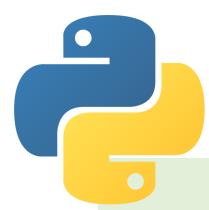


Python Programming Crash Course



Presented by Mohammad Taghizadeh



چگونه برنامه

نویس خوبی
شویم؟

چگونه پروژه
بگیریم و رزومه
بنویسیم؟

برنامه نویسی
بک اند با
پایتون (وب
فریمورک جنگو
و فلسف)

مقدمه ای بر
هوش مصنوعی
و یادگیری
ماشین در
پایتون

شیرجه عمیق
به پایتون و
هوش مصنوعی

نقشه راه
آموزش جامع
پایتون با تمرکز
بر هوش
مصنوعی

چرا پایتون و
هوش مصنوعی
در دنیای امروز

نکاتی در جهت
بهبود یادگیری
آموزش ها

چالش های
زبان برنامه
نویسی پایتون

پرسش و پاسخ

فلسفه پایتون ، انواع داده ای ، کار با رشته ها ، ساختمان های داده در
پایتون (list, dict, set, tuple, صف و پشته) و ...

تفکر برنامه نویسی رویه ای و کار با توابع و برنامه نویسی شی گرا در پایتون

کتابخانه های استاندارد ، ماژول ها و پکیج منیجر PIP

مدیریت فایل های متنی و باینری ، مدیریت خطاهای (try except) ، بررسی
عبارت های با قاعده Regex ها ، API نویسی و JSON در پایتون و ...

سرفصل و بینار



Monty Python

From Wikipedia, the free encyclopedia

"Pythonesque" redirects here. For the play by Roy Smiles, see *Pythonesque (play)*.

This article is about the comedy group. For their TV show frequently called *Monty Python*, see *Monty Python's Flying Circus*.

Monty Python (also collectively known as the Pythons)^{[2][3]} were a British surreal comedy troupe who created the sketch comedy television show *Monty Python's Flying Circus*, which first aired on the BBC in 1969. Forty-five episodes were made over four series. The Python phenomenon developed from the television series into something larger in scope and influence, including touring stage shows, films, albums, books and musicals. The Pythons' influence on comedy has been compared to the Beatles' influence on music.^{[4][5][6]} Regarded as an enduring icon of 1970s pop culture, their sketch show has been referred to as being "an important moment in the evolution of television comedy".^[7]

Broadcast by the BBC between 1969 and 1974, *Monty Python's Flying Circus* was conceived, written and performed by its members Graham Chapman, John Cleese, Terry Gilliam, Eric Idle, Terry Jones, and Michael Palin. Loosely structured as a sketch show, but with an innovative stream-of-consciousness approach aided by Gilliam's animation, it pushed the boundaries of what was acceptable in style and content.^{[8][9]} A self-contained comedy team responsible for both writing and performing their work, the Pythons had creative control which allowed them to experiment with form and content, discarding rules of television comedy. Following their television work, they began making films, including *Monty Python and the Holy Grail* (1975), *Life of Brian* (1979) and *The Meaning of Life*

Monty Python	
The Pythons in 1969: Back row: Chapman, Idle, Gilliam Front row: Jones, Cleese, Palin	
Medium	Television · film · theatre · literature · audio recordings
Nationality	British ^[1]
Years active	1969–1983, 1989, 1998–1999, 2002, 2013–2014
Genres	Satire · surreal humour · black comedy · physical comedy
Former members	Graham Chapman John Cleese Terry Gilliam Eric Idle Terry Jones Michael Palin
Website	MontyPython.com ^[2]

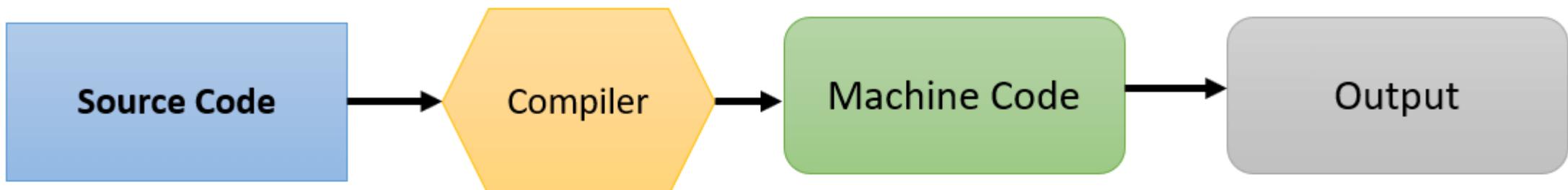
Guido
van
Rossum



هدف اصلی: سادگی و خوانایی بازی کد

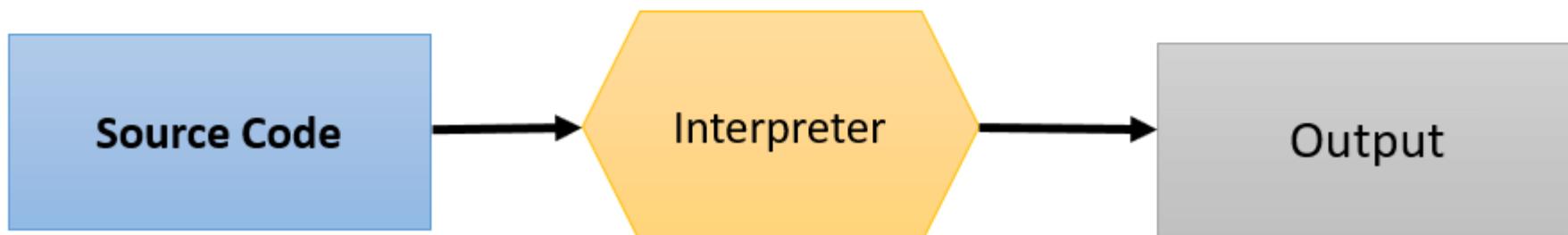


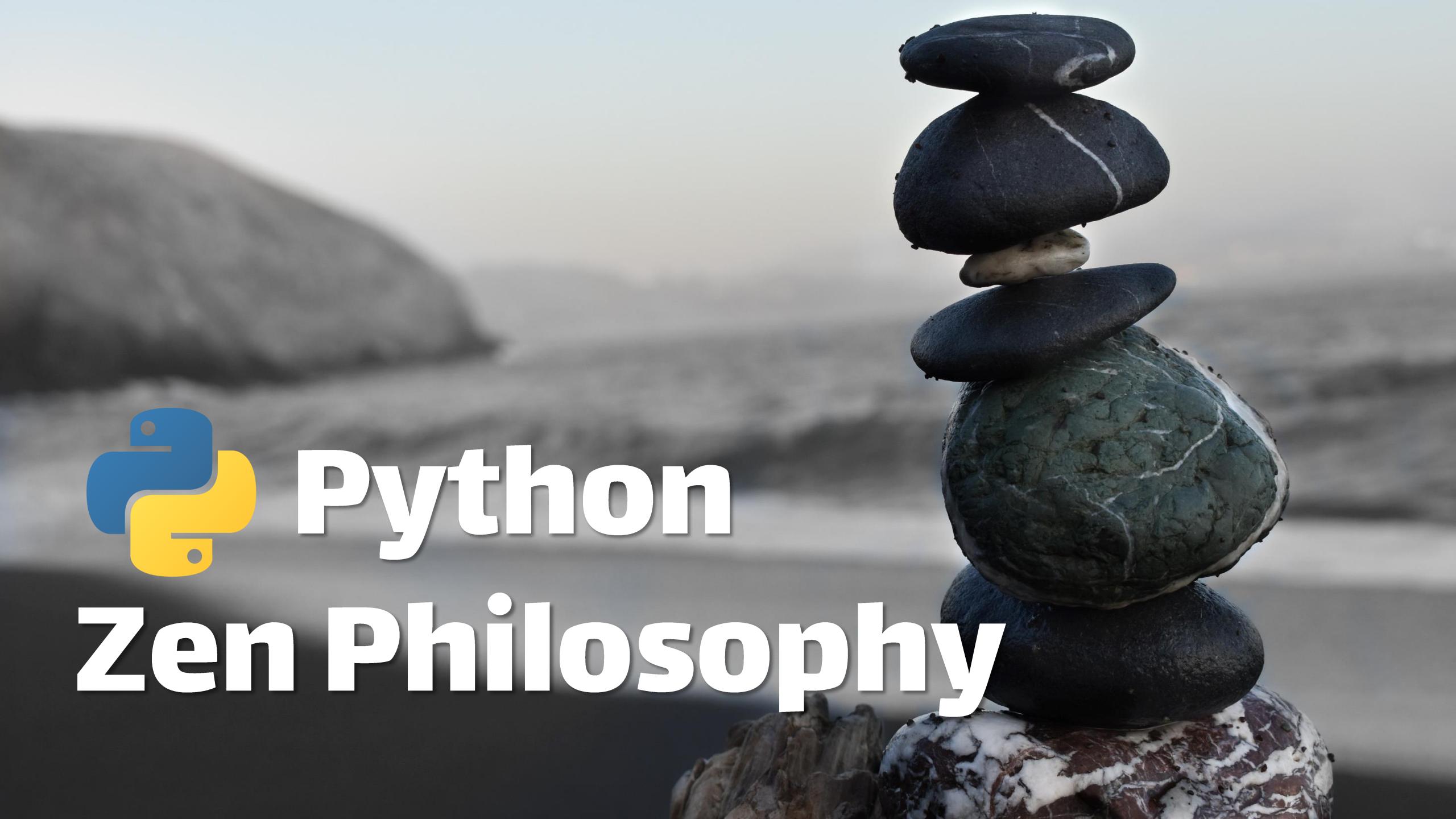
How Compiler Works



© guru99.com

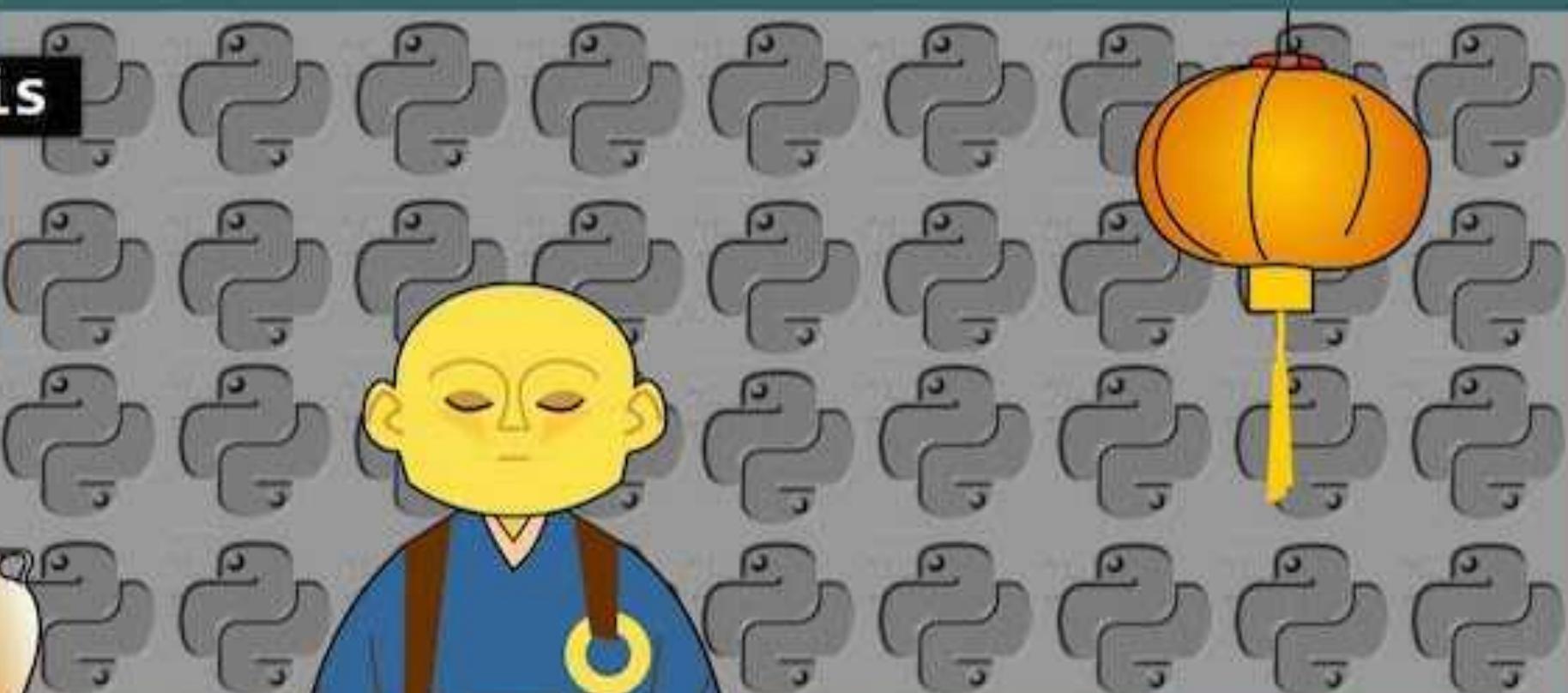
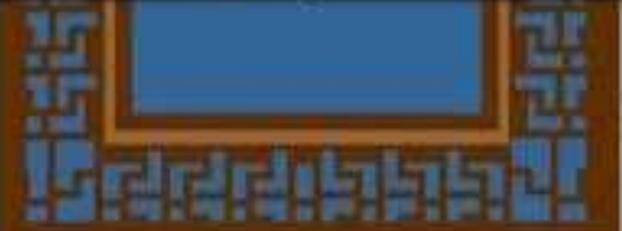
How Interpreter Works



The background of the image is a blurred photograph of a beach at sunset or sunrise. In the foreground, there is a stack of several dark-colored stones, possibly black lava rocks or polished pebbles, balanced in a loose pyramid shape. The stones have various textures and some white mineral streaks. The overall mood is peaceful and minimalist.

