

Tutorial 4 – Week 5

We'll be starting at the 10 minute mark

if nothing else, write `#cleancode`

Agenda

- Python built-in functions and modules
 - **min()**, **max()**, module **random**
- Lab Review
 - using **min()** and **max()** to limit the value of a variable to a given range
- Lecture Review
 - **Iteration: while** loops
- Practice problems

Learning Objectives

After completing this tutorial, learners should be able to:

- understand the semantics of Python built-in functions **min()**, **max()** and the functions in the built-in module **random**
- understand how to use min and max to limit the value of a variable to a given range
- understand the concept of iteration/looping
- recognize **while** loops
- understand the concept of “loop variable”
- understand the need to initialize and update loop variables
- understand the concept of “loop entry condition”
- recognize the need for using/implementing an iterative structure
- write syntactically and semantically correct **while** loops

Review of Python Built-in Functionality

if nothing else, write `#cleancode`

Python built-in functions: `min()` and `max()`

- **Semantics:** `min()` / `max()` return the smallest / largest value among the given inputs.
- **Function call syntax:**

```
min(arg1, arg2, arg3,...)
```

```
max(arg1, arg2, arg3,...)
```

```
var min = min(6, 3, 8, 10)  
print(var_min)
```

```
var max = max(4, 3, 5, 8)  
print(var_max)
```

Python built-in module **random**

- **random** implements pseudo-random number generators for various distributions
- Functions defined in module **random** (a partial list):
 - **randrange(start, stop, step)**: returns a randomly selected element from `range(start, stop, step)`
 - **random()**: returns the next random floating point number in the range `[0.0, 1.0)`.
 - **randint(a, b)**: returns a random integer `N` such that $a \leq N \leq b$
 - **uniform(a, b)**: returns a random floating point number `N` such that $a \leq N \leq b$ for $a \leq b$ and $b \leq N \leq a$ for $b < a$.

Review of Lab 2

if nothing else, write `#cleancode`

Limiting the value of a variable to a particular range

TASK: limit the value of a variable within given lower and upper limits

Multiple options, e.g., :

- use conditional statements, i.e., IF-statements
- use `min()` and `max()`

constrain the value of **var** to be **LARGER** than lower limit

```
var = 10  
lower = 0  
upper = 20
```

```
var = max(lower, var)
```

constrain the value of **var** to be **SMALLER** than the upper limit

```
var = min(upper, var)
```

equivalent

```
var = 10  
lower = 0  
upper = 20
```

```
var = max(lower, min(upper, var))
```


Review of Lecture

Topic 1: Iteration

WHILE Loops

if nothing else, write `#cleancode`

Iteration using `while` loops

- When to use it?
 - When you want a piece of code to be executed repeatedly, as long as a particular condition is **True**.
- Syntax (general structure) of a `while` loop:

