

# Upper School 1

## Long Live the Listless!



Nasri Academy  
Thurs. Sept. 19<sup>th</sup>, 2019  
By Julio B. Figueroa

# Overview

- **Review**
- **List**
- **Tuples, Sets, and Dictionaries**
- **Indices**
- **Classwork Assignment: (Dictionary)**
- **Introduction to matrices**

# Review

- <https://docs.python.org/3/>
- Click on tutorial to get started

The screenshot shows the Python 3.7.4 documentation homepage. On the left, there's a sidebar with links for 'Download', 'Docs by version' (listing versions from 3.9 down to 2.7), and 'Other resources' (including PEP Index, Beginner's Guide, Book List, and Audio/Visual Talks). The main content area has a title 'Python 3.7.4 documentation' and a welcome message: 'Welcome! This is the documentation for Python 3.7.4.' Below this, under 'Parts of the documentation:', there are several sections with titles and descriptions: 'What's new in Python 3.7?' (with a note about 'all "What's new" documents since 2.0'), 'Tutorial' (with a 'start here' link), 'Library Reference' (with a note about 'keep this under your pillow'), 'Language Reference' (with a note about 'describes syntax and language elements'), 'Python Setup and Usage' (with a note about 'how to use Python on different platforms'), 'Python HOWTOs' (with a note about 'in-depth documents on specific topics'), 'Installing Python Modules' (with a note about 'installing from the Python Package Index & other sources'), 'Distributing Python Modules' (with a note about 'publishing modules for installation by others'), 'Extending and Embedding' (with a note about 'tutorial for C/C++ programmers'), 'Python/C API' (with a note about 'reference for C/C++ programmers'), and 'FAQs' (with a note about 'frequently asked questions (with answers!)'). At the bottom of the main content area, there's a link 'Indices and tables'.

Download  
Download these documents

Docs by version

- Python 3.9 (in development)
- Python 3.8 (pre-release)
- Python 3.7 (stable)
- Python 3.6 (security-fixes)
- Python 3.5 (security-fixes)
- Python 2.7 (stable)
- All versions

Other resources

- PEP Index
- Beginner's Guide
- Book List
- Audio/Visual Talks

Indices and tables

## Python 3.7.4 documentation

Welcome! This is the documentation for Python 3.7.4.

**Parts of the documentation:**

[What's new in Python 3.7?](#)  
*or all "What's new" documents since 2.0*

[Tutorial](#)  
*start here*

[Library Reference](#)  
*keep this under your pillow*

[Language Reference](#)  
*describes syntax and language elements*

[Python Setup and Usage](#)  
*how to use Python on different platforms*

[Python HOWTOs](#)  
*in-depth documents on specific topics*

[Installing Python Modules](#)  
*installing from the Python Package Index & other sources*

[Distributing Python Modules](#)  
*publishing modules for installation by others*

[Extending and Embedding](#)  
*tutorial for C/C++ programmers*

[Python/C API](#)  
*reference for C/C++ programmers*

[FAQs](#)  
*frequently asked questions (with answers!)*

# Lists

- **Written as a data type with comma separated values**
  - Values, that are separated by commas
- **Lists are created using brackets []**
- **Values are assigned to a variable like so**
  - ListA = [1, 2, 3, 4]
  - ListB = ["car", "planes", "boat"]
- **ListA, and ListB are the variables**
- **1,2, "car", "planes" are values**

# Lists (cont.)

```
Welcome  x  listMath.py  x
1     listA = [3, 4, 3, 3, 1, 2]
2     listB = [1, 1.1, 2, 2, 3, 3]
3
4     listC = ["fruit", "vegetable", "herb"]
5     listD = ["apple", "carrot", "basil"]
6
7     print(listA+listB)
8     print(listC+listD)
9     # what happens when you do listA+listD?
```



# Lists (cont.)

## What happens when your list is too long?

```
listLong = ["longstring", "longerString", "longestString",
            "evenLongerString", "ridicuouslyLongerString",
            "whyAretheseStringsssoLong?"]
print(listLong)
```



```
listLong = ["longstring", "longerString", "longestString", \
            "evenLongerString", "ridicuouslyLongerString", \
            "whyAretheseStringsssoLong?"]
print(longList)
```

# Tuples

- **Tuples are not mutable**
  - Or non-mutable
  - It means they can't be modified like lists
- **Tuples are parentheses ()**
  - Lists use brackets []
- **Use Tuples for constants, or things that you don't want changed**

# Sets

- **Sets are not mutable**
  - Or non-mutable
  - It means they can't be modified like lists
- **Used to check for membership**
  - Sets use curly braces {}
- **Use sets to check for membership or to delete duplicate entries**
  - Ask if c is a set of B
    - $B = \{a, b, c, d, e, f\}$
  - You can't have duplicate entries
    - $B \neq \{a, b, a, d, e, f\}$

# Indices

- To search through lists, tuples, and sets we use an index
  - $A = [1, 2, 3, 4]$
- An index is like a house address
  - $A[0] = 1$
  - $A[1] = 2$
  - $A[2] = 3$
  - $A[3] = 4$
- The index starts at 0, and runs through 3 in this example
  - 0, 1, 2, 3
- What if?
  - $B = [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1028]$
  - What is  $B[2] + A[3]$
  - What is  $A[5] + B[0]$



# Classwork Assignment: (Dictionary)

- **There are 100's of pokemon**
  - (originally referred to as pocket monsters)
- **Make a dictionary containing your favorite 10.**
  - 
  - Set the key to their name
  - For reference visit:  
<https://pokemondb.net/pokedex/national>

# Classwork Assignment: (Dictionary)

- Mr. Figueroa's favorite

# Introduction to matrices

- **Singular Matrix**
- **Plural Matrices**
  - Aka the rice of mathematics
- **The index address uses 2 values**
  - $A[1,2] = A_{12}$

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & & & A_{2n} \\ \vdots & & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$