I THINK THAT MODERN PHYSICS HAS DEFINITELY DECIDED IN FAVOR OF PLATO. IN FACT THE SMALLEST UNITS OF MATTER ARE NOT PHYSICAL OBJECTS IN THE ORDINARY SENSE; THEY ARE FORMS, IDEAS WHICH CAN BE EXPRESSED UNAMBIGUOUSLY ONLY IN MATHEMATICAL LANGUAGE.
WERNER HEISENBERG

THE ONLY SHIBBOLETH THE WEST HAS IS SCIENCE. IT IS THE PREMISE OF MODERNITY AND IT DEFINES ITSELF AS A RATIONALITY CAPABLE OF, INDEED REQUIRING SEPARATION FROM POLITICS, RELIGION AND REALLY, SOCIETY. MODERNISATION IS TO WORK TOWARDS THIS. . . IF ONE LOOKS AT THE WORKS OF NEWTON TO EINSTEIN, THEY WERE NEVER SCIENTISTS IN THE WAY MODERNITY UNDERSTANDS THE TERM.
BRUNO LATOUR

THE BOUNDARY BETWEEN SCIENCE FICTION AND SOCIAL REALITY IS AN OPTICAL ILLUSION.
DONNA HARAWAY

DR. DOEG

INTRODUCTION TO MECHANICS

THE INVISIBLE COLLEGE

PUBLISHED BY THE INVISIBLE COLLEGE

HTTPS://GITHUB.COM/TRISMEG/PHYSICSBOOK

First printing, November 2015

Contents

List of Figures

List of Tables

Dedicated to the ghosts of the college and

the spectral bodies of physics.

Note

This physics text is an OpenSource academic project developed in abstraction by the Invisible College. The manuscript is written in LaTeX and makes use of the `tufte-book` and `tufte-handout` document classes.

http://latex-project.org/ftp.html
https://git-scm.com/downloads

Array 1-D, 2-D, 3-D

Intro

Array is like a storage, it can fill with string or integer. In 1-D, 2-D it can also represents the x-axis and y axis.

Creating arrays

Arrays is created buy blanket.

Example:

```
a=[ ]        *a is the array name that you want.
```

The things in the [ ] and be store and when you want to access it you will need its position in the array and type like a[0]

Example:

```
a=["apple","orange","banana"]
```

If you want to print banana form the array, you may want to type

```
print a[2]
```

Filling arrays

Everything can be store in the array, strings, integers, arrays. When you create an array you can fill things in it as the default things that the array have.

Example:

```
a=["Billy","Bud",90,60,50]
b=["Anne","Chow",90,95,100]
c=["Jen","Bo",60,80,90]
```

If you want to add things into the array that u create already, you can use

```
array_name=array_name+[Things you want to add]
```

Example:

```
a=[apple]
```

and now I want to add orange into it, so we add

```
a=a+[orange]
```

To create 2-D or more array we need to create array in the nested for-loop.

Example:

```
a=[]
for i in range(N): *N how long you want the array to be
    b=[]    *This is a temporary array to generate every array inside the main array.
    for j in range(N):
        *Things you want to put in the array by b=b+[ ]
    a=a+[b]     *Here put the temporary array back to the main array.
```

Traversing array

Traversing array is visiting each element in the array and do something. In 1-D we can do it with for loop to identify things in array.
Example:

```
a=[1,2,3]
for i in range(len(a))    *len(a) =  Numbers of elements in the array
    *Things put here can edit the specific element a[i]
```

In 2-D we start using nested for-loop to identify the x-axis and y-axis.
So we use nested for-loop to traversing it too.

Example:

```
a=[[0,1],[0,0],[0,1]]
for i in range(len(a)):
    for j in range(len(a)):
        *Things put here can edit the specific element a[i][j]
```

In 3-D we use more for-loop to identify the more dimension.

Example:

```
a=[[[0,0],[0,0]],[[0,0],[0,0]]]
for x in range(len(a)):
    for y in range(len(a)):
        for z in range(len(a)):
            *Things put here can edit the specific element a[x][y][z]
```