

PAYS 2023

INTRODUCTION TO PROGRAMMING USING PYTHON

6: while loops



Alexandra Papoutsaki

she/her/hers

Lecture 6: while loops and main function

- ▶ prime numbers
- ▶ while loops

Prime numbers

- ▶ A number that is only divisible by 1 and itself.
- ▶ How can we tell if a number is prime?
 - ▶ Try and divide it by all of the numbers between 1 and the number.
 - ▶ If none of them divide it evenly, then it's prime, otherwise it's not.
 - ▶ How can we check to see if a number divides our number evenly?
 - ▶ We can use the remainder/modulo operator (%) and see if it equals 0 (i.e. no remainder)
- ▶ How can we check all of the numbers?
 - ▶ We can use a for loop.

isprime_slow function

```
def isprime_slow(num):  
    """ Returns True if the input is a prime number, False otherwise """  
    for i in range(2, num):  
        if num % i == 0:  
            return False  
  
    return True
```

```
>>> isprime_slow(5)  
True  
>>> isprime_slow(6)  
False  
>>> isprime_slow(100)  
False  
>>> isprime_slow(101)  
True
```

Prime numbers - a faster version

- ▶ Do we need to check all of the numbers up to that number?
 - ▶ No, we just need to check up to $\sqrt{\text{number}}$ (inclusive).

isprime function

```
import math

def isprime(num):
    """ Returns True if the input is a prime number, False otherwise """
    for i in range(2, int(math.sqrt(num)) + 1):
        if num % i == 0:
            return False

    return True
```

```
>>> isprime(5)
True
>>> isprime(6)
False
>>> isprime(100)
False
>>> isprime(101)
True
```

```
import math
```

- ▶ A second way to import a module: `import module_name`
- ▶ To reference a function within that module, you then say `module_name.function_name`
- ▶ Why might we use this option, i.e. when would we use `from math import *` instead of `import math`?
- ▶ Use the first if you're going to be using the functions a lot and it's clear that they come from that module.
- ▶ Use the second to be extra clear where the functions are coming from and to avoid naming conflicts.

Print the first x prime numbers

- ▶ How could we use `isprime` to print out the first 10 (100, 1000, etc) prime numbers?
 - ▶ Some sort of loop? Will a for loop work?
 - ▶ We don't know when we're going to stop.
 - ▶ We'd like to keep a count of how many we've seen and only stop when we've reached the number we want.

Lecture 5: while loops and main function

- ▶ prime numbers
- ▶ while loops

while loops

```
while <bool expression>:
```

```
    statement1
```

```
    statement2
```

```
    ...
```

```
statement3
```

- ▶ As long as the <bool expression> evaluates to True, it continues to repeat the statements statement1 and statement2. When the expression evaluates to False, it exits the loop and then continues on and executes statement3, etc.

Using a while loop for our prime numbers problem

- ▶ Keep a count of how many primes we've found (initially starts at 0).
- ▶ Check each number (starting at 2):
 - ▶ if it's prime:
 - ▶ print it out
 - ▶ increment the counter of how many primes we've found
- ▶ keep repeating this as long as (while) the number of primes we've printed is less than the number we want.

firstprimes function

```
def firstprimes(num):  
    """ Prints out the first num primes """  
    count = 0 # the number of primes we've printed out  
    current = 2 # the current number we're checking  
  
    while count < num:  
        if isprime(current):  
            count += 1 # same as count = count + 1  
            print(current)  
  
        current += 1 # same as current = current + 1
```

Emulate a while loop as a for loop

```
▶ for i in range(10):  
    ...
```

is equivalent to writing:

```
i = 0
```

```
while i < 10:
```

```
    ...  
    i = i + 1
```

Infinite loops

- ▶ `while True:`
 `print("hello")`
- ▶ This code will never stop! It will keep printing `hello\nhello\nhello...`.
- ▶ Sometimes, it will look like the program just froze if you're not actually printing anything out.
- ▶ You can stop this by selecting "Stop console" (red square).
- ▶ Be careful about these with your program. They're called infinite loops.
- ▶ If you think you might have an infinite loop, put in some print statements to debug and think about when the boolean expression will become `False` and make sure that is going to happen in your loop.

Resources

- ▶ Textbook: [Chapter 8](#).
- ▶ [while.py](#)

Practice Problems

- ▶ [Practice 3 \(solution\)](#)