Python Zen Philosophy

```
>>> import this
```





```
>>> import this  
The Zen of Python, by Tim Peters
```

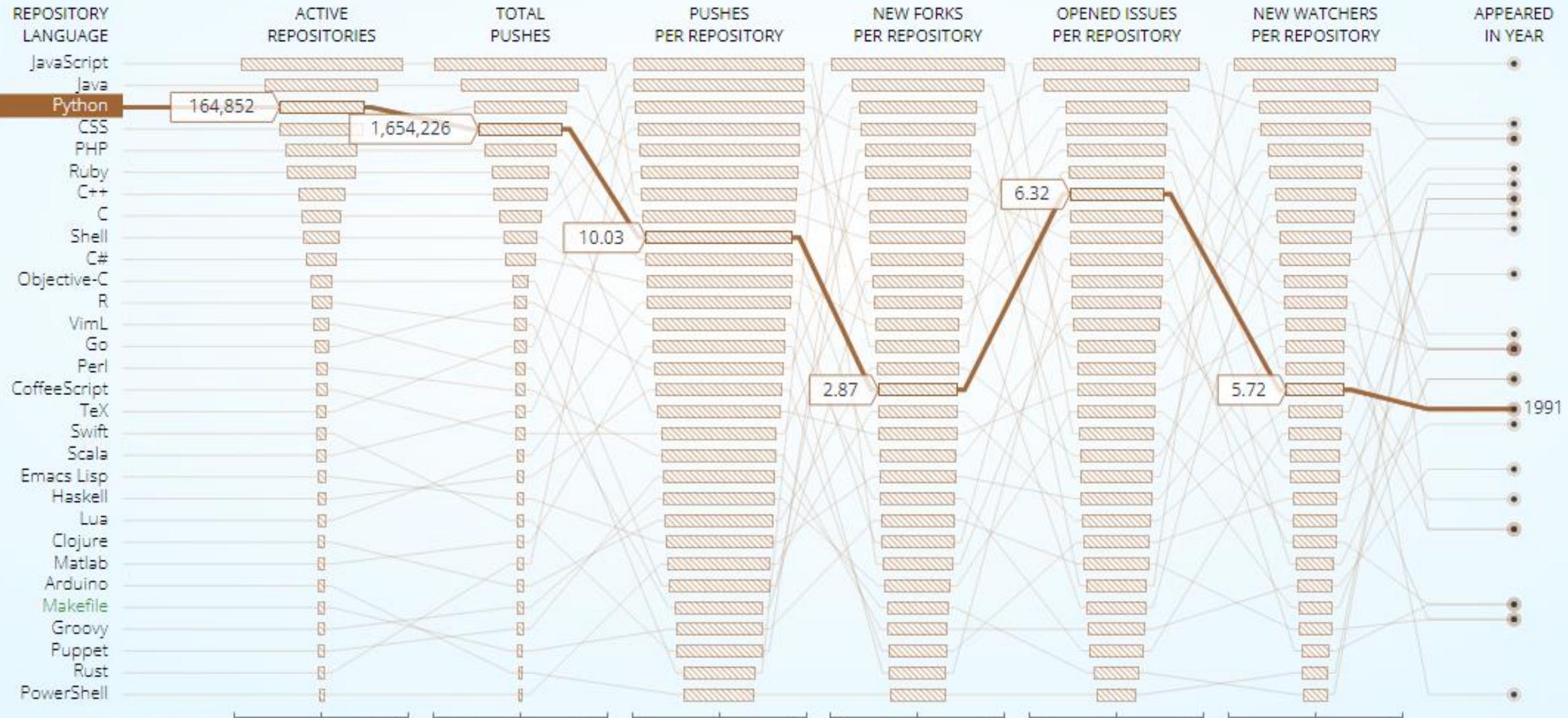
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!



Feb 2022	Feb 2021	Change	Programming Language	Ratings	Change
1	3	▲	 Python	15.33%	+4.47%
2	1	▼	 C	14.08%	-2.26%
3	2	▼	 Java	12.13%	+0.84%
4	4		 C++	8.01%	+1.13%
5	5		 C#	5.37%	+0.93%
6	6		 Visual Basic	5.23%	+0.90%
7	7		 JavaScript	1.83%	-0.45%
8	8		 PHP	1.79%	+0.04%
9	10	▲	 Assembly language	1.60%	-0.06%
10	9	▼	 SQL	1.55%	-0.18%



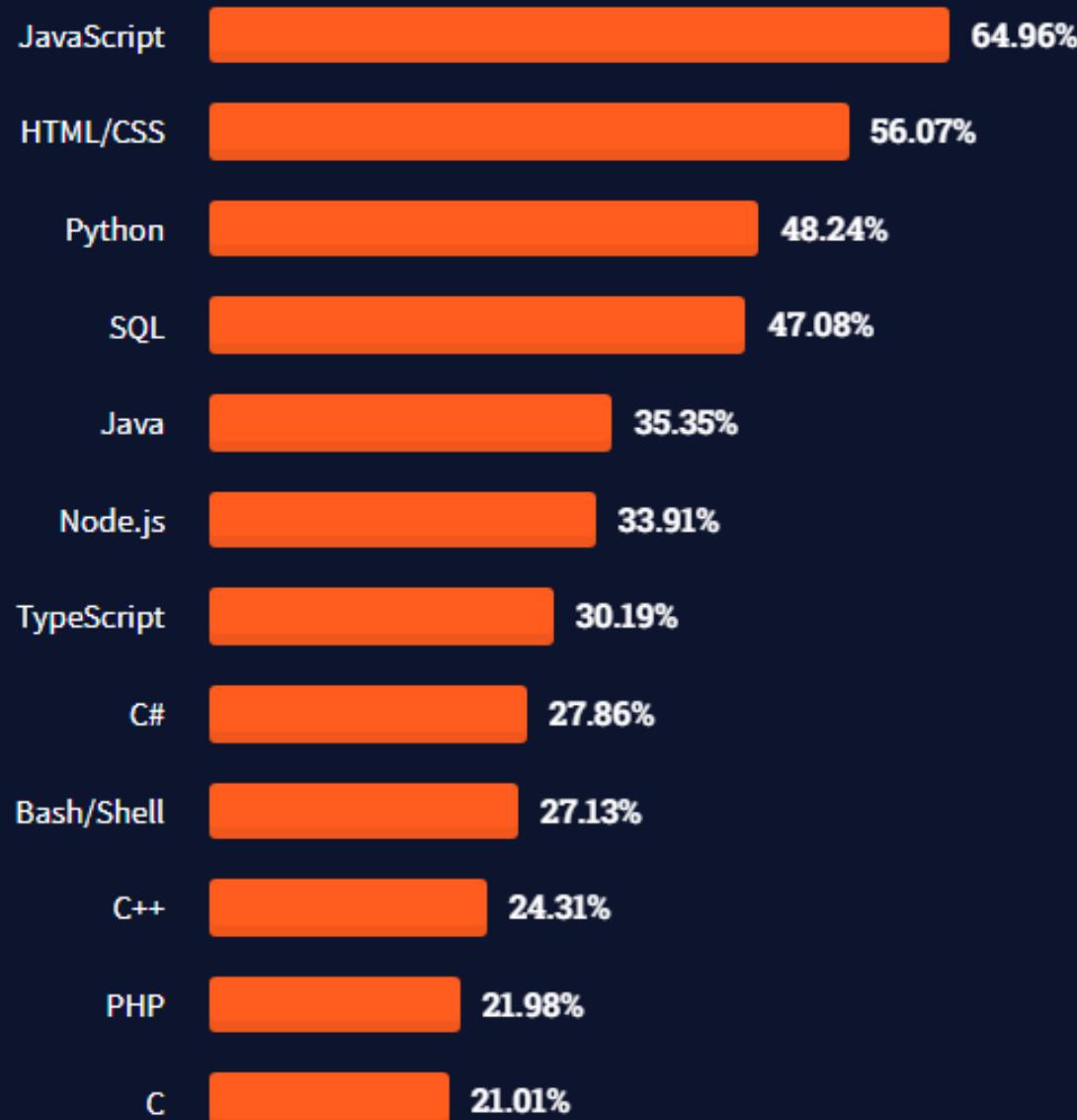
< Q4/14 >



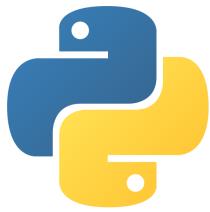
All Respondents

Professional Developers

83,052 responses



Python Beginner friendly 😊



False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	



```
class Main{  
    public static void main(String... args) {  
        System.out.println("Hello World!");  
    }  
}
```



```
print("Hello World!")
```



Python Community



Web Development



Flask

web development,
one drop at a time



django



Tornado

WEB2PY



Pyramid™

FastAPI



Bottle

Artificial intelligence



 TensorFlow

 NumPy

 pandas

 SciPy

 Keras

 scikit
learn

Vim

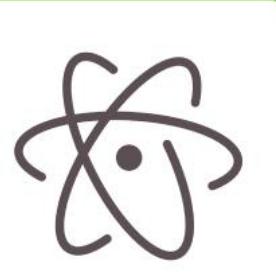
Spyder

Atom

Jupyter

Eclipse

Notepad++



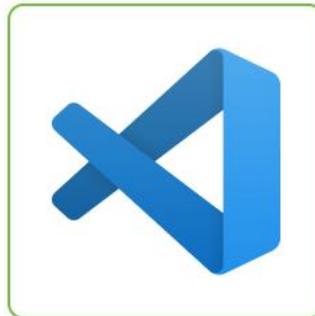
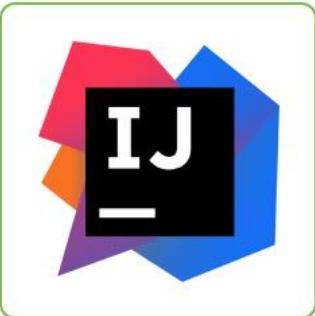
IntelliJ IDEA

PyCharm

Visual
Studio Code

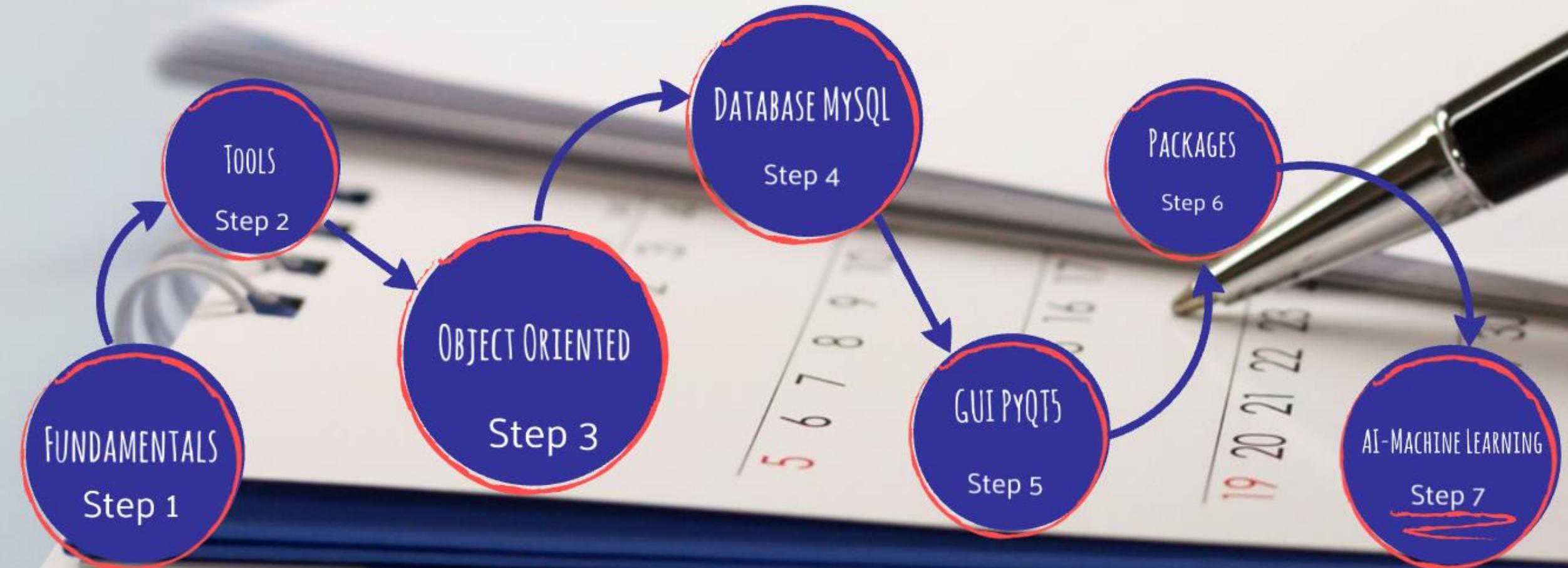
Sublime Text

GNU Emacs



Th

PyDev



Daneshjooyar.com



PYTHON PLANNING

<http://M-Taghizadeh.ir>





شیرجه عمیق به زبان پایتون

Outline

01-Data type

06-Set

11-Function

15-Exception
Handling

02-String

07-Dict

12-Lambda

16-Modules

03-Operators

08-if else

12-Generator

17-PIP

04-List

09-Match case

13-Files

18-Regex

05-Tuple

10-Loop

14-Pickle

19-Doc string



Files





متد `read` : کل فایل یا در صورت تعیین تعداد بایت، تا تعداد بایت مشخص شده را بازگشت می‌دهد.

متد `readline` : یک خط منفرد یا در صورت تعیین تعداد بایت‌ها تا آن خط را بازگشت می‌دهد.

متد `readlines` : کل افزار می‌کند فایل یا در صورت تعیین تعداد فایل، مقدار مشخص شده را بازگشت داده و به لیستی از خطوط

File Edit View Settings ?

