AUBURN
ENGINEERING

ENGR 1110

Module 4 Lecture

Resources

Python Software Foundation

`https://python.org/`

Python Anywhere

`https://www.pythonanywhere.com/`

Official Python Documentation

`https://docs.python.org/`

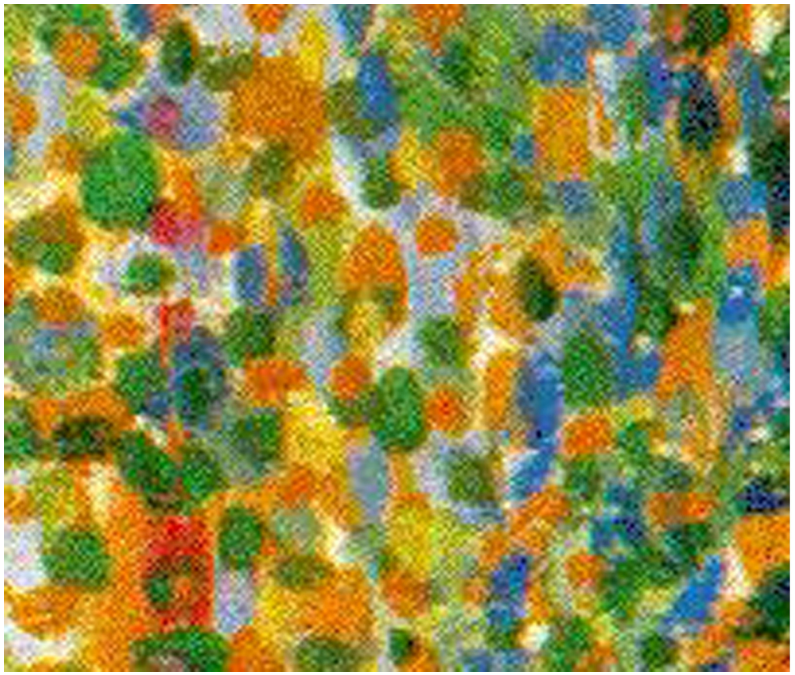
Python Tutor

`https://pythontutor.com/`

Python Style Guide (PEP 8)

`https://peps.python.org/pep-0008/`

See the forest *and* the trees…



https://www.principlegallery.com/technique-tuesday-pointillism-take-two/

Georges Seurat, *A Sunday Afternoon on the Island of La Grande Jatte*, 1884-86, Public domain, via Wikimedia Commons

Programming fundamentals

Consume input, Create output Ch 2: Intro to Python

Remember things

Calculate things

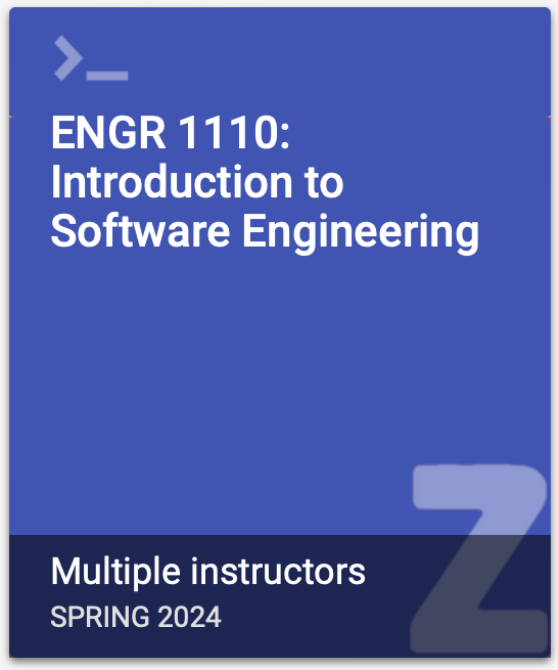
} Ch 3 & 4: Variables, Expressions, Types, Strings, Lists, Sets, Tuples, Dictionaries

