# CSCI 1070: Taming Big Data

Dr. Abby Stylianou

#### Logistics

- Are you on the waitlist? Talk to me after class!
- Class meetings: Tu/Th, 11AM-12:15PM, Ritter 117
  - Lecture, discussions, in class coding
  - Occasional quizzes about readings
  - Attendance not mandatory, but you're gonna have a bad time if you don't show up
- Office Hours: Tues, 2:30-3:30PM, Ritter 107

#### Logistics

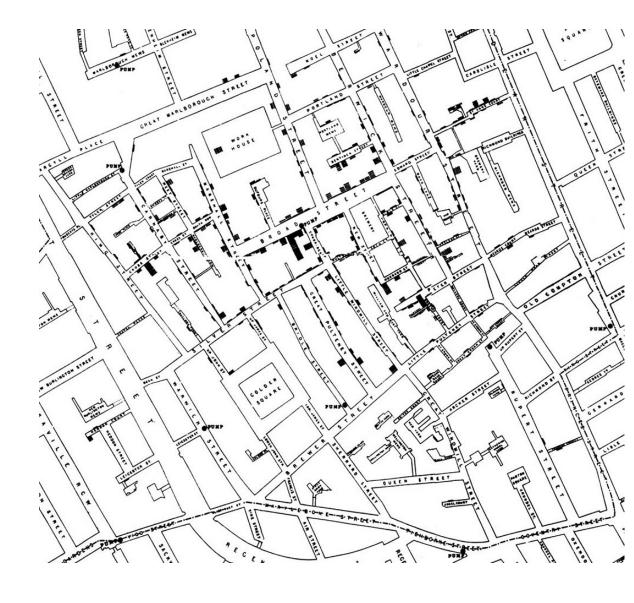
 Website/schedule/materials: https://cs.slu.edu/~stylianou/1070\_sp2020/

- Blackboard for turning in assignments and seeing grades
- Technologies: Python (w/ data science + scientific libraries),
   Jupyter Notebooks
- Class communications: Piazza! (not email)

• 1663: John Graunt is the first person credited w/ statistical data analysis in his studies of the bubonic plague in Europe, dealing with what he referred to as "overwhelming amounts of information"