PKETX EOT DC4 NUL NUL NUL BS NUL à-

GRxYÍ ETX NUL NUL ¥ BS NUL NUL VT NUL NUL NUL eulerlib.py UÓ0 DLE } n³b DLE BJô . ÄE ' oÀ ESC BÀ VT ZU+ · ¶ PMì DLE ' Úæi9¶ä»

ENOn6ö\íññä%E /éçQTÑNqL{] ÑySK-FFÖ%Uuw5} jr@~ -ô6çÂ

jÛö¶ -cmRSdäGSéºu0NSrÜDC3oS¥8ö\ôPcIE=HETBcDLEtö±ÙNvºúØé . æóULDC4É¢ÔUM | 5ÙÛ; É#WACKæ; @özR{ Ô²zÑâC4ª:0

-èi^¢BS %sÝT¶E#Ó; bQº@ DV±ÖT¢FACK¤. ðNAK) -EST¢E¤ùNULÈIñSO», >SÔ&DC3Æ0"i%¢EM

Qé^£Ñé (sEEêENOVPFSÙBSöjs8;¢SYNLGS-:_, hÀ!øÖGi"SUB÷¶} @uNAKBUGSÄ3ò<K>GS Ayà"éACKIGÔÆÒÁ

GSFÔlpY{ ¶esÜökÉí¢EMakeØEOSBSi«h] NAK?ÑYC\$#pÖ7s"VT%

(KSYNLª) ¶\9úØ%¶s@EOTVT } ¶s¹ôEn8SUBqRSÈVTäi"EM1"ha¥0ÙK0~í@NAK%4EOT%r

ihUSÈS' H\¥] 2Ù' { iEMú 3[a.M

59ÖSI SOH d&wDLE: ö-b±; °V, øÖLë38îPS?/B

| Éé [1 ÍVTò ; i¥mú4i : iCAN¹RS\Om-1 CAN BEL US ÜDC2 Y Tá STX CAN (6ÑÈcASYN · SYNhETXÖÄEö & | ÜÙACKÖi } ÷@' Í [kû , ETBëE5Öax f ÿ~ðwSTXM gNAK { ; \$NULg@%USÐÅSOHxÉÖ#Ès«%1ÈkiUDC2NULÈ; ESC SOH ' æZ STXÀiXe8 SUBØ2@ÅEMÖYDC4eî

US\ETX; Ùl&~ÿ6±%Y¤EOTé4IémJ69%RSüþRS SUBzåÁyd2IéYDC2SI SYNiñn. SIÄ%* ' ÑETB r+ÝP%CANEMgO, 5ACKIÆSOHq| u°ii

Euaàä<3í

¿@ö.húÁ; 5@ö.oFFn<US) > ÁÝÆÀqnpzv! k?pG- ' fàÂ

C

BEL@òt ENOj 0ÈÝBSQÓDC2É } -o' ETB { , 1D; CVTÈDXDC4ÖETX? «±Üþ¬B ' NUL »uP ` DC4VvVTzmñC5¥US SOH þúENO4EOT\k6. DC1]

Ý2; ÕÙSOKÔEN; ' AÁxUS\$%ØÙ>uy" wHC@-. ¿ðÄl, çó«í;

x2øÝ DC3 WÉÁl%> SO%¥K þ«çÖESCV!ç%; wrñºñDÍ¥EHDINUL PKETX EOT DC4 NUL NUL BS NUL FS DC4 Gx SOH1 ! ENO NUL NUL 5 VT

NUL NUL FF NUL NUL NUL eulertest.pyev] S1 DLE } ¶; " Õ@USÝDLDC2ÁBELñETX EOT DC1 QÁÇRÍU «ý¢%EEÜiIwENOÉ%ç\$3sf2þ»n

r x%USNf; xää>iííqymø4°ççíptÉÍE%ù5~ÝiÜi; 9éýi EOT; nö%þþ: _OÖÖ×q<<°þþñäç | Sô| %; >, ; FF x BEL ÜþpDLE - SI o \u DC3

%ç-çöOùiµÙ^; é6ïy° LÖÜÄþ8: ûÚ-?>päz; L&ÖÙmëÝiêÑäí, USESCYpÙÉGS IpACK-Ø?ü<Àùrw°rk\$ BEL 7Äjöäö» SI ïß%¢CAN

ÛóiJ| ÜÙiWç

<ÜACKÛÓÅVfÓÃiG>@þLçp;] SUB àê

| SqyFS{ | ùÈ; -, ?çåACKÑþbn\$óÄy' åå°<Öy| i; »q} .

¶i°Q_ : FS x>q6} ETX-aëLå# | ÕUS ·=y³Ûé! »ùón5û7ø=ETBñ-WW SI ÜtêÖÝ? ° USþ DLE ; O: | iÝ»ç^<yùJRS | (ÄEç{ Qê, ç¥ffENO=

"æFÈ-Ù-EOT | p¹ýPB; îGnJ1 ACK@ÓBEL 8q%DC1DC1EOT x\ FF% æ} 3ÖJfìþ (SYN * BELN!w | >DC4

W5È!1%BS3° ; ÜEMhSYN³D-ÄJp DC1 EM+gi¹SCHCANÍø» XæCa DC3å5ðaQ4 kâLþRÅÈ2 ESC * "4 EOT è°æq ÜEOT D+Sj

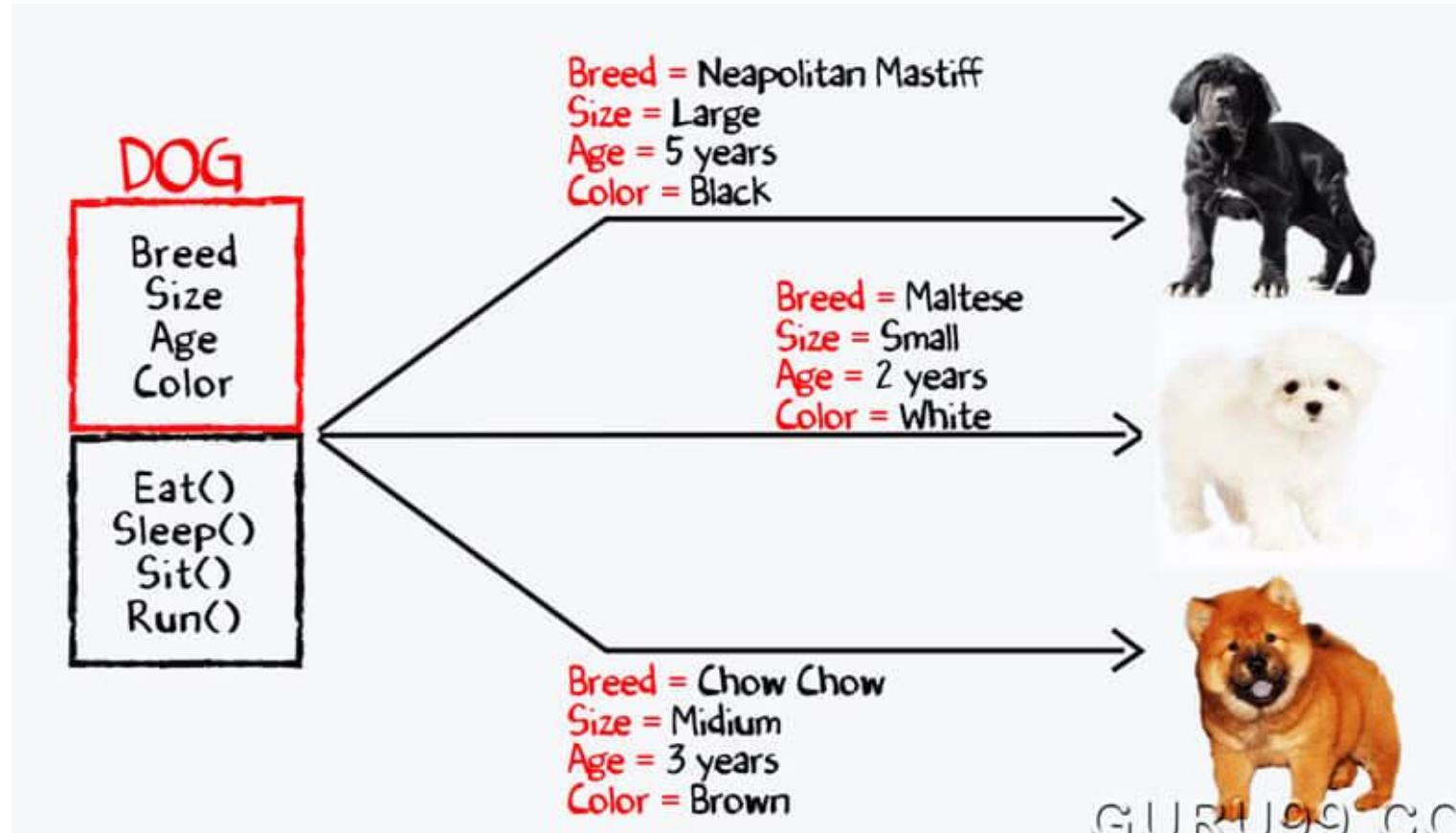
h*Ll\$r ; !

2J EOT | PZ%Ø' Tj1! S^HÈ°@ \LDY «ÑBS { ESC An^%<7 Èñ%SMön DLE þøJ DC1 SOH FF k NAK # wçQO= DC3 U" SIFF \$ YÓ%vµ¶öì EMá ØFF

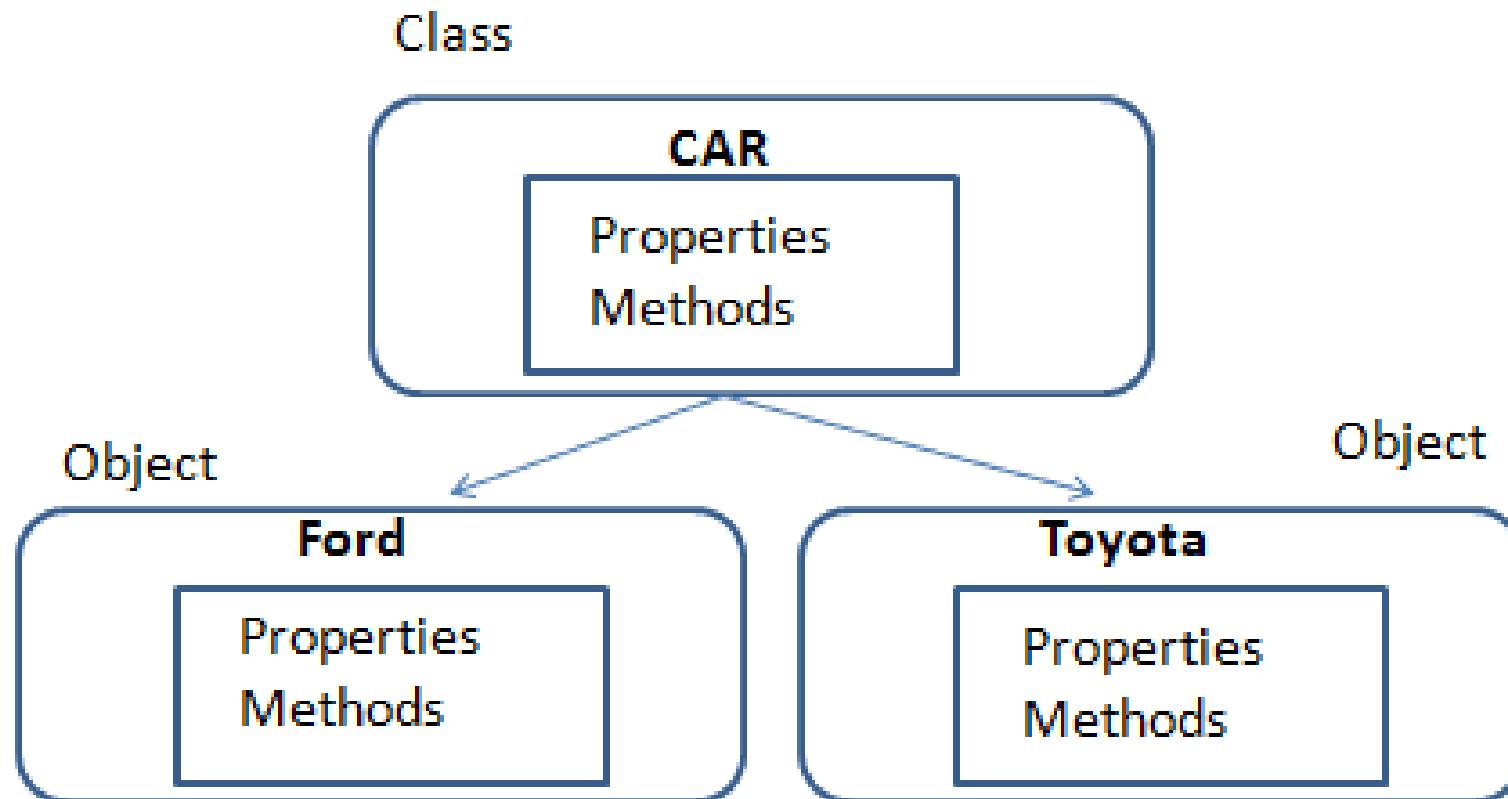


OOPS
in
Python

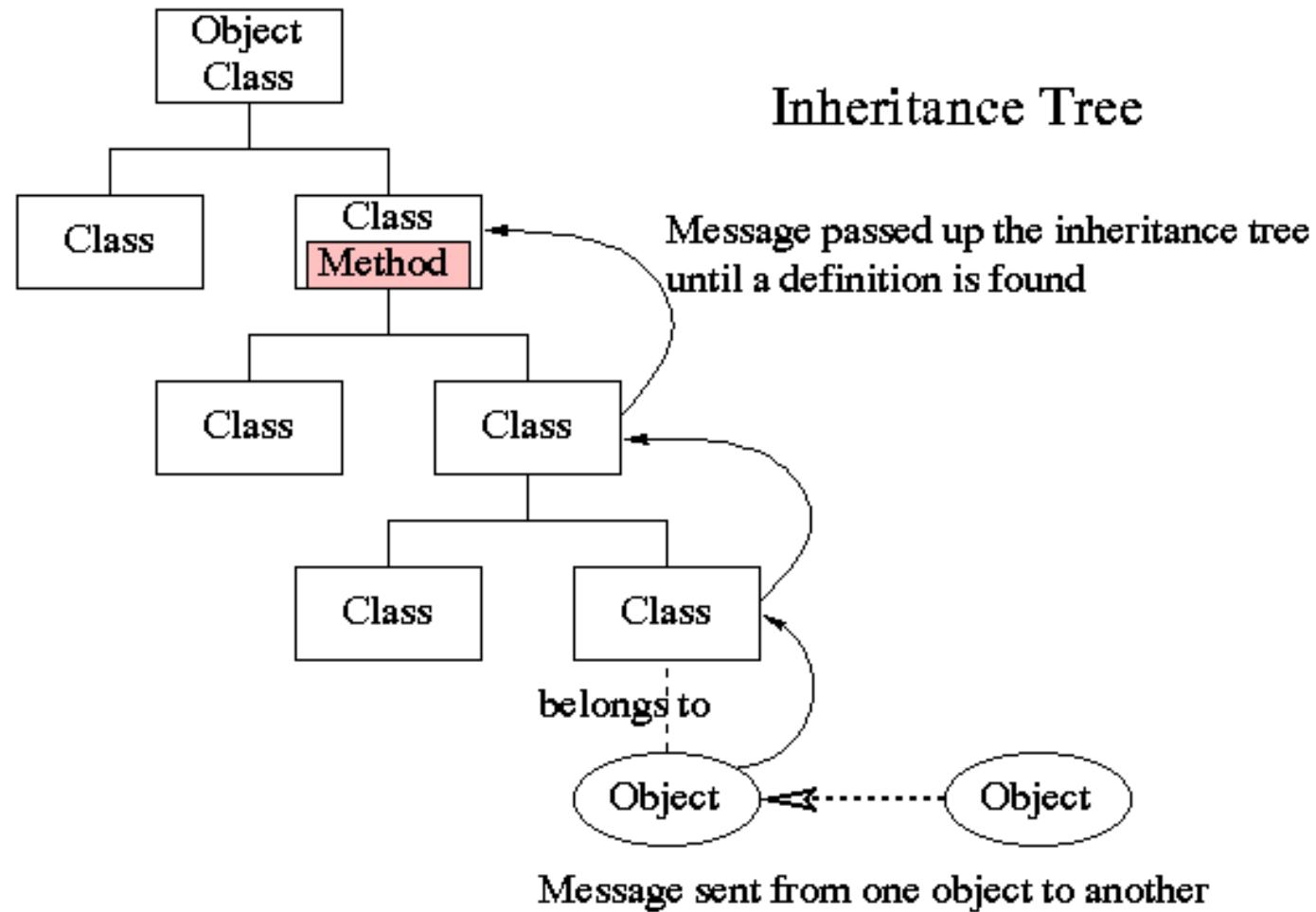
What Is OOP...?



Class, Object, ...?



