

# CE 311K: Introduction to Computer Methods

Krishna Kumar

University of Texas at Austin

krishnak@utexas.edu

August 26, 2019

Q What would you like to learn from the course?

1 Simulations

2 Aspects of languages

3 Python variables and bindings

## On computable numbers



Alan Turing



3-Rotor Enigma



The Bombe

Enigma has 150,738,274,937,250 possible states. (credit: Rutherford journal)

# The Prophet and the Pioneer Computer Scientists



Ada Lovelace, circa 1838 (credit:  
Science Museum, California)



Grace Hopper (credit: unknown)

## To infinity and beyond...



Margaret Hamilton next to a stack of the Apollo Guidance Computer source code (1969, credit: MIT Museum) and Katie Bouman who developed the algorithm for creating the first-ever image of black hole (2019, credits: PBS).

② Could you guess the storage size requirements?



## ➊ Simulations

## ➋ Aspects of languages

## ➌ Python variables and bindings

# Disney's Frozen: Modeling snow



(c) Disney

## How to bury Anna under the snow?



(c) Disney

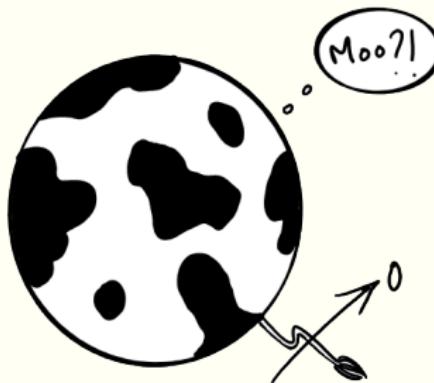
# How to animate like Disney?



(c) Disney

⌚ How to achieve the snow simulation?

## Modeling the real world: Spherical Cow



Consider a spherical cow of radius ' $R$ '  
and a uniform density ' $\rho$ '...



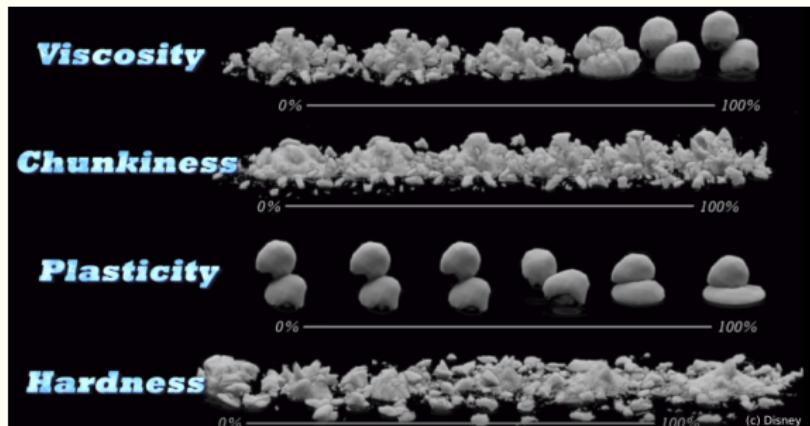
## How to animate like Disney: Effect of snow quantity



# What type of snow?



## Snow properties



## Snow material parameters

$$\begin{aligned}E_0 &= 1.4 \times 10^5 \\ \theta_c &= 2.5 \times 10^{-2} \\ \theta_s &= 5.0 \times 10^{-3} \\ \xi &= 10\end{aligned}$$



$$\begin{aligned}E_0 &= 1.4 \times 10^5 \\ \theta_c &= 2.5 \times 10^{-2} \\ \theta_s &= 7.5 \times 10^{-3} \\ \xi &= 10\end{aligned}$$



(c) Disney

$$\begin{aligned}E_0 &= 1.4 \times 10^5 \\ \theta_c &= 1.9 \times 10^{-2} \\ \theta_s &= 5.0 \times 10^{-3} \\ \xi &= 10\end{aligned}$$



$$\begin{aligned}E_0 &= 1.4 \times 10^5 \\ \theta_c &= 1.9 \times 10^{-2} \\ \theta_s &= 7.5 \times 10^{-3} \\ \xi &= 10\end{aligned}$$



## How to model snow?

(c) Disney



## 1 Simulations

### 2 Aspects of languages

### 3 Python variables and bindings

## Q What does a computer do?



IBM Bluegene/P supercomputer (credit: unknown)

# Companies using python

The popular YouTube video sharing system is largely written in Python

Google makes extensive use of Python in its web search system

Dropbox storage service codes both its server and client software primarily in Python

The Raspberry Pi single-board computer promotes Python as its educational language



## COMPANIES USING PYTHON



BitTorrent



NETFLIX

BitTorrent peer-to-peer file sharing system began its life as a Python Program

NASA uses Python for specific Programming Task

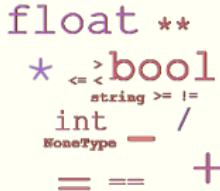
The NSA uses Python for cryptography and intelligence analysis

Netflix and Yelp have both documented the role of Python in their software infrastructures

⌚ What are the primary aspects of a language?



English word cloud (credit: Michael Twardos)



## Python word cloud (credit: unknown)

## Python program

- We'll be using Python 3.x
  - a program is a sequence of definitions and commands  
  - programs manipulate **data objects**
  - objects are

```
In [1]: type(5)
Out[1]: int
In [2]: type(3.0)
Out[2]: float
```

## Aspects of languages: Syntax and Static semantics

### Syntax

**Static semantics** is when syntactically valid strings have meaning:

Semantics is the meaning associated with a syntactically correct string of symbols with no static semantic errors:

- *English*: can have many meanings "Flying planes can be dangerous".
- *programming languages*: have only one meaning but may not be what programmer intended

## Expressions and operations

- Operations on 'ints' and 'floats':
- $i + j$ : *sum* (int || float)
- $i - j$ : *difference* (int || float)
- $i * j$ : *product* (int || float)
- $i / j$ : *division* (float)
- $i \% j$ : *remainder* of  $i$  divided by  $j$
- $i ** j$ :  $i$  to the *power* of  $j$

## ➊ Simulations

## ➋ Aspects of languages

## ➌ Python variables and bindings

Equal sign is an **assignment** of a value to a variable name.

```
pi = 3.14159  
pi_approx = 22/7
```

## Changing bindings

- can *re-bind* variable names using new assignment statements
- previous value may still stored in memory but lost the handle for it
- value for area does not change until you tell the computer to do the calculation again

```
pi = 3  
radius = 11  
area = 363  
radius = 14
```

$\pi$ : 3.141592653589793

e: 2.7182818284590452

Engineers:



## Naming matters

② What's wrong with the following code segment?

```
a = 3.14159  
b = 11.2  
c = a*(b**2)  
print(c)
```

```
pi = 3.14159  
diameter = 11.2  
area = pi*(diameter**2)  
print(area)
```