					Y			Tab								1,600	40-				1629 16 1630 16 1631 16	34 164	01652	1657 164	In 2
The Years of our Lord	1647	648	1649	1650	1651	1652				1057	1658	559 16	60 16	529153	0 1031	1032	1033	1034	035					1658 165	
give and Stil-born	335	329	327	351	389	381	384	433 48			467	421 5	44 4	199 4			500	475		2 71	1 2 mm 2 2 2	4 2224	62452	1832 124	
and Fever	916	835 884	889 751	970	780	834	282 1		9 87	5 999	1800	009 10	48 9	579 71	11115	1108	7°4 953	623	622 2	3604	418 023	5 3865	4903 4	1303 4010	2378
elex and Suddenly	68	74	64	74	106	111	118	86 9	2 10	2 113	138	91	67	22 3	6	17	24	35	26		75 8	5 280	421	445 17	1 130
· A	4	,	1	3	6	6	7	1	4	5	5	3	8	13	8 10	13	6	4	3	4	54 1			14 10	9
ding ding	3	2	5	1	. 3	4	3	386 16	7	3 362	4	7	2	5	2 5	348	4	3	246	220	87 (46	11	21811	19 17	
by Flux, Scouring and Flux	155	176	802	289	833	762	200	7 1	8 36	7	233 3	6 2	51 4	149 43	8 352	5	278	512	346 12	3	25 . 19	24	3 1	26 19	125
entro	1		20	í		2	1	1		3			-		1	28		1500	1	3	85 112	105	157	150 114	609
gr,Gangrene and Fiftula -	26	29	31	19	31	53	36	37 7	3 31	24	35	63	52	20 1	4 23	-	27	30	24	30	1				8
er, Sore-mouth and Thruth	66	28	54	42	68	51	53	72 4			27		68	9	4 4	171			63	74	15 75			161 133	3364
3-brd forms and Infants	161	106	114	990	206	2801	158	192 17 343 108 101 8	7 201	236	225 3	220 1	94 1	05 237	7 112 8 2035		132	143	13 1	305 9	277 8453	4978	49104	788 4519	32100
kand Wind	103	71	85	82	76	102	80	101 8	5 120	113	170	116 1	67	48 5	71	7.7	- 10	3337	371	50	105 87 174 207 157 8266	341		197 247	1389
and Cough	2422	2200	2388	1988	22502	1103	28621	36 2 68 260	58	2757	31	33	24 18	10 5	0 1713	1797	45	955 20	80 24	57 5	57 8266	8999	991412	57 7197	44487
ramption and Cough	684	491			569	653	606	328 70	2 1027	807	841	742 10	31	52 8	7 18	241	221	386 4		109	198 173-	2198	265633	77 1374	9973
0.0			1						1 .			1			1	0	0	5	0	0	5 10	6	4	13 47	38
pie and Tympany	185	434	421		444			104 66	100	631			72 2	235 25		280		250 3	29	89 19	48 1734		182 2	15 1302	9623 827
rowned	47	40			49	50		30 4				57	48	43 3	3 29	34	37	32	32	45		2		2	2
xceffive drinking xecuted	8	17	29	43	24	12	19	21 19	22	20	18	7	18	19	13 12	18	13	13	13	13	62 52	97	12.66	79 55	384
sinted in a Rath	1 ,	4	2	2	1	3	4	1	4 3	1		4	5	3 1	0 7	7	2	5	6	8 1271	701 1846	10	8	8 9	74
alling-Sickness lox and fmall Pox	139	400	1190	184	525	1279		812 129	100000			523	54		10 58		72	1354	93	24	03 65	1913		4/ 29	10576
ound dead in the Streets	18	6	9	8	7	20	14	20 2	9 23		53	51	31		12 12	1 3	7	17	24	22	53 48	80		30 83	392
rench-Pox righted	4	29	15		3		2		1 1	0		14	9	1		I			1	3	14 24	35	25	34 28	134
Jout	9	5			7	7	5	17 1	8 7	10	13	13	4	18	8 20	11	14	17	7	20	71 50	48	59	45 47 7- 3-	279
Brief langed, and made-away themfelves		13	16		9	14	15	9 I	4 16		18	II	36	8	8 6	15		3	5	7 2	0 6	48	14	17 46	051
lead-Ach	900	1	11	2		2		71 6	5 3		77	35	76	47	59 3	43	35	45	54	63	47 197	180	212 2	6 16	998
aundice aw-faln	57.	35	39	49	41	43	57	2	2	, 3	1		1	10	16 1:		50	10	4	130 2	82 315	260	35	28 228	1619