Inheritance Architecture



4 Principal of OOP

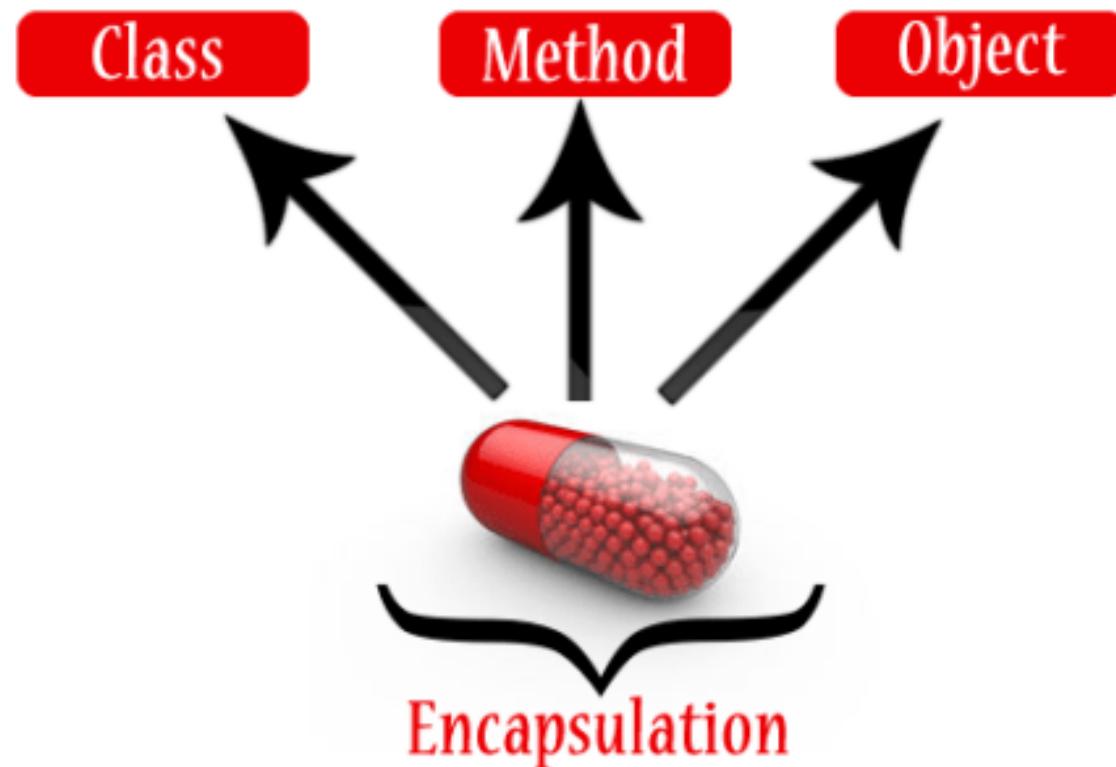
1 - Encapsulation

2 - Abstraction

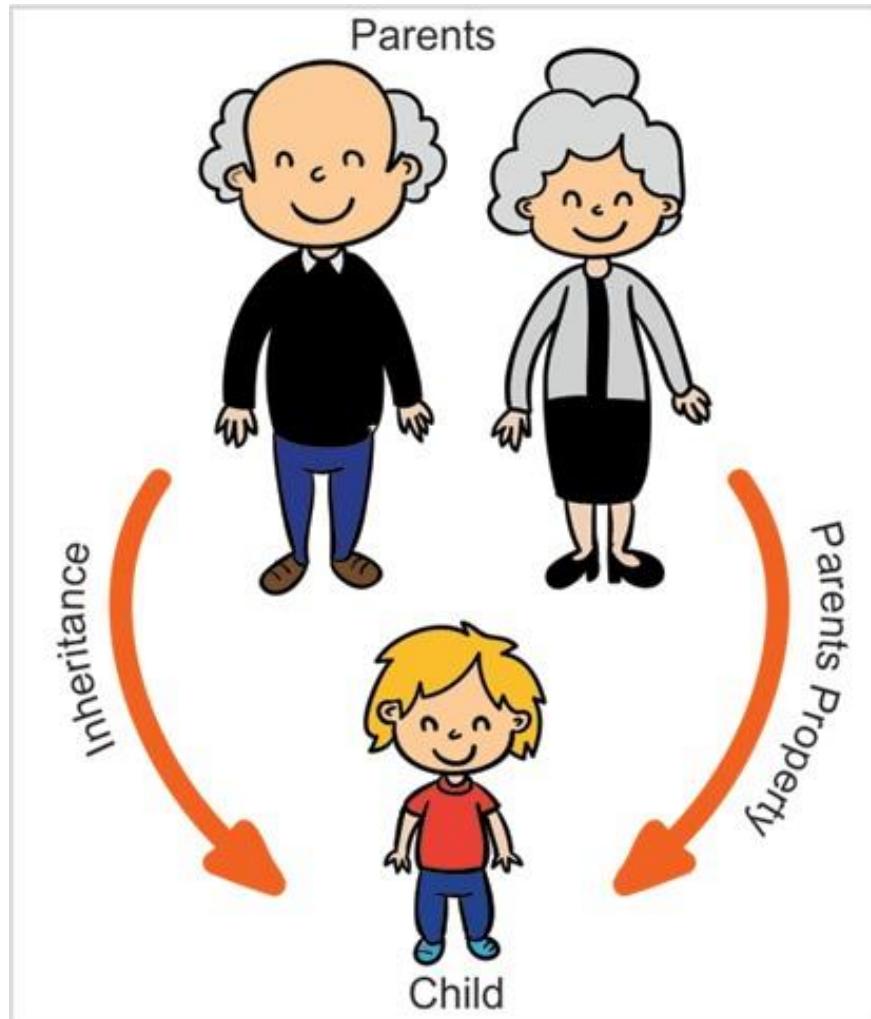
3 - Inheritance

4 - Polymorphism

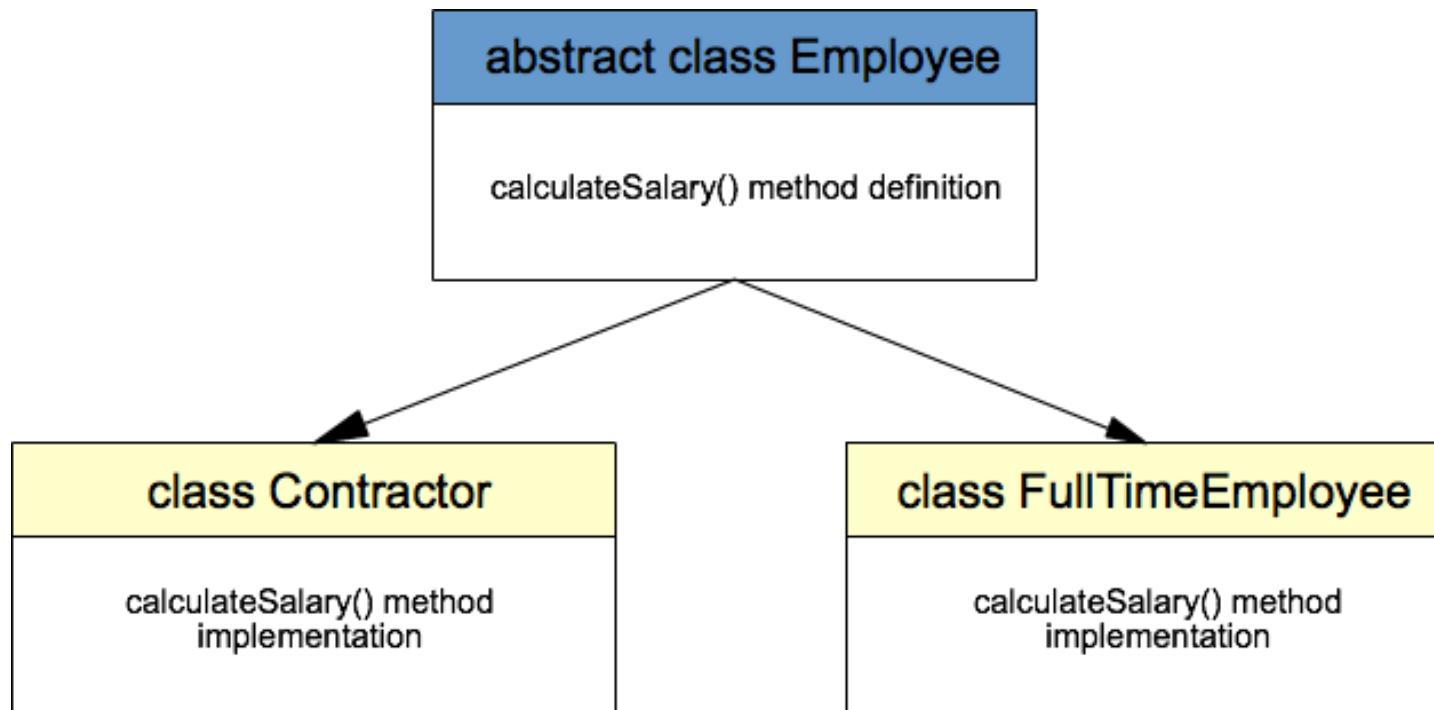
1 - Encapsulation



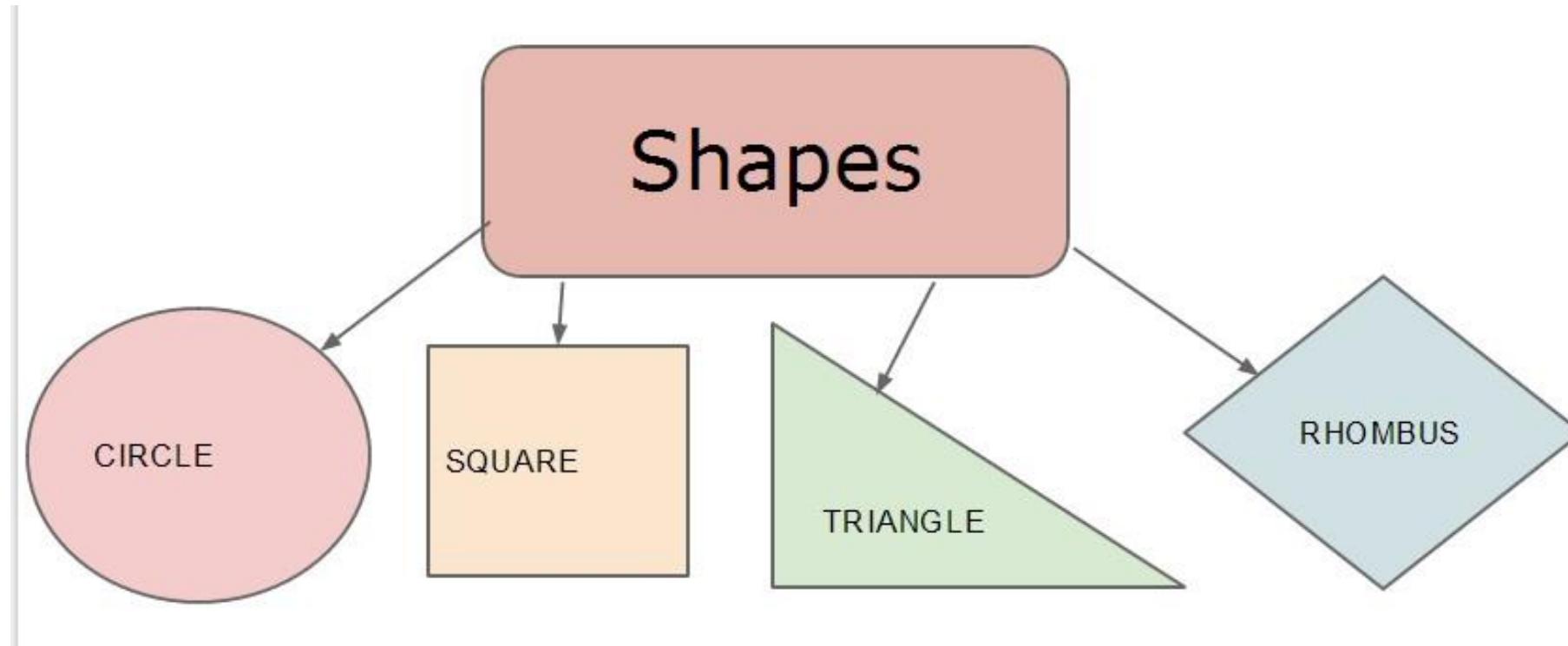
2 - Inheritance



3 - Abstraction



4 - Polymorphism



```
class Employee:  
    def __init__(self, name):  
        self.name = name  
    def display(self):  
        print('The name of employee is:', self.name)  
  
first = Employee('Rushabh')  
second = Employee('Dhaval')  
  
second.display()  
first.display()
```

Define a bicycle object prototype

attributes:

- speed
- gear

behaviours:

- speed up
- apply brake
- change gear

```
class bicycle:  
    ''' properties'''  
    # Class variables.  
    gear = 1  
    speed = 0  
  
    def __init__(self, gear, speed):  
        self.gear = gear  
        self.speed = speed  
  
    def speedUp(self, increase):  
        self.speed += increase  
  
    def changeGear(self,newGear):  
        self.gear = newGear  
  
    def applyBrake(self, decrease):  
        self.speed -= decrease
```