Make things happen in sequence Ch 2, 3, 4

Ask a question, decide what to do based on answer Ch 5: Branching

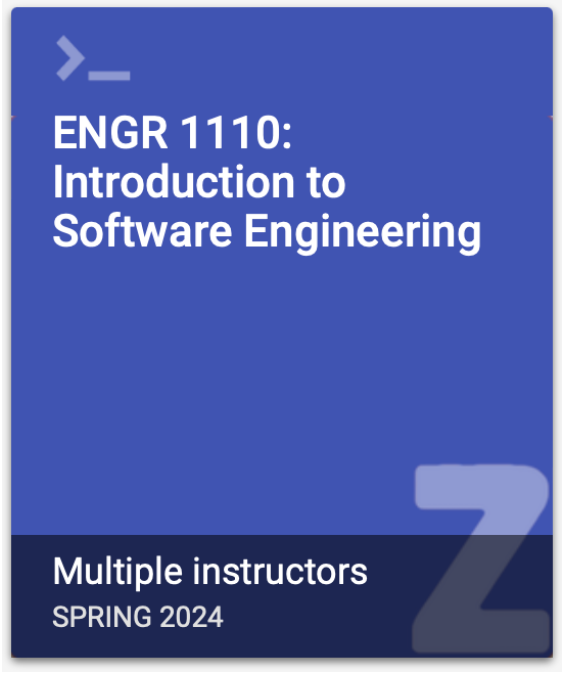
Make things happen repeatedly Ch 6: Loops

zyBook



Chapter 6. Loops

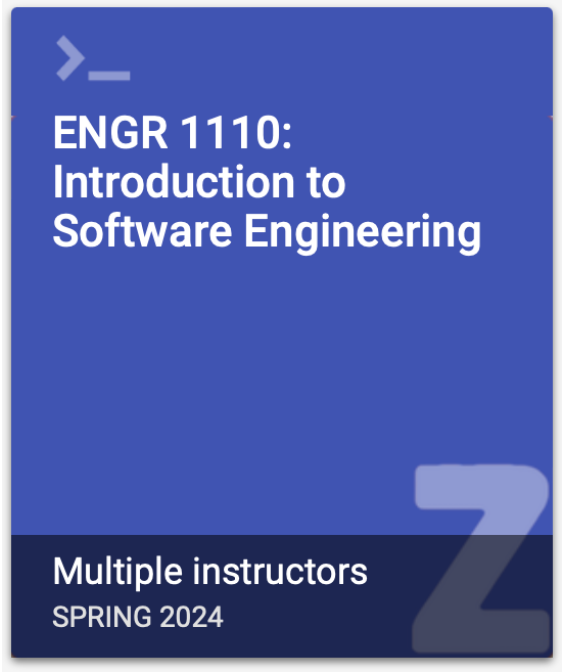
zyBook



Chapter 6. Loops

zyBook

zyBooks
A Wiley Brand



Chapter 6. Loops

6) Loops

zyBook

6.16 LAB: Adjust values in a list by normalizing

When analyzing data sets, such as data for human heights or for human weights, a common step is to adjust the data. This adjustment can be done by normalizing to values between 0 and 1, or throwing away outliers.

For this program, adjust the values by dividing all values by the largest value. The input begins with an integer indicating the number of floating-point values that follow. Assume that the list will always contain positive floating-point values.

Output each floating-point value with two digits after the decimal point, which can be achieved as follows:
`print(f'{your_value:.2f}')`

Ex: If the input is:

```
5
30.0
50.0
10.0
100.0
65.0
```

the output is:

```
0.30
0.50
0.10
1.00
0.65
```

The 5 indicates that there are five floating-point values in the list, namely 30.0, 50.0, 10.0, 100.0, and 65.0. 100.0 is the largest value in the list, so each value is divided by 100.0.

Incremental programming

```
# initialize and input list of values
```

Incremental programming

```
# initialize and input list of values




# output list
```

Incremental programming

```
# initialize and input list of values

# calculate maximum value



# output list
```

Incremental programming

```
# initialize and input list of values

# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values




# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []









# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input









# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:









# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:
    input a float value and store it in the values list




# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:
    input a float value and store it in the values list

## TEMP: print the list just to check




# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:
    input a float value and store it in the values list

## TEMP: print the list just to check
for index, value in enumerate(values):
    print(f'{index}, {value:.2f}')


# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:
    input a float value and store it in the values list

# calculate maximum value

# normalize list with maximum

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:
    input a float value and store it in the values list

# calculate maximum value
    # Use the max() function to do this

# normalize list with maximum


# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:
    input a float value and store it in the values list

# calculate maximum value
    # Use the max() function to do this

# normalize list with maximum
    # Use a for loop with enumerate() to divide each value by max

# output list
```

Incremental programming

```
# initialize and input list of values

values = []
set num_values to the value entered from input
use a loop that runs num_value times to:
    input a float value and store it in the values list

# calculate maximum value
    # Use the max() function to do this

# normalize list with maximum
    # Use a for loop with enumerate() to divide each value by max

# output list
    # Use a for loop with a print statement to do this
```