

# CS 114 Tutorial 3

Sept 26 2025

# Goals for this Week:

- ❑ Submit Assignment 01 (due tonight at 5:30 pm)
- ❑ Look at assignment 02 (releases tonight at 5:30pm)
- ❑ Understand how to write while and for loops
- ❑ Be on the lookout for infinite loops
- ❑ Know how to use functions as parameters (callables)
- ❑ Remember to use import typing

## Warm Up Questions

Create a function `sum_odd` that takes an integer `n` and returns the sum of odd numbers from `1,...,n`. For `n <= 0`, return `0`.

*`assert sum_odd(5) == (5+3+1)`*

*Write this question using a while loop then with a for loop.*

Write a function `do_math(x: float, y: float, operator: typing.Callable)`. The function will return the integer value after operator is performed on `x` and `y`.

Assert `do_math(3.1, 5.0, max) == 5`, "operator is max"

Go to [vevox.com](https://vevox.com)

Sign in using the session ID: 118-503-607



# While Loops

```
def loops(x: int) -> int:  
    while x > 3:  
        x = x // 2  
    return (x)  
  
print(loops(15))
```

What would this print?

A. 5

B. 3

C. 2

D. 1



## While Loops

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

How many times will this print a number?

A. 1

B. infinite

C. 0

D. 5



# While Loops

```
var1 = -2
var2 = 0
while var1 != 0:
    var1 = var1 + 1
    var2 = var2 - 1
print("var1:", var1, "var2:", var2)
```

What will var1 and var2 be in the print statement?

- A. -2 and 0      B. 0 and -1      C. 0 and -2      D. nothing!  $\infty$  loop



## For Loops

```
sum = 0
for counter in range (1,7,2):
    sum = sum + counter
print (sum)
```

What will sum be after running?

A. 21

B. 16

C. 8

D. 9



# For Loops

```
for number in range (11):  
    print ("Number:", number)
```

What will be the last thing printed?

- A. Number: 10   B. Number: number   C. Number: 0   D. Number: 11





# Annotations

```
import typing

def function(x, n1):
    return (n1*x)
```

```
import typing

def function(x, n2):
    return (n2(x))
```

How would you annotate n in each function?

A. n1: float

n2: float

B. n1: typing.Callable

n2: float

C. n1: float

n2: typing.Callable

Write a function called `exponent(x)` that takes an integer value `x` and approximates the value of  $e^x$  using the taylor series expansion below. The taylor series will stop when it reaches a term smaller than 0.001.

**Note:** You can use `math.factorial()` in your code and `math.e` to test your code. You will also need to use a loop, is while or for a better fit?

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^k}{k!}$$

Example:

$$e^1 \sim 1 + 1 + 1/2! + 1/3! + 1/4! + 1/5! + 1/6! \\ 2.71828 \sim 2.7180$$