Machine Learning





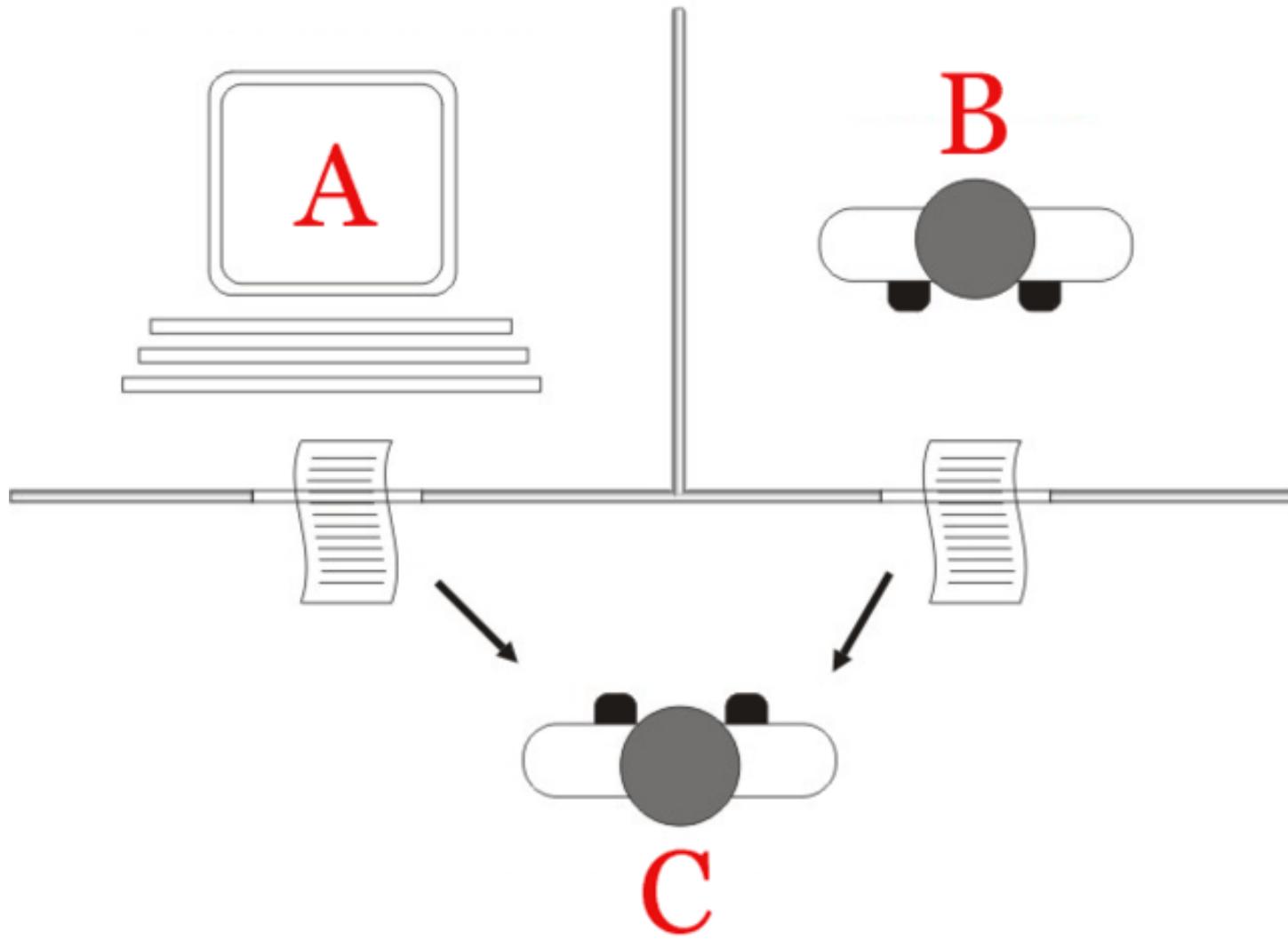
Thinking humanly

Thinking rationally

Acting humanly

Acting rationally

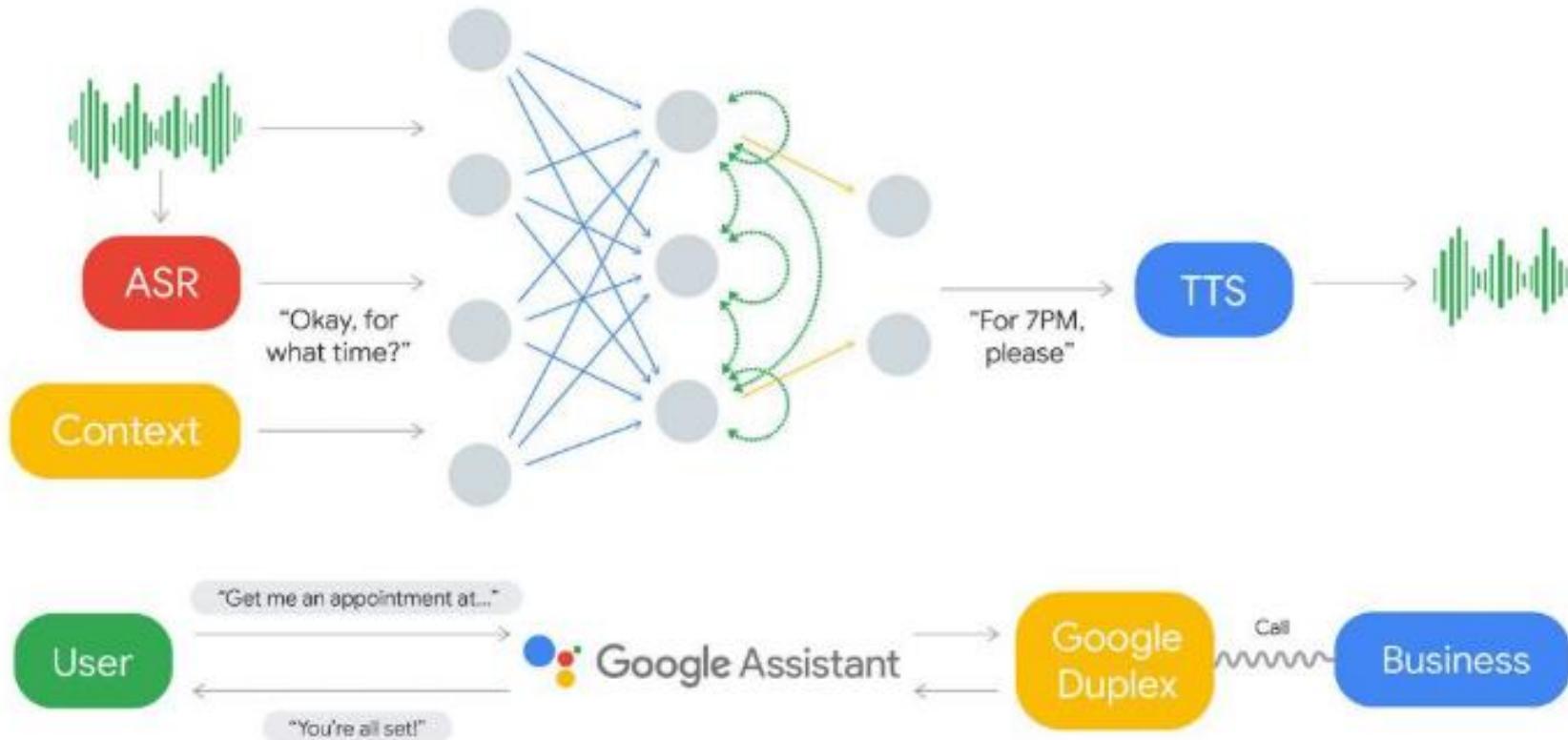
Artificial intelligence

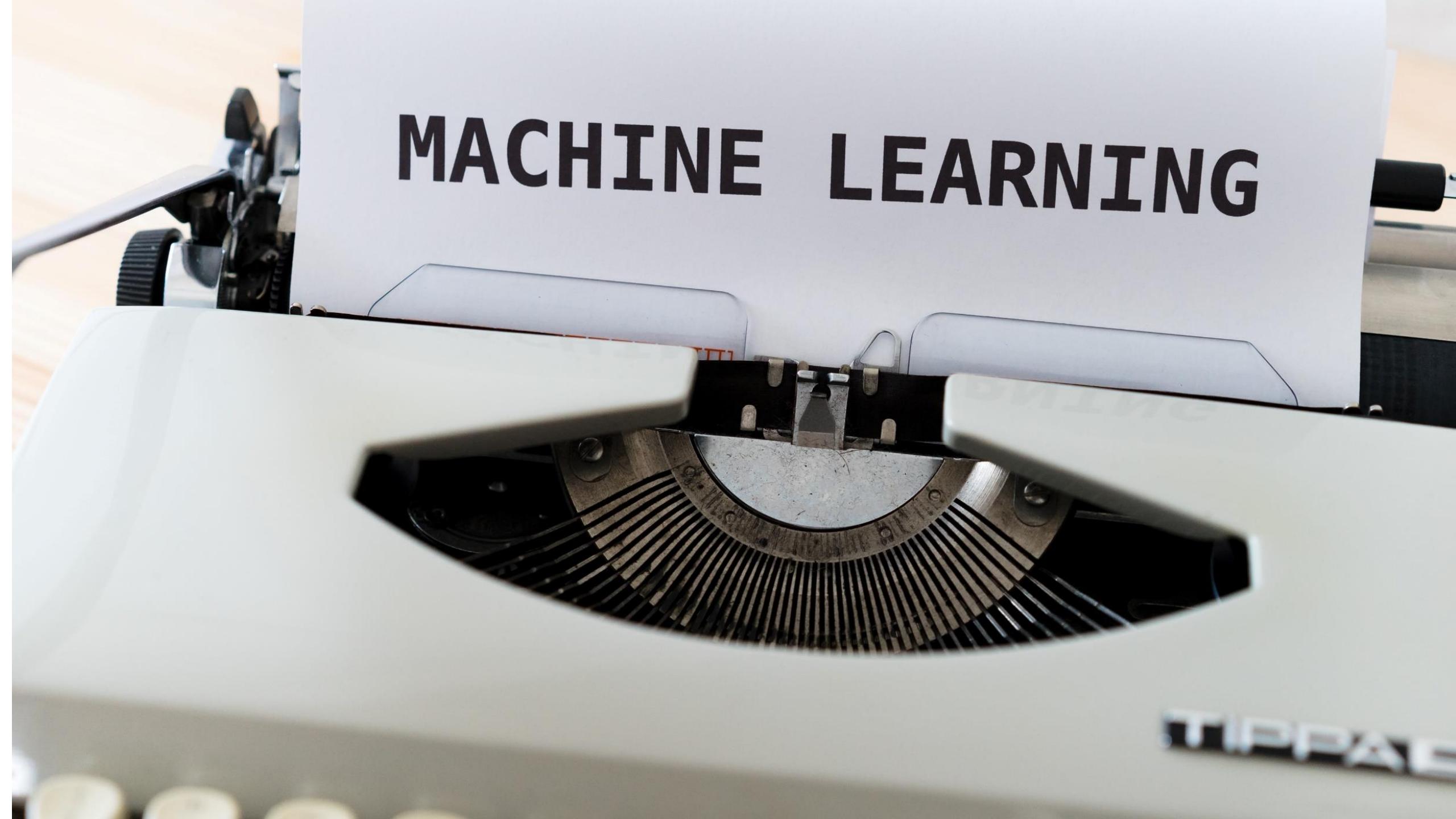


Did Google Duplex really pass the Turing Test?

