

# Upper School 2 - Coding

## FruityLoops



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

# Overview

- **Review**
- **Loops**
  - For
  - While
- **Choice**
  - If/Else
  - switch
- **Def my\_function():**
- **Assignment: function(myFunction)**
- **Introduction to matrices**

# Review

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

The screenshot shows the Python 3.7.4 documentation page. On the left, there is a sidebar with the following sections:

- 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

The main content area is titled "Python 3.7.4 documentation" and includes a welcome message: "Welcome! This is the documentation for Python 3.7.4." Below this, it lists "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!)*

At the bottom of the main content area, there is a link for "Indices and tables:".

# Loops

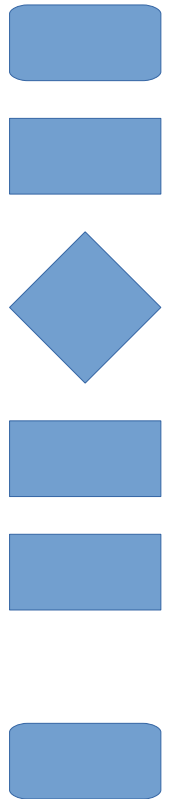
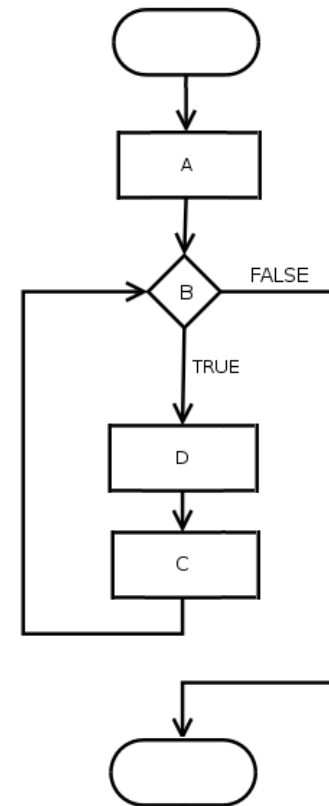
**These structures control execution of a program.**

**They run repeatedly until some condition(s) is/are met**

**We introduce two types of loops**

- **For loops**
- **While loops**

for(A;B;C)  
D;



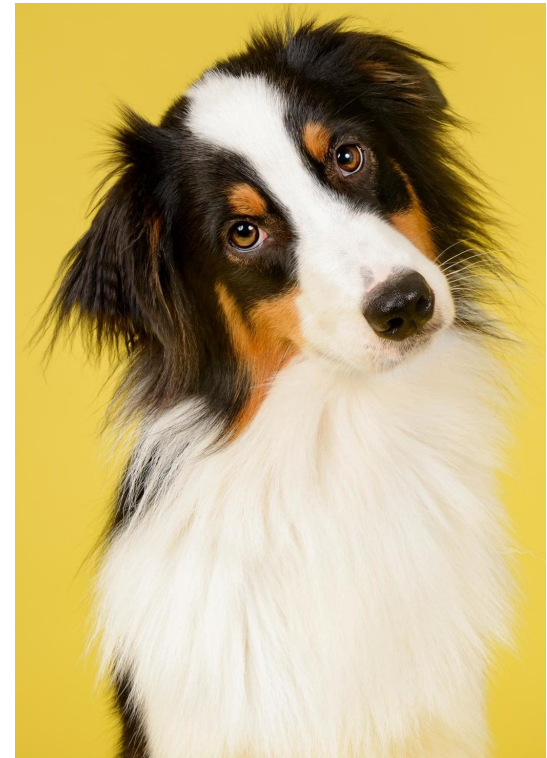
# For

**These loops continue until a condition is met.**

- **If that condition is not met, the execution jumps to the else portion**
- **For  $j < 1$ :**
  - Do something
- **Else  $j = 0$ :**
  - Do something else
- **Else  $j > 0$ :**
  - Do yet another thing

# For (cont.)

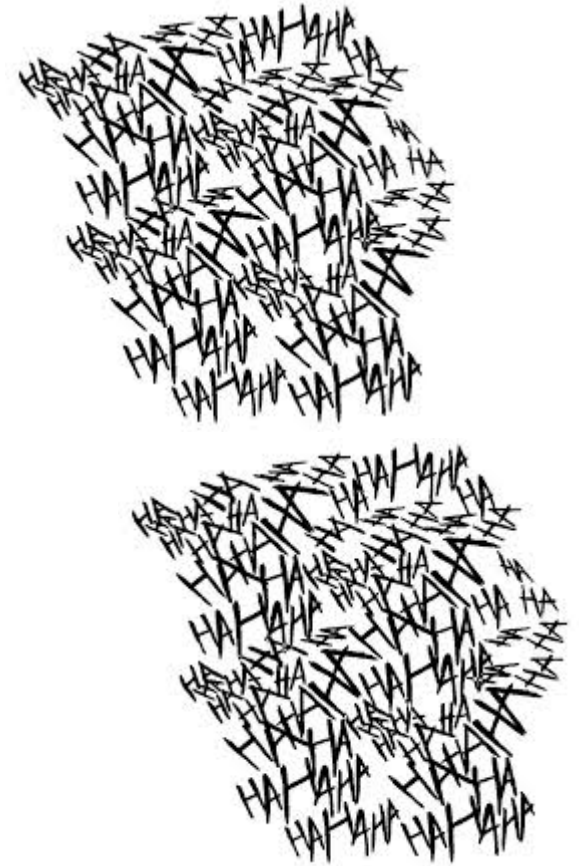
```
1  # for example (pun intended)
2  words = ['dog', 'funeral', 'flowers']
3  for i in words:
4      print(i, len(i))
5
6  # i is an index that ranges from 0 to 2
7  # word is a list with 3 values
8  # len() stands for length
9
10 for i in range(10):
11     i = i+i*2
12     print(i)
13 # PEMDAS rules apply
14
```



# While

```
i = 1
while i < 6:
    print(i)
    i = i + 1

# save before running this next part
x = 1
while x >= 1:
    print("Ha Ha Ha")
    x = x + 1
```



# While (cont.)

- **while a < 10:**
- **while A > 10:**
- **while the sky is blue && dogs > cats**

**An exit strategy is recommended**



# If/Else

```
1  # loop stuff
2  import random
3
4  a = random.randint(-10, 10)
5  if a > 0:
6      print("yes")
7  else:
8      print("no")
9
10 # run this (ctrl+shift+b) 10 times
11 # Do you see the two different outcomes?
```

# Def my\_function():

- **Sets are not mutable**
  - Or non-mutable
  - It means they can't be modified like lists
- **Used to check for membership**
  - Sets use curls {}
- **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
    - $B \setminus = \{a, b, a, d, e, f\}$

# Assignment: function(myFunction)

- **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  $A[5] + B[0]$



# Introduction to matrices

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

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

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