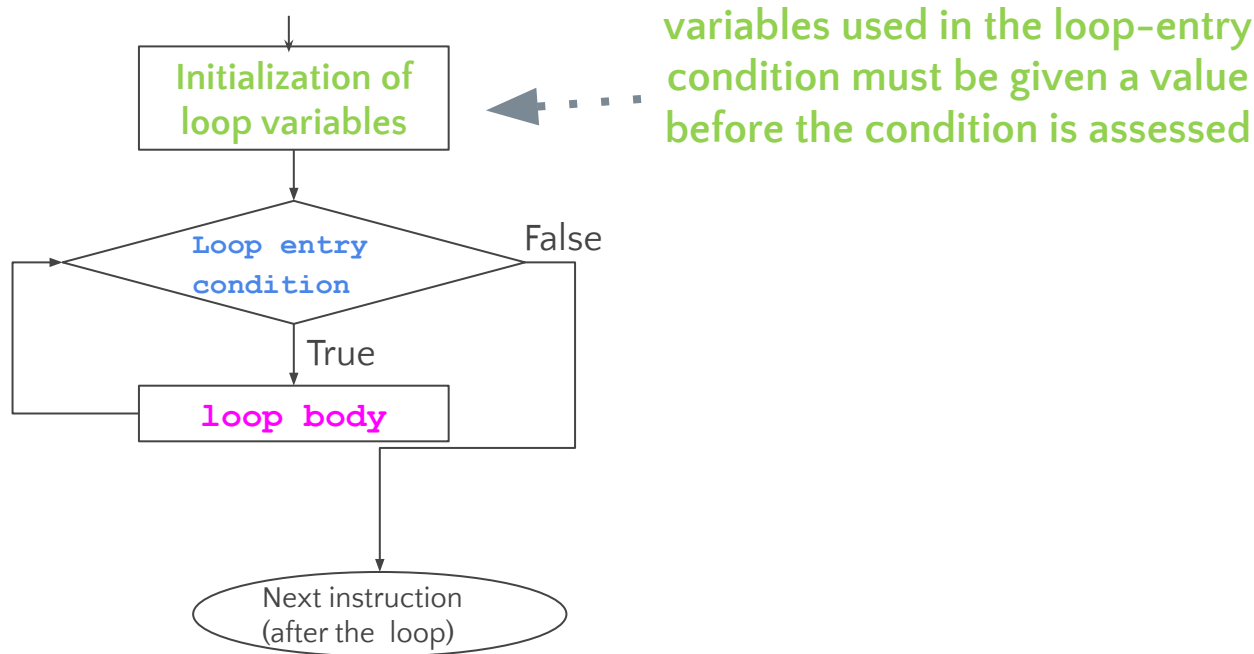
```
while condition:  
    body
```

Must be indented !

Don't forget the colon !

Flowchart representation of a generic **while** loop

Flowchart representation of a **generic** `while` loop



Notes on **while** loops

- A while loop is particularly useful when you **don't know** **how many iterations** will occur, but you **know** **when the iteration should stop**.
- Make sure to appropriately update in the body the loop the variables that appear in the entry condition.
 - Failing to make sure that the condition statement becomes **False** at some point may result in an **infinite loop**.
- Two useful commands for additional flow control:
 - **continue**: terminates the current iteration immediately and continue to the next iteration
 - **break**: terminates the loop immediately

Practice Problems

if nothing else, write `#cleancode`

slido



What is the output of the following code?

① Start presenting to display the poll results on this slide.

Review Practice Problem 1

What is the output of the following code?

```
i = 0
j = 3
while 0 < j < 10:
    if j % 2 == 0:
        j = j * 2
    else:
        j = j + 1

    print(i, j)
    i += 1
```

- A 0 3
 1 4
 2 8
- B 0 4
 1 8
 2 16
- C 1 4
 2 8
 3 16
- D Error is
 thrown
- E None of the
 above

slido



**What is the result of
executing the following
code?**

① Start presenting to display the poll results on this slide.

Review Practice Problem 2

What is the result of executing the following code?

```
number = 5
while number <= 5:
    if number < 5:
        number = number + 1
    print(number)
```

- A. The program will loop indefinitely
- B. The value of **number** will be printed exactly 1 time
- C. The **while** loop will never get executed
- D. The value of **number** will be printed exactly 5 times

Coding Question 1

Problem statement

Write a Python program that:

- asks the user to input an integer value (in base 10)
- finds the smallest digit(s) of the representation of the integer value in base 10.

Additional requirements

Your solution should:

- **not** cast the integer value into a string value.
- **use while** loops

Coding Question 2

Problem Statement

Write a python program to find the first 9 prime numbers of the Fibonacci sequence.

Recall that:

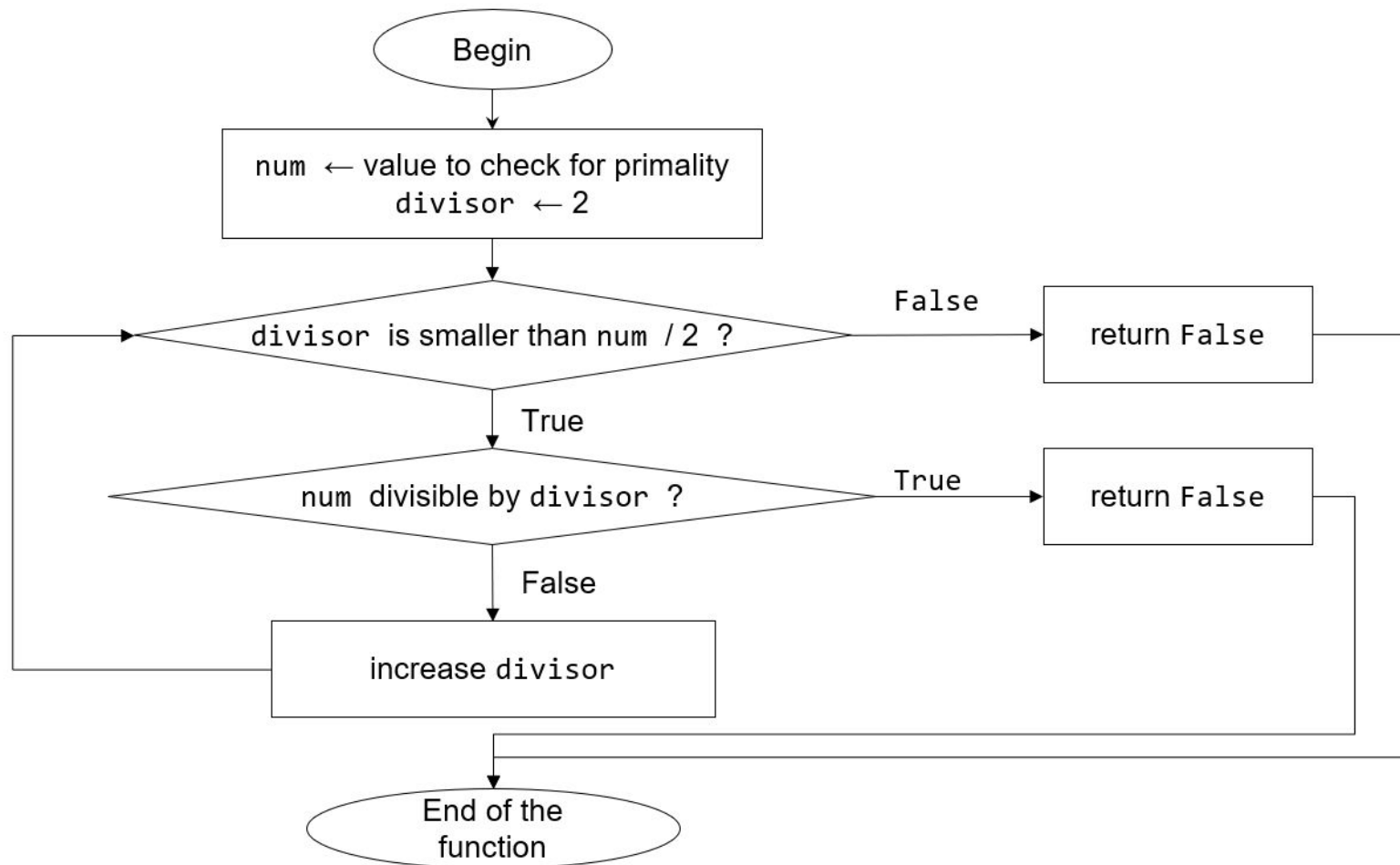
- Each number in the Fibonacci sequence is the sum of the two preceding ones. The sequence starts with 0 and 1.
 >> For example: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...
- Prime numbers are numbers that have only 2 factors: 1 and themselves

Hints: continue on the next page →

Coding Question 2 Hints

Take baby steps:

1. Write a function to check if a number is prime. Your function should take in an integer and return a boolean value (**True** if the input is prime and **False** otherwise).
 - a flowchart describing a strategy for primality checking is given on the next slide.
2. Write a `while` loop that:
 - at every iteration
 - generates the next number in the Fibonacci sequence
 - checks if the number just generated is a prime number using the function you implemented at Step 1
 - stops after 9 primes were generated



slido



Any questions?

① Start presenting to display the poll results on this slide.