



## Task: Data Structures

[www.hyperiondev.com](http://www.hyperiondev.com)

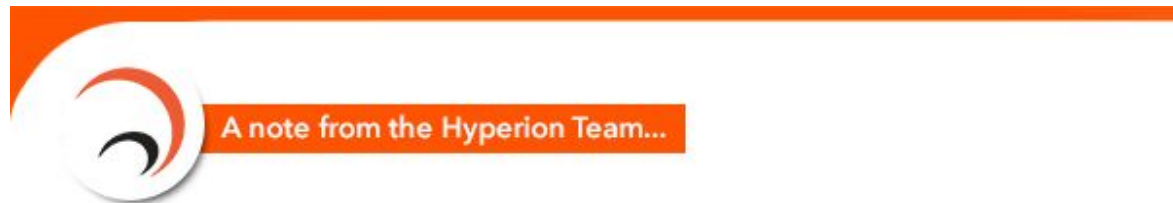
# Introduction

## Welcome to the Data Structures Task!

### Overview:

In this Task you will learn about data structures in programming. A data structure is a specialized format for organizing and storing data so that it may be used efficiently (retrieving or manipulating it). General data structure types include arrays, lists, files, tables, trees and so on. Different data structures are suited to different kinds of applications and some are highly specialized to specific tasks. The most common data structure in Python is a list and this is what we will be focusing more on.

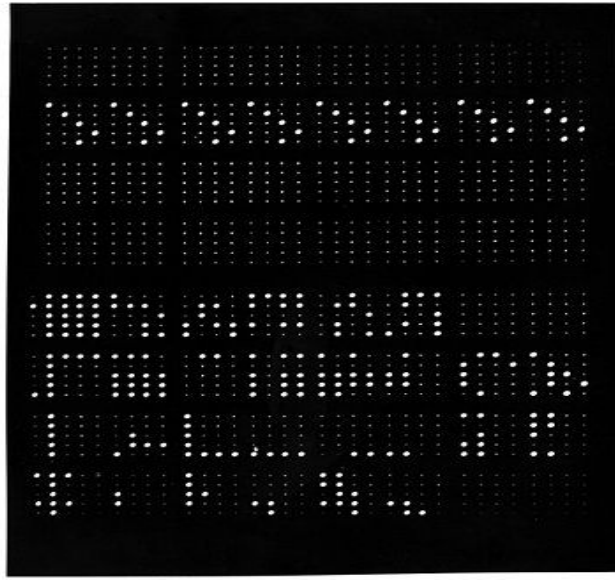
-The Hyperion Team



### The Williams-Kilburn Tube

The Williams-Kilburn Tube(or Williams Tube) was an early computer storage device that used cathode-ray tubes to store bits(0s and 1s). This device was the first electronic memory that stored its bits as dots on a screen. On the face of the tube the dots represent the 1s and the spaces represent the 0s as seen in the image below. The dots faded after less than a second and needed to be refreshed constantly.

The device was invented by Fred Williams at Manchester University in 1946, and was developed alongside Tom Kilburn. The device was used in the Manchester Mark I Computer and by IBM, only having 2k bits of memory.



To find out more about how this storage device worked, visit [http://www.radiomuseum.org/forum/williams\\_kilburn\\_williams\\_kilburn\\_ram.html](http://www.radiomuseum.org/forum/williams_kilburn_williams_kilburn_ram.html).

- The Hyperion Team

---

## Instructions

First read **example.py**, open it using Notepad++ (Right click the file and select 'Edit with Notepad++').

- **example.py** should help you understand some simple Python. Every task will have example code to help you get started. Make sure you read all of **example.py** and try your best to understand.
- You may run **example.py** to see the output. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing this Task to become more comfortable with Python.
- You are not required to read the **Additional Reading.pdf** in its entirety. It is purely for extra reference.

# Compulsory Task

## Follow these steps:

Once you have read and completely understood **example.py**, write a Python program that takes in a user input as a String. While the String is not "John", add every String entered to a list until "John" is entered. Then print out the list. This program basically stores all incorrectly entered Strings in a list where "John" is the only correct String. Save this program as **John.py** in this folder.

Example program run (what should show up in the Python Console when you run it):

Enter your name : <user enters Tim>

Enter your name : <user enters Mark>

Enter your name: <user enters John>

Incorrect names: ['Tim', 'Mark']

## Bonus Optional Task:

Edit the above program to allow the user to enter an integer before they enter the name. This integer defines how many 'tries' the user will get to enter the right name. If the user exceeds the number of tries, the program must stop.

# Task Statistics

Last update to task: 07/01/2016.

Course Content Manager: Riaz Moola.

Task Authors: Riaz Moola and Sinead Urisohn.

Task Feedback link: [Hyperion Development Feedback](#).