mpostume	75	61	65	59	80	105	79	90 9	122	80	134	105	96	58	76 73	74	10	62	73		00 10	01		194 148	11
Killed by feveral Accidents	27	57	39	94	47	45	57	58 5		52	47	55	47	54	55 47	46 38	49 35	41	51 20		97 150	217	94	102 66	537
King's Evil	27	26	2.2	19	2.2	20	26	26 2	7 24	23	28	28	54	16 E	5 18	2	3	20	20	2	5 7	13	21	21 9	67
Lethargy Lepvolie	1 3	4	1	4	4	*			9 4		1	8	2	2		87	82	-	2		92 356	213		191 158	1421
Liver-grown, Spleen and Rickets	53	46	56		65	72	67	65 5		38	5 5		15		12 99		4	77	90		28 13	47	39	31 20	158
Meagrom	12	1	3	5	8	62	6	14.	3 2	5 7	80	5	74	42	2 3	80	21	33			27 8	133	155	59 51	757 18
feafles Aother	2		1 3	33	20	1	1	2	2 3		3 7	1	8 20	1					-1	3 8	10 1	17	13		86
furdered	3	2	2 3	6 28	28	3	30	36 5	8 5		50	70	43	4	10 13		8	14	51	14	34 40 82 7	111	123	27 77 215 86	525
verlaid and starved at Nurse	2 9	2	I L	9 20	23	20	29	18 2	2 2	3 20		17			13 17	25	14		25	17 1	599 1040		90	33 103	1638
lague lague in the Guts	3597	61	1 6	7 15	23	110	32	16	7 31		14		14	131	7 274			1	10	400	00 00	61	142	844 253	99
lcuritie	30	2	6 1	3 20	23		17			9 17		12	10	26 2	24 26	36	21		45	24	00	89	72	52 51	41
oifoned urples and Spotted Fever	14	5 4	7 4	2 60	54	60	75	89 5	6 5	2 56	126		146	32 5	8 58	38	24	125 2	45	397	22 79	300	278	290 243	184
Luintie and Sore-throat	1.	4 1	1 1	2 17	24	20	75 18 220	9 1	5 1		10	21	521	01	8 6	7	24	04	49	50	00 11	780	71	598 657	
Aickets fother, rifing of the Lights	15				260	1329	135	178 10	6 21	2 203	228	210	249	44	72 95		60	84	72	104	21 3		100	809 369	270
Lupture	1	6	7	7 1	6 7			15	2 2	0 19	18	12	28	2	6 4	9	4	3	10	13		2	1	2	1 0
ical'd head Scurvy	3	2 2	0 2	11 2	1 29	43	41	44 10		1 82	82	95	12	5	7 5		9		00	25	33 3	4 94	132	300 115	59
Smothered and flifled	6 19	1	13	2			25				47	61	48	23	24 20	0 48	19	19	22	29	91 8	9 65	115	144 141	50.
sores, Illcers, broken and bruife shot (Lim	bs I	5 1	7 1	7 1	20	32	1 "	34		1		7	20				1			18	100	25	26	13 07	2
pleen Shingles	1	2 1	7	1			13	13		6 2	5	7	7				1		-			1		i	
Starved	1	1	4	8	7 1	2	1	1	3	1 3	6	7	14			1	100			-	14	19	5	13 25	5
Stirch	1	-		29 2	81 50	41	44	38	49	57 7	2 69	22	30	35	39 5	8 50	58	49	33	45	114 18	5 14	4 173	247 5	93
Stone and Strangury Sciatica		1			1		1				1	186	214	1		1	3			6	1	6 12	1 295	247 216	66
Stopping of the Stomach	1		29	36 13	3 10	4 17	1 66		28 1	45 12 61 13		The same	192	63	157 14	19 8	6 104	114	132	371	445 7	21 61	3 671	644 40	309
Surfet Swine-Pox	10	4	4	31			1	4	2	1	1 1	2		5	8,	4	0 432	454	539	1207	1751 26	13 1		915 1819	
Teeth and Worms		67 9		140 5	98 70	9 90	691	1131 8	111	98 87	8 1036	839	1008	440	506 3	35 47 14 3	4 2	3 15	27		68	65, 10			8 24
Tiflick Thrufh		62	47		6		1		1	5	57 6		1	15	23	17 4	0 2	al r	34	3	95	93 16 1	7 27	123 1 69 1	5 21
Vomiting		1	6	3	65 8	85 8	6 53	14	7	27 1	16 1		10	19	31	28 3	27 1	9 28	27			74 42	4 224	12	4 8
Worms Wen	1	147	107	1	3	2	6 53	1	1	9	1	2	1 1	1	-1	1	62 5	8 62	-0	34	221 2	33	2 4	4 6	3 4
Suddenly	1	1	1	1	1	1	1	1	- 1	1	1	1	1	63	59	371	021 3		1		1		1	13419	229.
																				Cable	at page :				