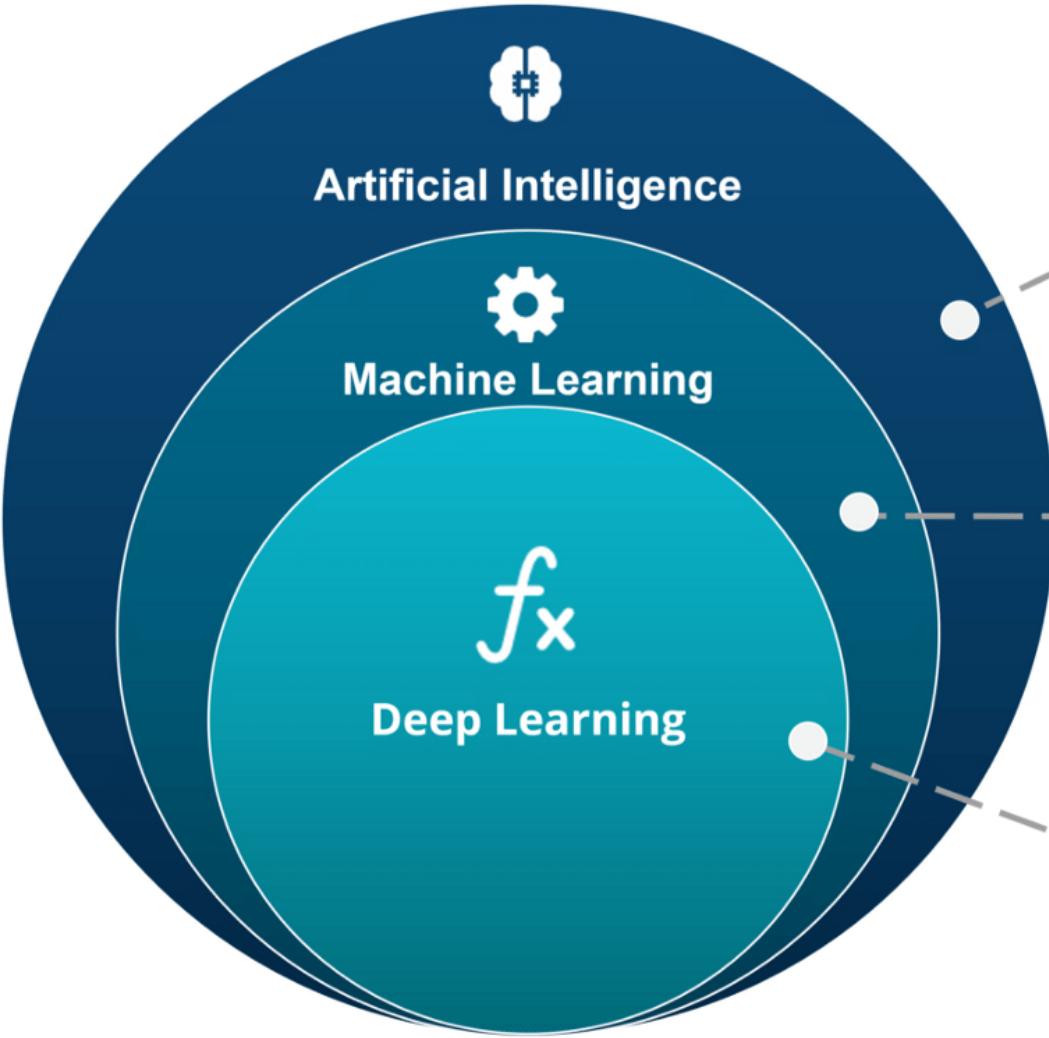


Pranjalya Tiwari · Sep 10, 2019 · 2 min read





MACHINE LEARNING



ARTIFICIAL INTELLIGENCE

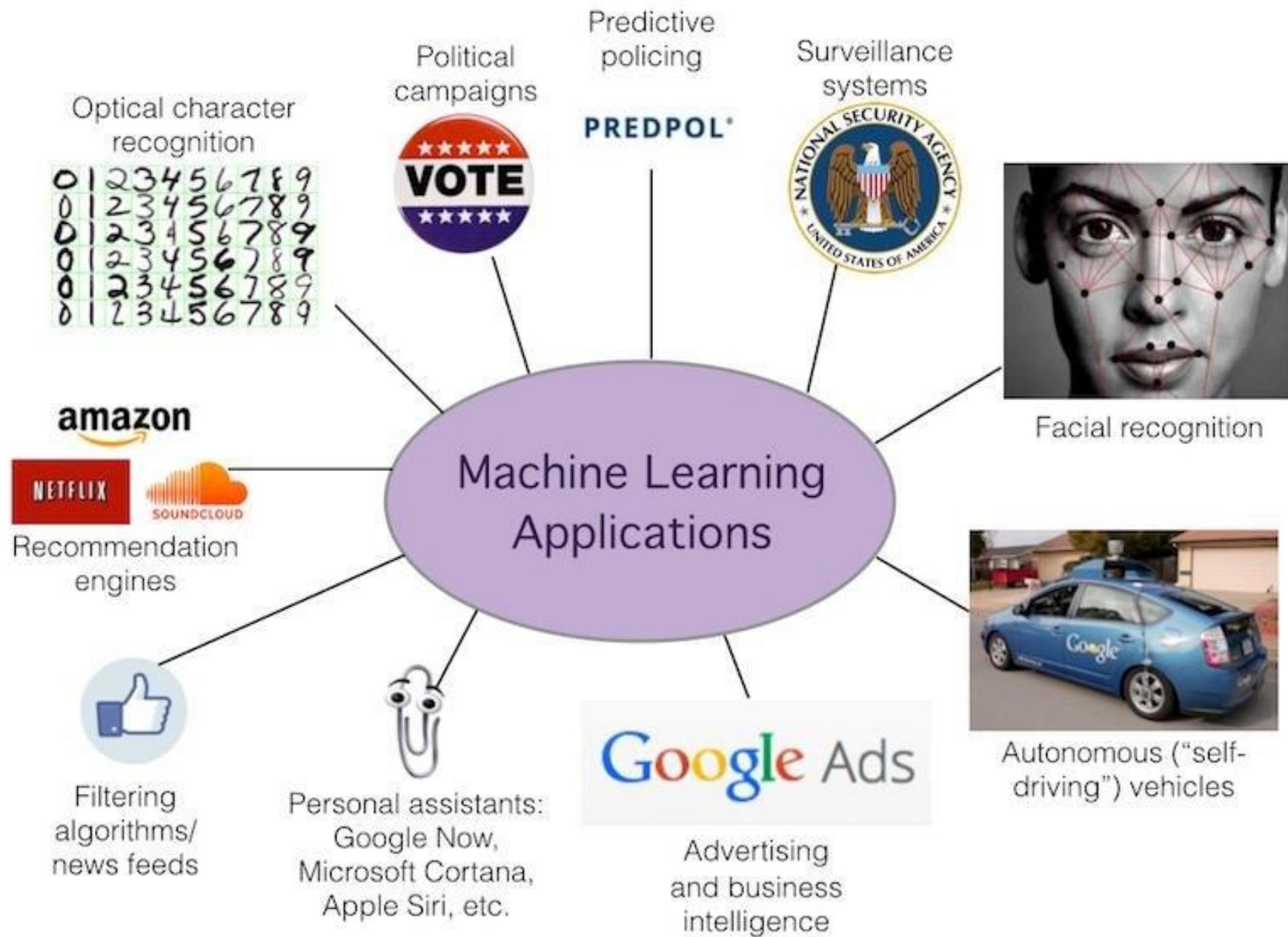
A technique which enables machines to mimic human behaviour

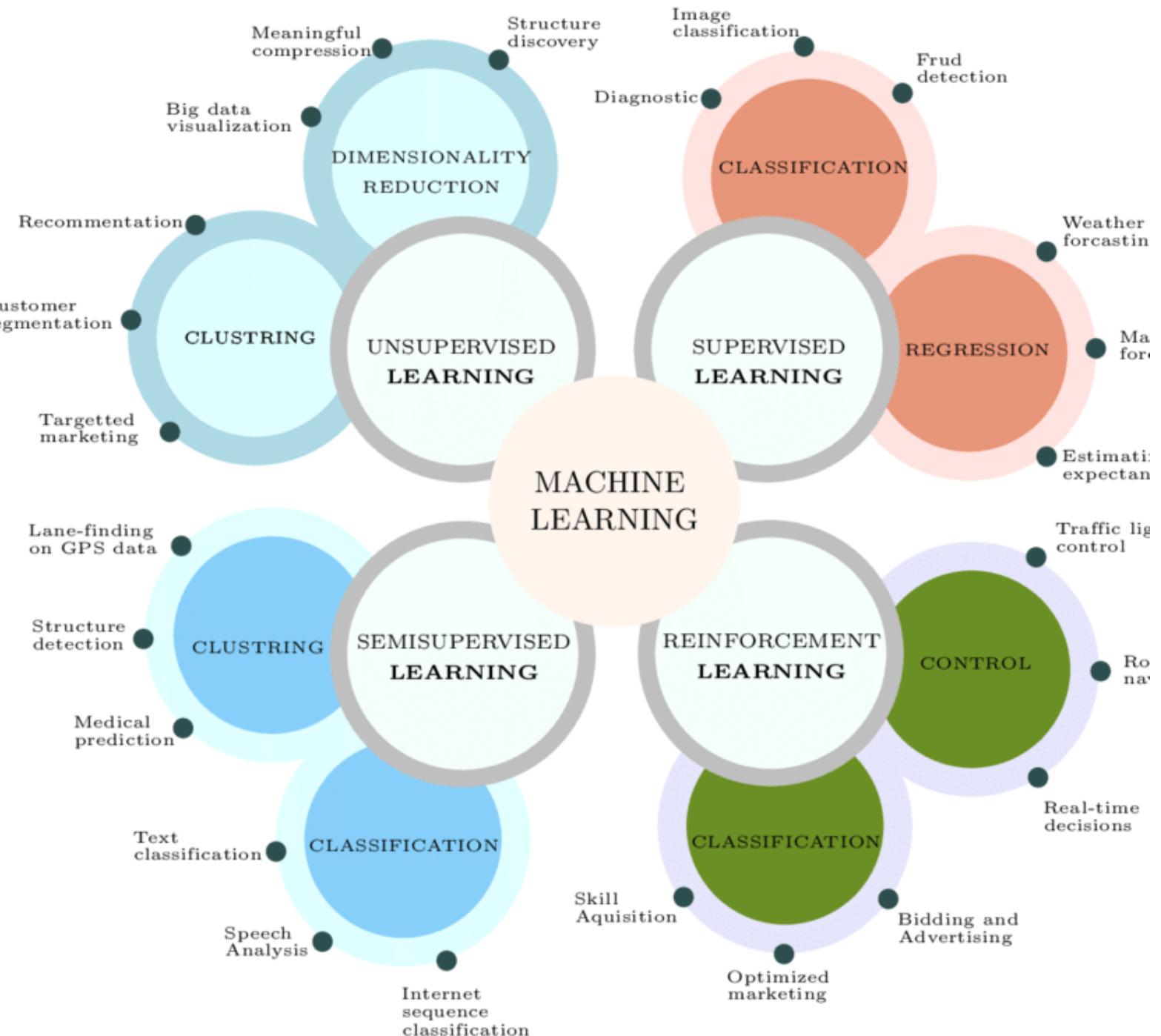
MACHINE LEARNING

Subset of AI technique which use statistical methods to enable machines to improve with experience

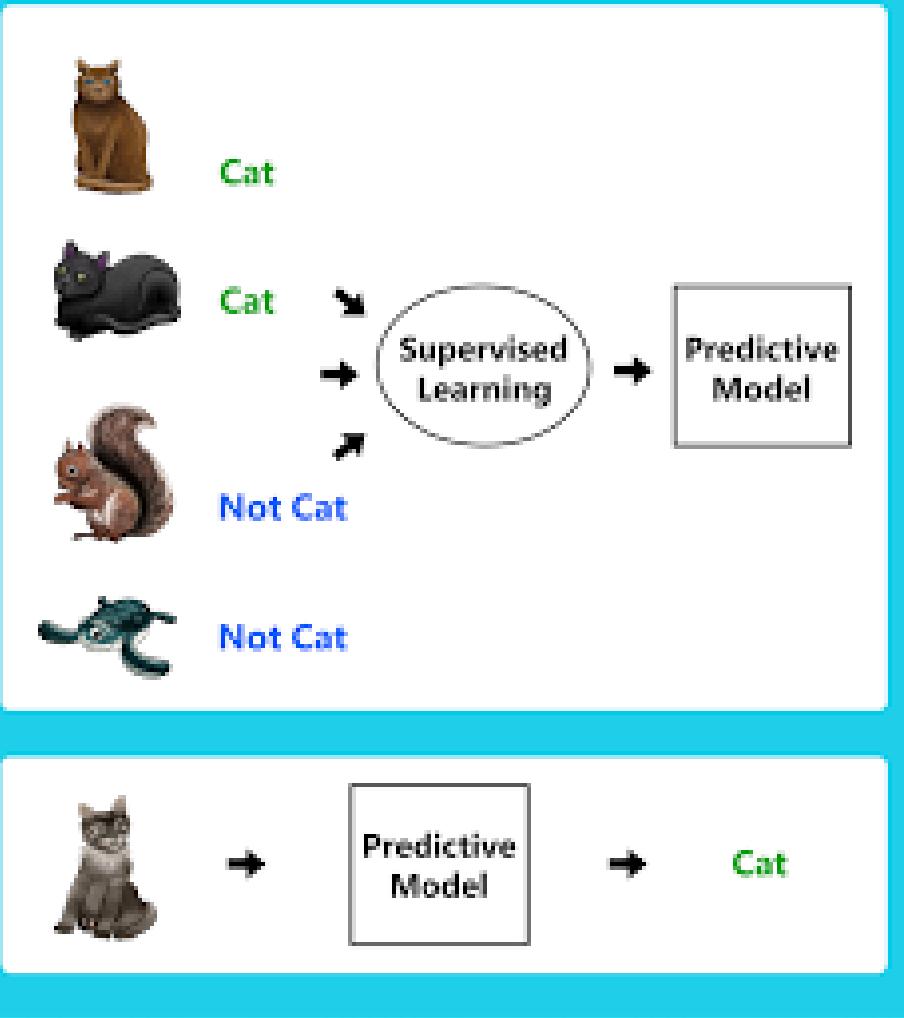
DEEP LEARNING

Subset of ML which make the computation of multi-layer neural network feasible





Supervised Learning (Classification Algorithm)



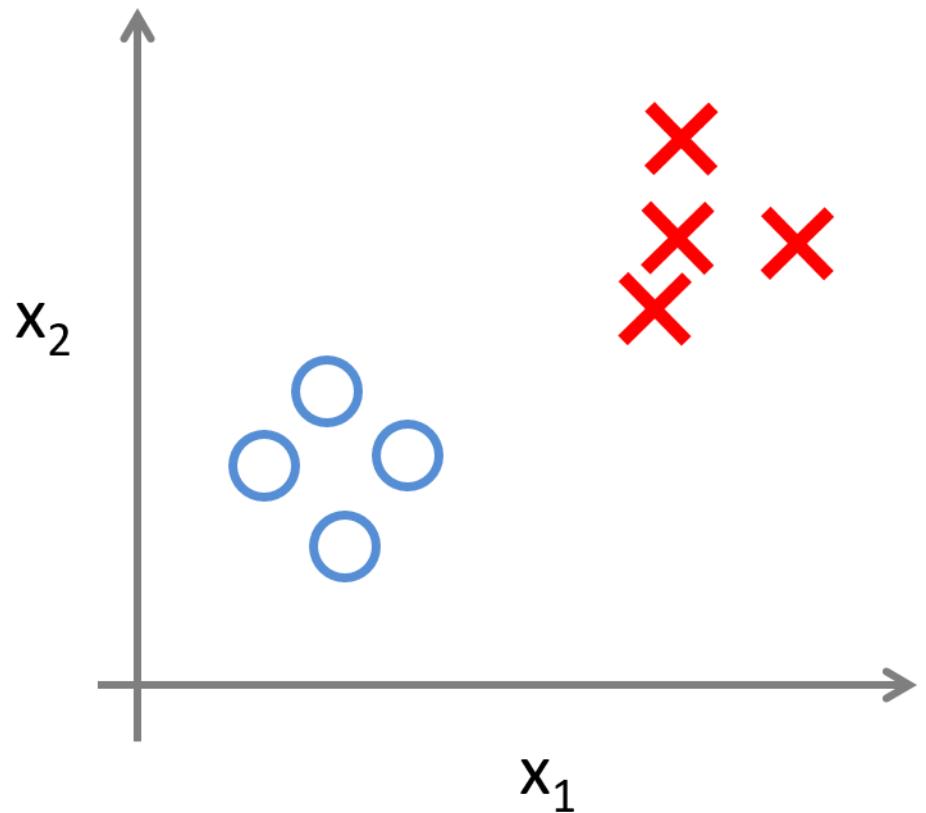
Unsupervised Learning (Clustering Algorithm)



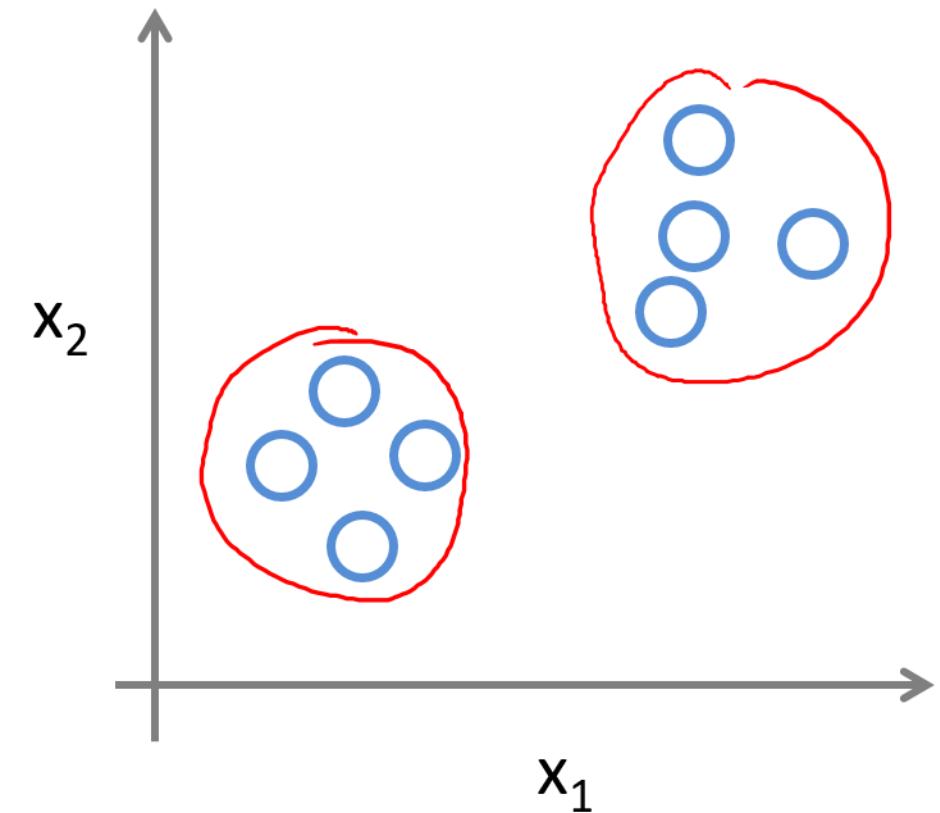
Simple Data set : Excel or CSV File

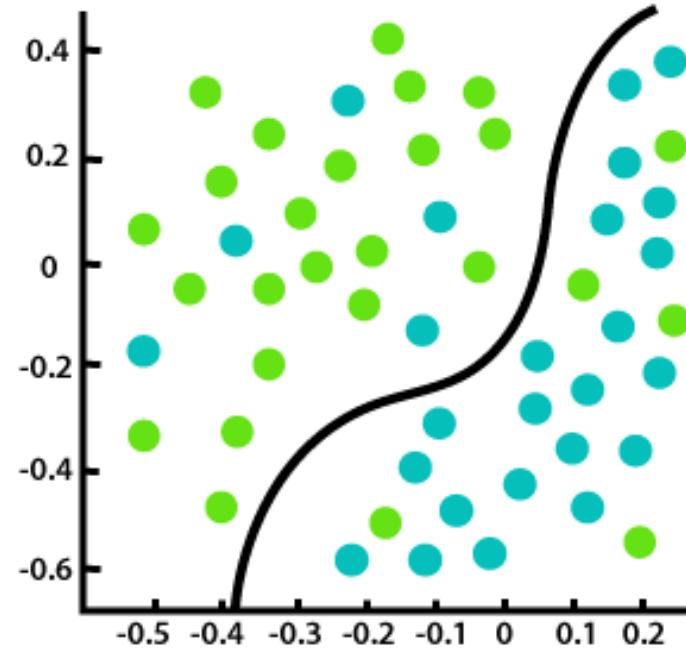
Person	height (feet)	weight (lbs)	foot size (inches)
0	male	6.00	180
1	male	5.92	190
2	male	5.58	170
3	male	5.92	165
4	female	5.00	100

