```

Exercise 1

Define a function `authenticate` that prompts user for a password, repeating until the correct password is entered, then prints "Login Successful" and returns the number of attempts.

Assume that the correct password is "12345678"

```
count = authenticate()
print(str(count)+" attempts")
```

```
Enter your password:
    password
Enter your password:
    abc123
Enter your password:
    12345678
Login Successful
3 attempts
```

Exercise 2

Using an infinite loop, define a function `authenticate2` that prompts user for a password, repeating until the correct password is entered, then prints "Login Successful” and returns the number of attempts.

Assume that the correct password is "12345678"

```
count = authenticate()
print(str(count)+" attempts")
```

```
Enter your password:
    password
Enter your password:
    abc123
Enter your password:
    12345678
Login Successful
3 attempts
```

Example 2: Replacing for loops

```
def pyramid(height):
  string = ""
  for x in range(height):
    string = string + "*" * x + "\n"

 return string
```

```
full_string = pyramid(5)
print(3*full_string)
```

Exercise 3

Using a while loop, define a function `sum_odds` that takes one parameter `max` (an int) and then returns the sum of the odd values in the range [1, max]

For example, `sum_odds(5)` should return 9 (since 1 + 3 + 5 == 9)

Practice Problem

Define a function `sum_squares` that takes two arguments, a min and a max, both integers, and returns the sum of the squares between min and max. For example, `sum_squares(2, 4)` should return 29.

Then write a program that asks the user for a positive integer (and keeps asking until the integer is positive) then askes the user for a second integer (and keeps asking until the integer is positive) and then prints the sum of the squares between the two integers (using `sum_squares`).

Hint: it may be helpful to define a helper function (that gets a positive integer) so you don't have to copy the same code twice!