 1854: John Snow maps London Cholera outbreaks and finds that they are clustered around a single pump; it was found that a cesspit was leaking into that pump

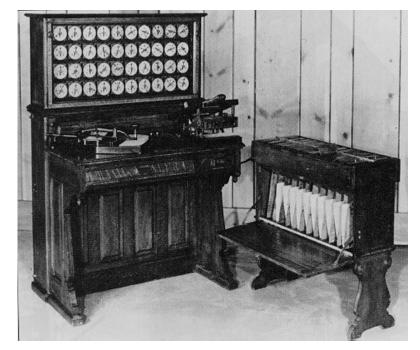


 1880: US Census Bureau estimates it will take eight years to process the data collected in the 1880 census, and over 10 years to process 1890 census data

Hollerith Tabulating Machine

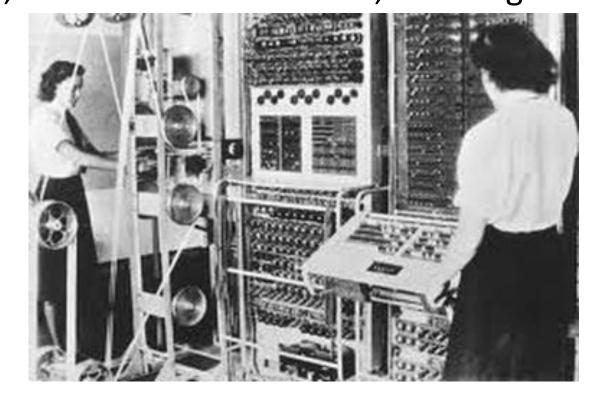
(punch card tabulation) reduces to ~3 months





• **WW2**: British invent the Colossus machine to scan for patterns in intercepted Nazi codes. Scans 5,000 characters a second, reducing

workload from weeks to hours



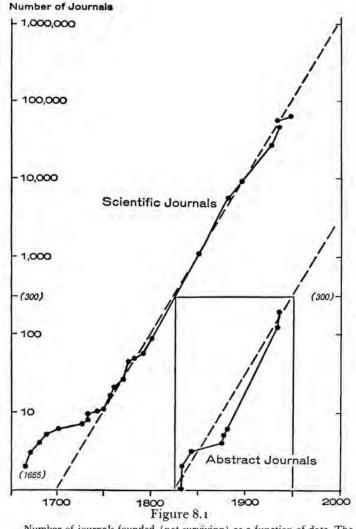
• 1944: Librarian Fremont Rider @ Wesleyan estimates American university libraries doubling in size every 16 years:

"the Yale Library in 2040 will have approximately 200,000,000 volumes, which will occupy over 6,000 miles of shelves...

[requiring] a cataloging staff of over six thousand persons"

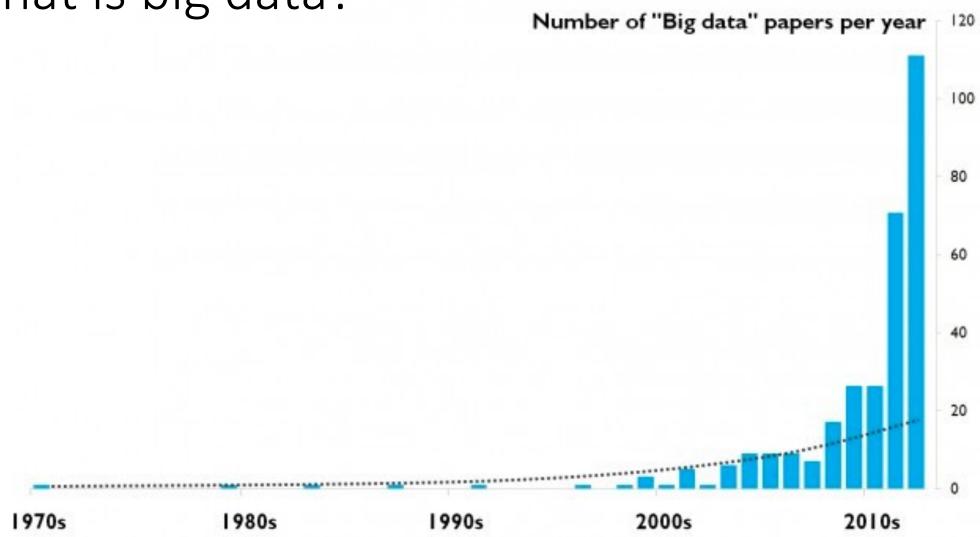
 1961: Derek Price shows # of new scientific journals growing exponentially rather than linearly

"each advance generates a new series of advances at a reasonably constant birth rate, so that the number of births is strictly proportional to the size of the population of discoveries at any given time"