Supervised Learning

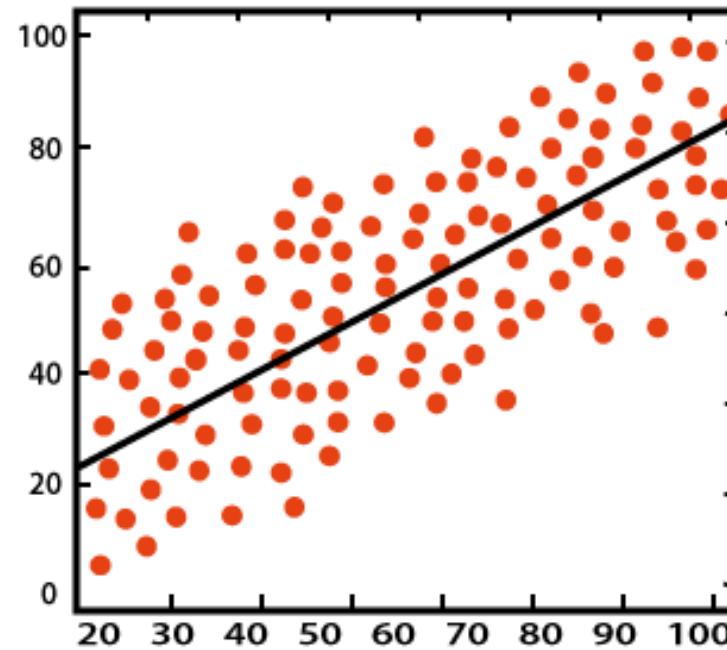


Unsupervised Learning



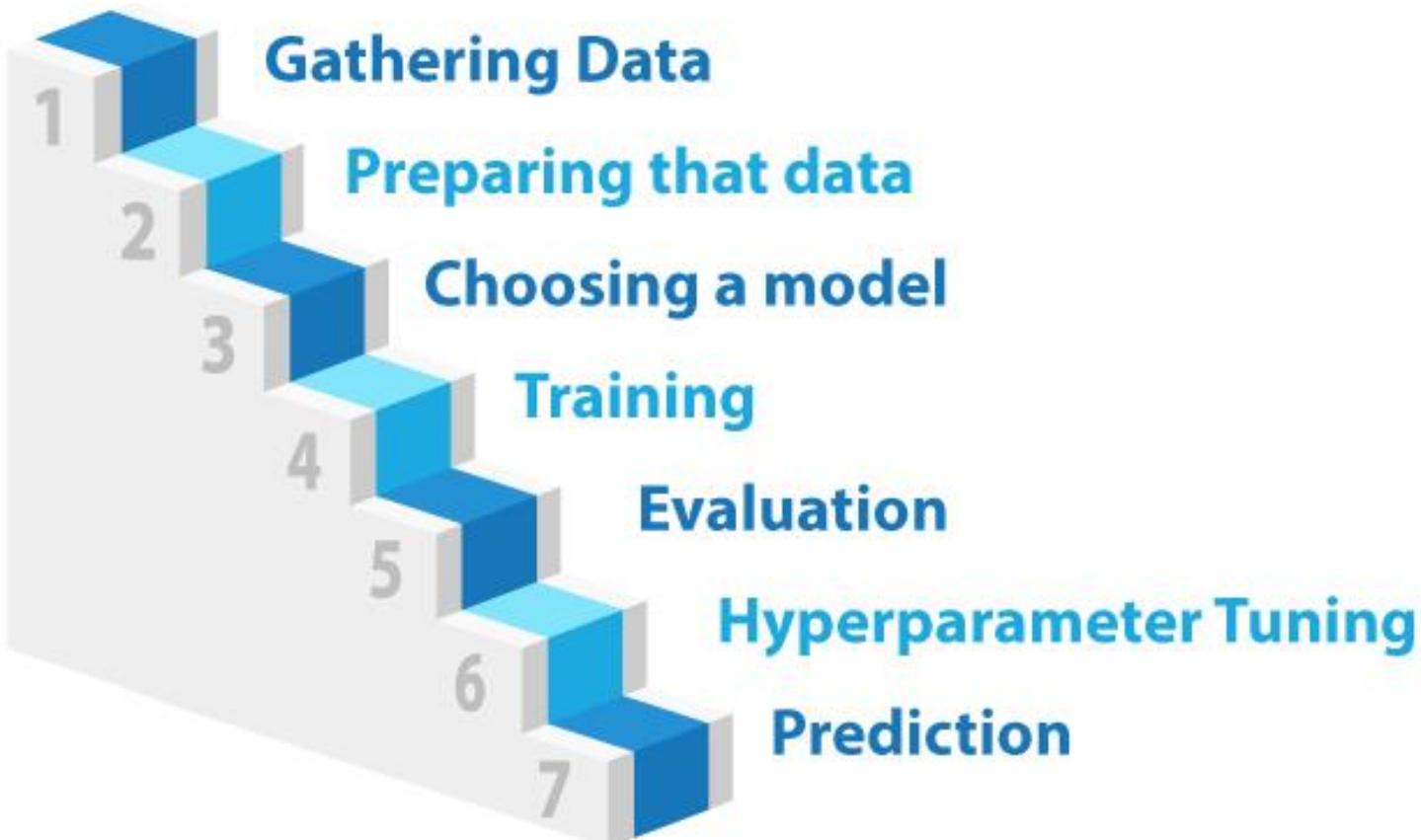


Classification

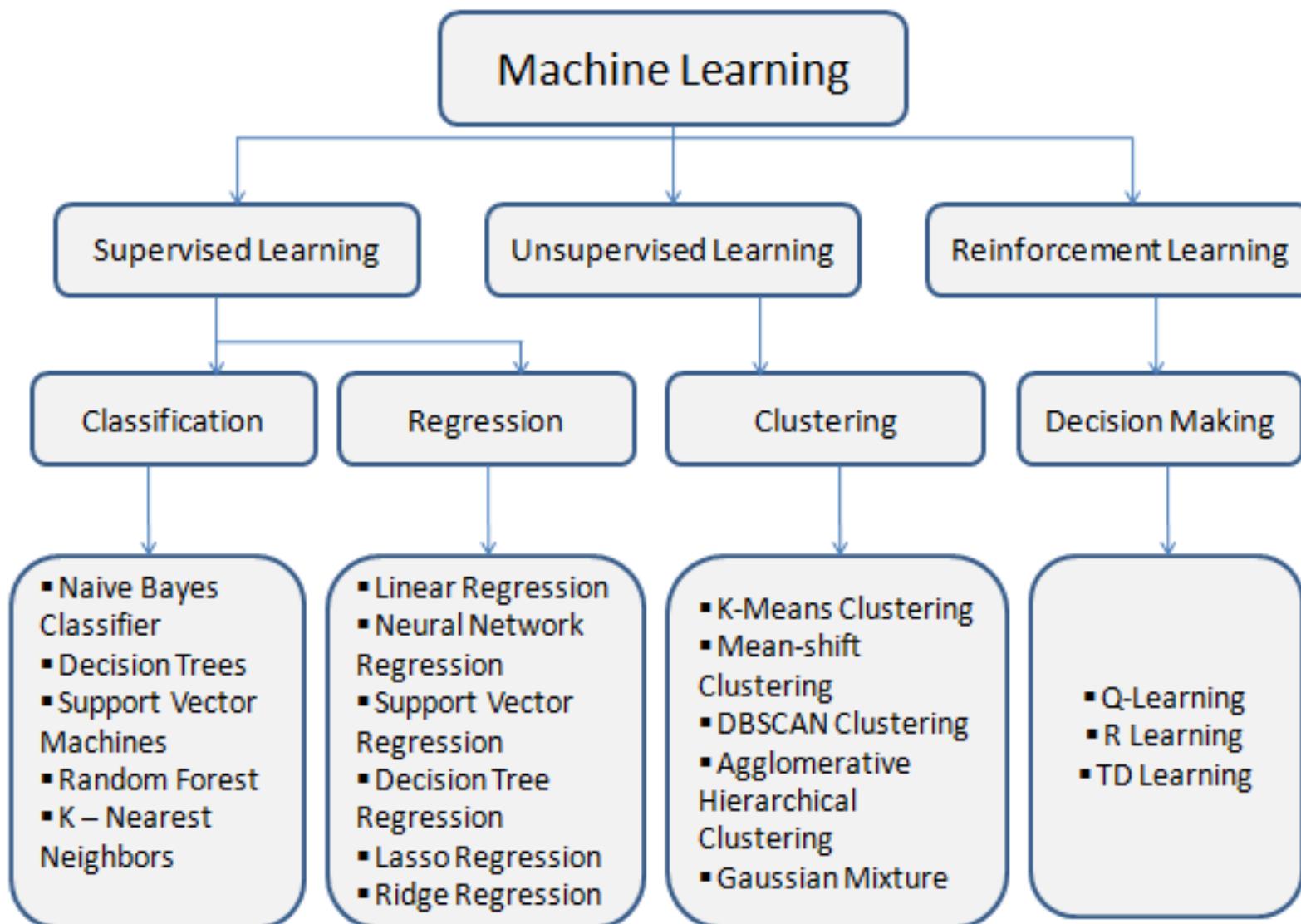


Regression

7 steps of Machine Learning



ML Algorithms and Models



scikit-learn

Machine Learning in Python

[Getting Started](#)[Release Highlights for 1.0](#)[GitHub](#)

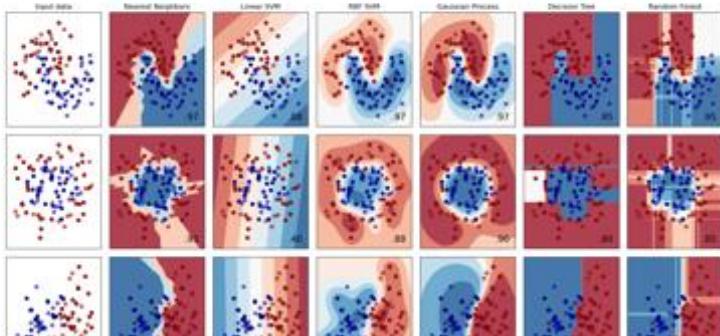
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

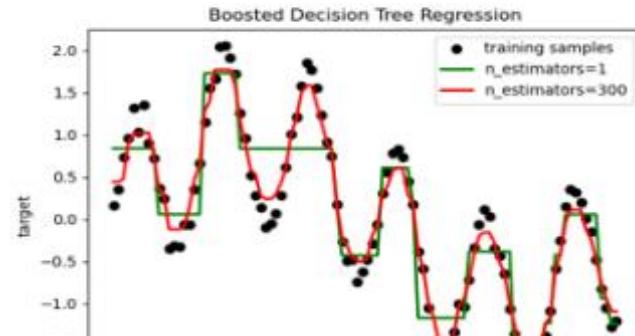


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

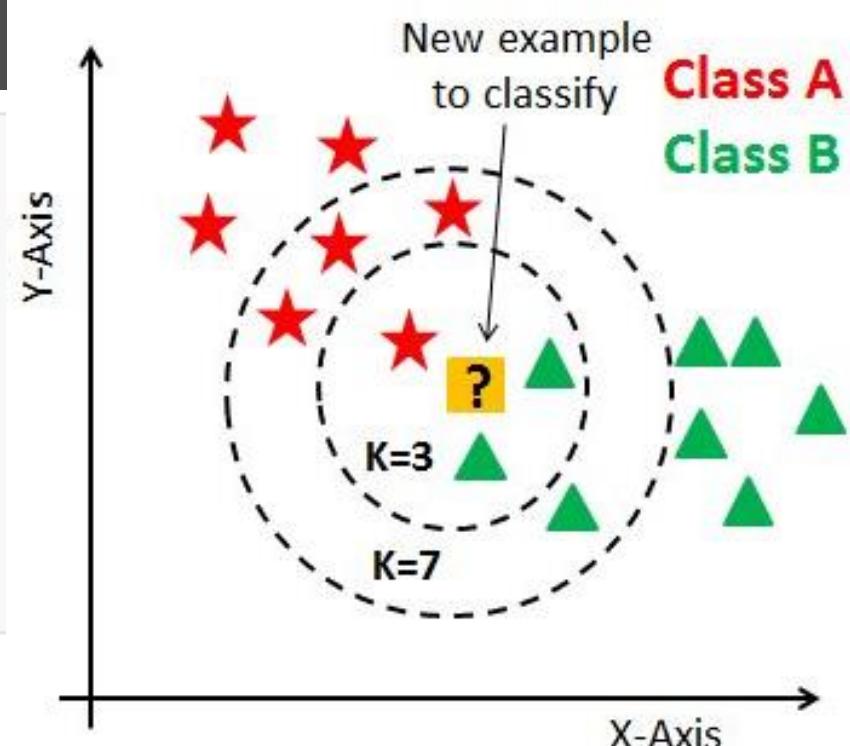
Algorithms: k-Means, spectral clustering, mean-shift, and more...