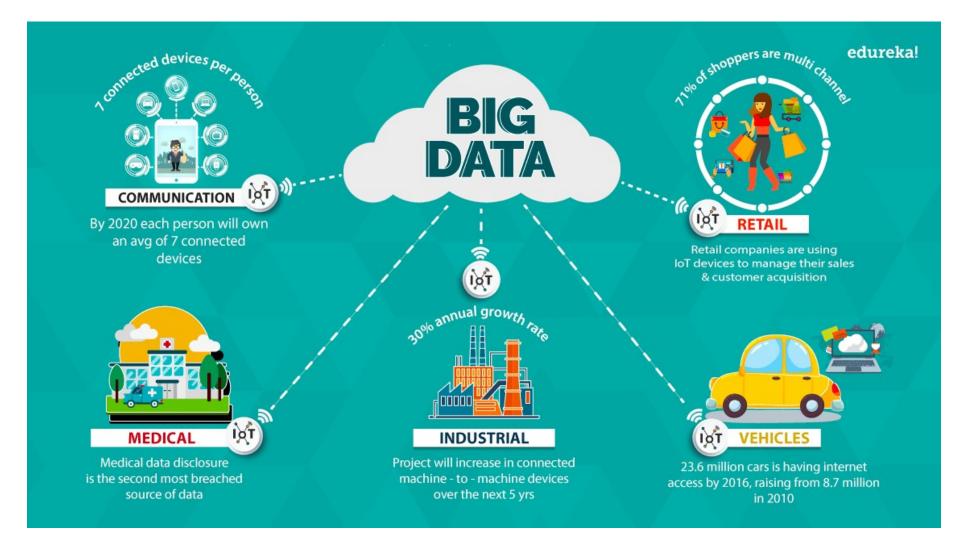


Number of journals founded (not surviving) as a function of date. The two uppermost points are taken from a slightly differently based list.

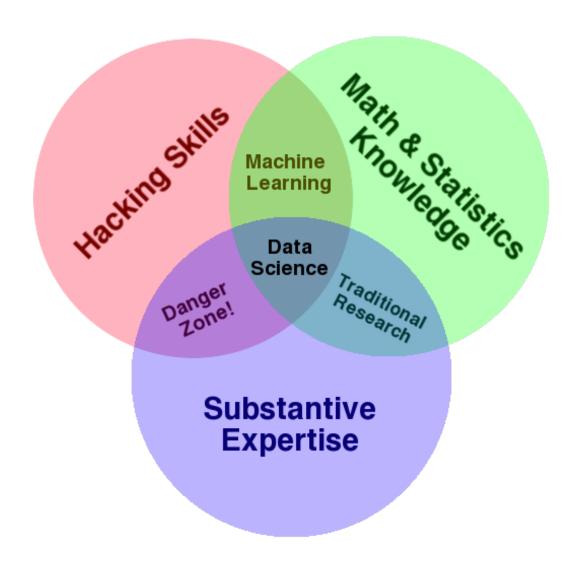
- 1969: ARPANET kicks off the Internet
- 1977: First personal computers come on the market
- 1989: Tim Berners-Lee introduces the concept of the World Wide Web and the underlying protocols that support it (HTML, URL, HTTP)
- 1993: CERN announces WWW will be free for everyone to develop and use
- 1990s- 2000s: The explosion of the internet
- 2005: Roger Mougalas coins term 'Big Data', referring to the scale of data that is nearly impossible to manage and process w/ available tools



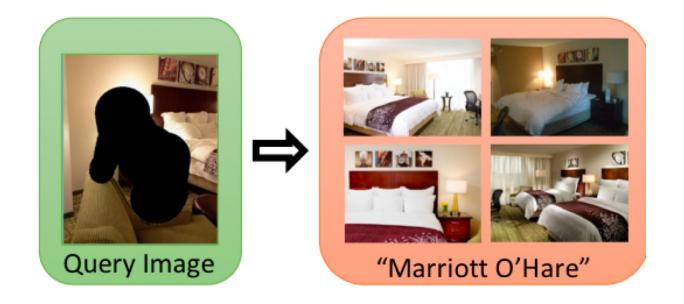
### Where is big data?



#### What is data science?



### The Big Data that I Work With















Settings Tools

About 459 results (0.17 seconds)



#### How to help AI find human trafficking victims

Three years ago an app hit the market called "Traffickcam." It's a simple. free app published by Exchange Initiative that lets you upload a picture .. Feb 12, 2019





#### Your hotel room photos could help catch sex traffickers

That's where TraffickCam comes in. It's a simple phone app that uses crowdsourced snapshots of hotel rooms to help law enforcement locate .. Mar 20, 2017





#### You can help stop human trafficking with the TraffickCam

In a world where the phrase "oh god, not another app" often springs to mind, along with "Yeah, yeah, I'm sure you want to make a world a better. Jun 25, 2016



#### top Washington Post

#### An incredibly simple way your phone may help save sex trafficked ...

She said the TraffickCam app, which is available for iOS and Android, isn't going to solve the problem, but it's one more tool that could help. Jul 1, 2016





#### Researchers Create Hotel-Recognition System to Aid **Human Trafficking Investigations**

... as crowdsourced images from the mobile app TraffickCam, which asks users who are traveling to take photos of their hotel rooms and submit .. Feb 5, 2019





Taking Pictures of Your Hotel Room Could Help Stop **Human Trafficking** 



 Learn the data science tools to collect, clean, understand, manipulate big data

• Exploration, Prediction and Inference

- Learn the data science tools to collect, clean, understand, manipulate big data
- Exploration, Prediction and Inference
  - Exploration: Identify patterns in information
    - Tools: Visualization + Descriptive Statistics

- Learn the data science tools to collect, clean, understand, manipulate big data
- Exploration, Prediction and Inference
  - Exploration: Identify patterns in information
    - Tools: Visualization + Descriptive Statistics
  - Prediction: make informed guesses about what we want to know, based on the patterns that we identified
    - Tools: Machine Learning + Optimization

- Learn the data science tools to collect, clean, understand, manipulate big data
- Exploration, Prediction and Inference
  - Exploration: Identify patterns in information
    - Tools: Visualization + Descriptive Statistics
  - Prediction: make informed guesses about what we want to know, based on the patterns that we identified
    - Tools: Machine Learning + Optimization
  - Inference: quantify our certainty about our predictions
    - Tools: Statisticals Tests + Models

Intro to Python,
Data Types,
Sequences



My strongest memory of learning to code is sitting thru my very first lecture silently panicking because while the FOR loop concept made perfect sense, I had no idea what I was supposed to do with it. I knew 'type it into Word' was wrong, but didn't know how to ask the Q...

### Programming / Coding / Hacking

- Instruct a computer to carry out tasks
  - "Add 2 + 4"
  - "Round 3.14159265"

### Programming / Coding / Hacking

- Instruct a computer to carry out tasks
  - "Add 2 + 4"
  - "Round 3.14159265"
- Combine multiple tasks into programs
  - "Add 2 + 4, then round 3.14159265, add the outputs together"
  - "Read in a file with the stock prices for a stock over the last 5 years and compute the average return"

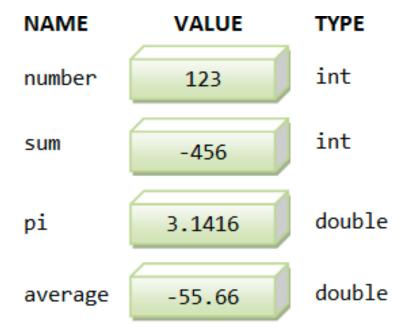
### Programming / Coding / Hacking

- Instruct a computer to carry out tasks
  - "Add 2 + 4"
  - "Round 3.14159265"
- Combine multiple tasks into programs
  - "Add 2 + 4, then round 3.14159265, add the outputs together"
  - "Read in a file with the stock prices for a stock over the last 5 years and compute the average return"
- Write these instructions in a language the computer can understand

#### • Expressions

<b>Expression Type</b>	Operator	Example	Value
Addition	+	2 + 3	5
Subtraction	-	2 - 3	-1
Multiplication	*	2 * 3	6
Division	/	7 / 3	2.66667
Remainder	%	7 % 3	1
Exponentiation	**	2 ** 0.5	1.41421

Variables



A variable has a <u>name</u>, stores a <u>value</u> of the declared <u>type</u>

Operations

$$a = 1$$
  
 $b = 2$   
 $c = a + b$ 

Operations

$$a = 1$$
  
 $b = 2$   
 $c = a + b$ 

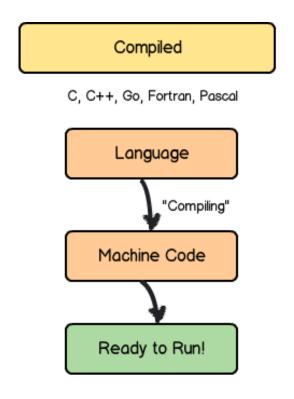
Functions (named operations)

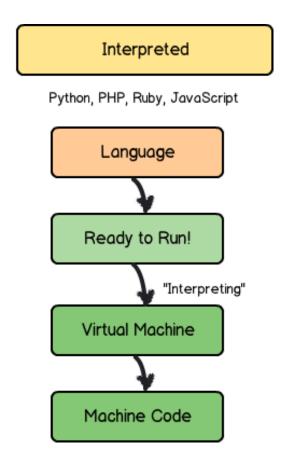
```
max (2, 2 + 3, 4)
give_everyone_in_class_an_a(class_list)
```

• Free & widely used (including in CSCI 1300)

- Free & widely used (including in CSCI 1300)
- Compatible w/ different systems

- Free & widely used (including in CSCI 1300)
- Compatible w/ different systems
- Interpreted rather than compiled





- Free & widely used (including in CSCI 1300)
- Compatible w/ different systems
- Interpreted rather than compiled
- Dynamically typed

```
Static typing:
String name; Variables have types
name = "John"; Values have types
name = 34; Variables cannot change type
```

```
Dynamic typing:

var name; Variables have no types

name = "John"; Values have types

name = 34; Variables change type dynamically
```

- Free & widely used (including in CSCI 1300)
- Compatible w/ different systems
- Interpreted rather than compiled
- Dynamically typed
- Built in memory management

- Free & widely used (including in CSCI 1300)
- Compatible w/ different systems
- Interpreted rather than compiled
- Dynamically typed
- Built in memory management
- Lots of built in modules + tons of very useful, well developed external libraries (NumPy, Pandas)

- Free & widely used (including in CSCI 1300)
- Compatible w/ different systems
- Interpreted rather than compiled
- Dynamically typed
- Built in memory management
- Lots of built in modules + tons of very useful, well developed external libraries (NumPy, Pandas)
- Readable!

# Hello World: Python vs. Java

A "Hello, World!" program generally is a computer program that outputs or displays the message "Hello, World!". Such a program is very simple in most programming languages, and is often used to illustrate the basic syntax of a programming language. It is often the first program written by people learning to code.<sup>[1][2]</sup>

https://en.wikipedia.org/wiki/%22Hello, World!%22 program

```
def hello_world():
    print("Hello, world!")
hello_world()
```

```
public class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello, world!");
    }
}
```

#### Whitespace in Python

```
def hello world():
   print("Hello, world!")
 hello_world()
def hello_world()
print("Hello, world!")
hello_world()
```

```
public class HelloWorld {
   public static void main (String[] args) {
      System.out.println("Hello, world!");
   }
}
```

```
public class HelloWorld {
 public static void main (String[] args) {
  System.out.println("Hello, world!");
 }
}
```

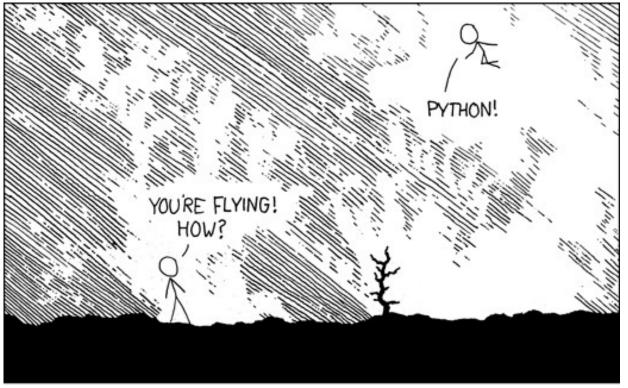
public class HelloWorld { public static void main (String) args) {System.out.println("Hello, world!"); }}

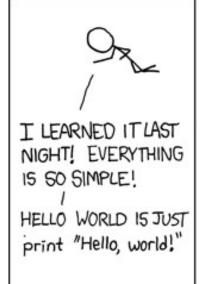
### Whitespace in Python

```
def hello_world():
    print("Hello, world!")
hello_world()
```

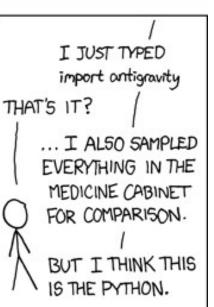
```
def hello_world()
print("Hello, world!")
hello_world()
```

```
>>> def hello_world():
        print("Hello, world!")
>>> hello_world()
Hello, world!
>>>
>>>
>>> def hello_world():
... print("Hello, world!")
  File "<stdin>", line 2
    print("Hello, world!")
IndentationError: expected an indented block
>>>
```









https://xkcd.com/353/