K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross



ML models implementation scikit-learn for example KNN Classifier

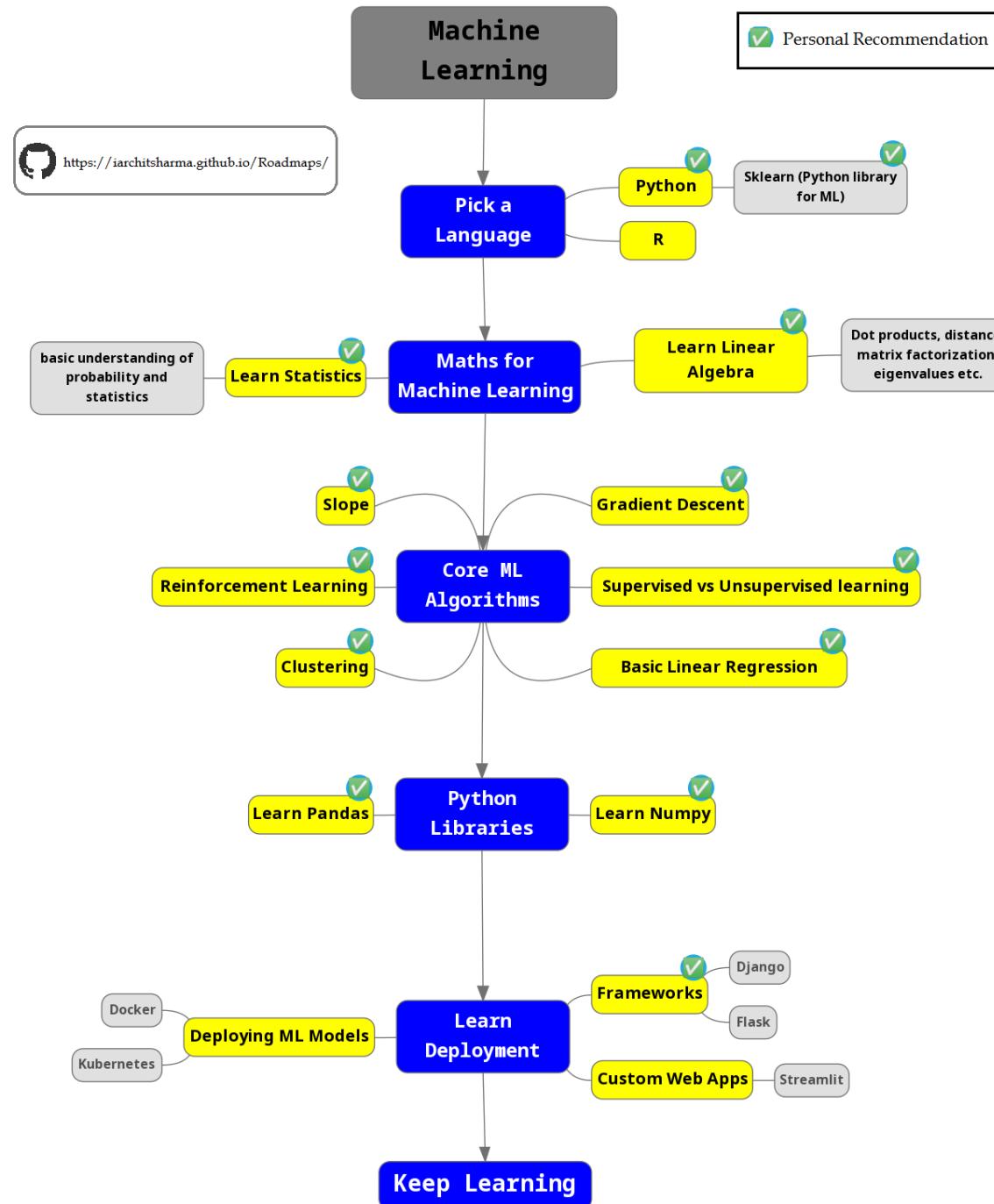
```
>>> X = [[0], [1], [2], [3]]  
>>> y = [0, 0, 1, 1]  
>>> from sklearn.neighbors import KNeighborsClassifier  
>>> neigh = KNeighborsClassifier(n_neighbors=3)  
>>> neigh.fit(X, y)  
KNeighborsClassifier(...)  
>>> print(neigh.predict([[1.1]]))  
[0]  
>>> print(neigh.predict_proba([[0.9]]))  
[[0.666... 0.333...]]
```

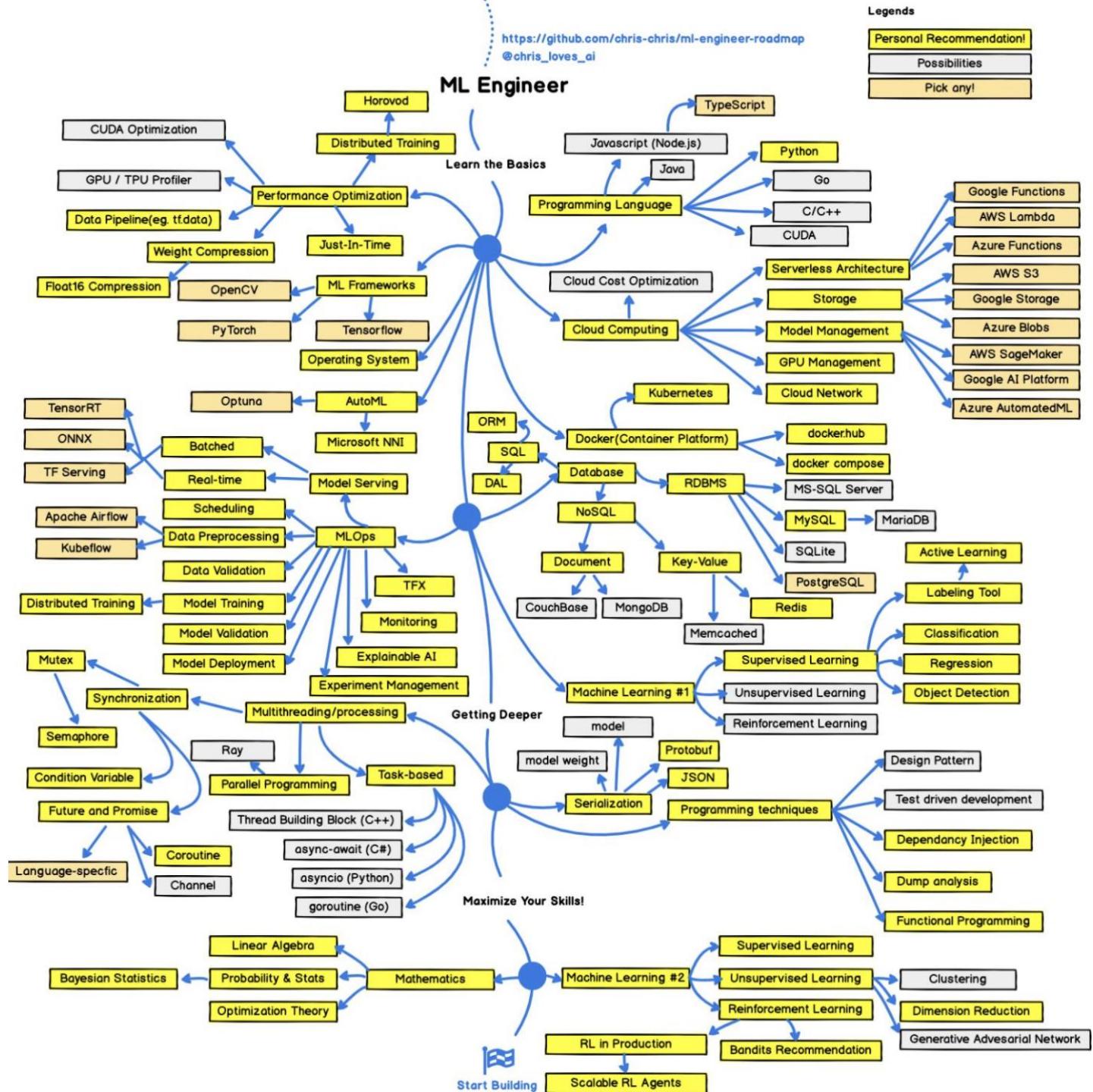


Methods

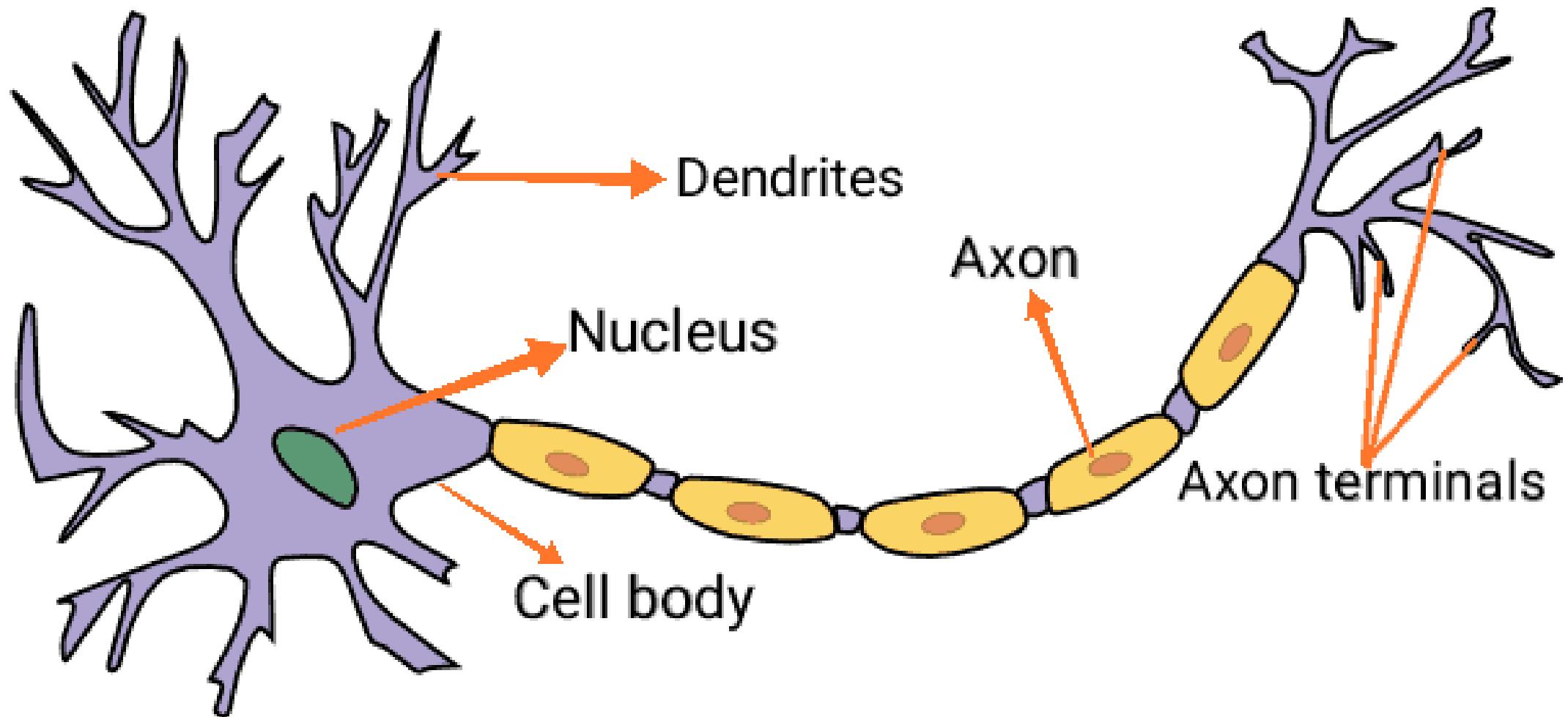
<code>fit(X, y)</code>	Fit the k-nearest neighbors classifier from the training dataset.
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>kneighbors([X, n_neighbors, return_distance])</code>	Find the K-neighbors of a point.
<code>kneighbors_graph([X, n_neighbors, mode])</code>	Compute the (weighted) graph of k-Neighbors for points in X.
<code>predict(X)</code>	Predict the class labels for the provided data.
<code>predict_proba(X)</code>	Return probability estimates for the test data X.
<code>score(X, y[, sample_weight])</code>	Return the mean accuracy on the given test data and labels.
<code>set_params(**params)</code>	Set the parameters of this estimator.



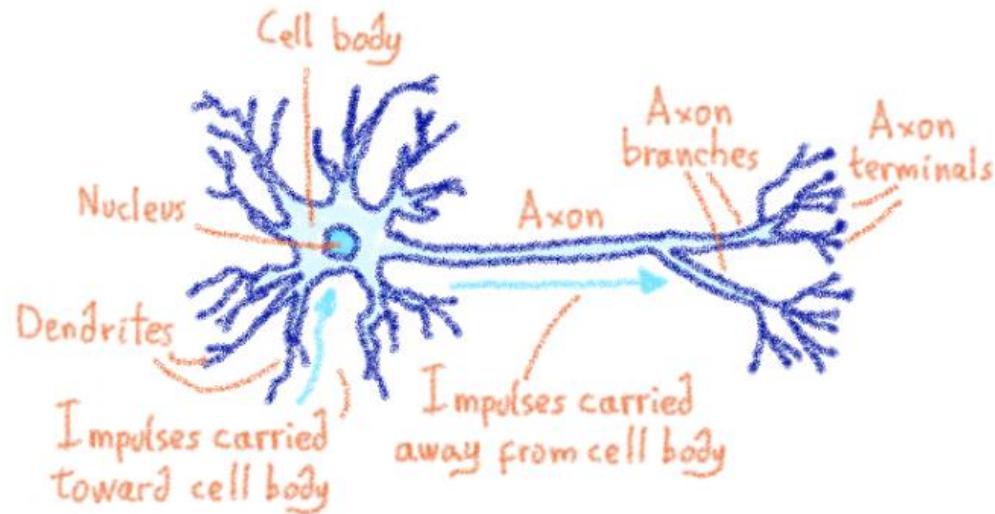




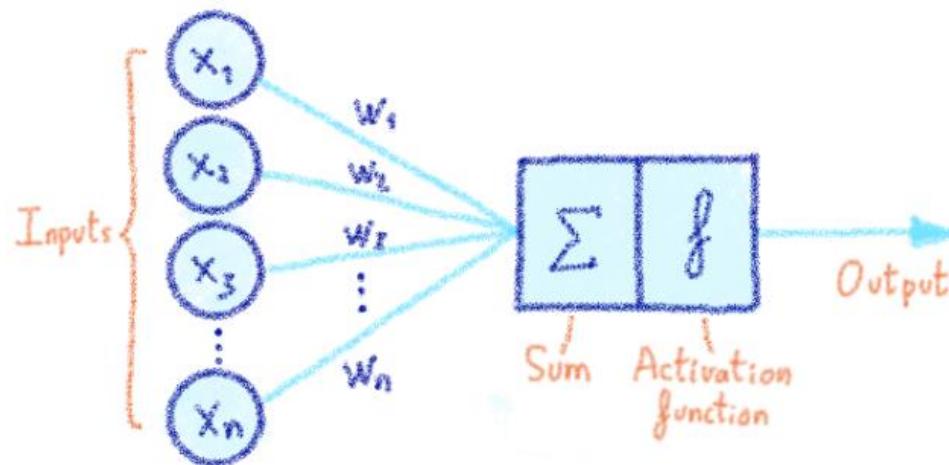
Deep Learning



Biological Neuron



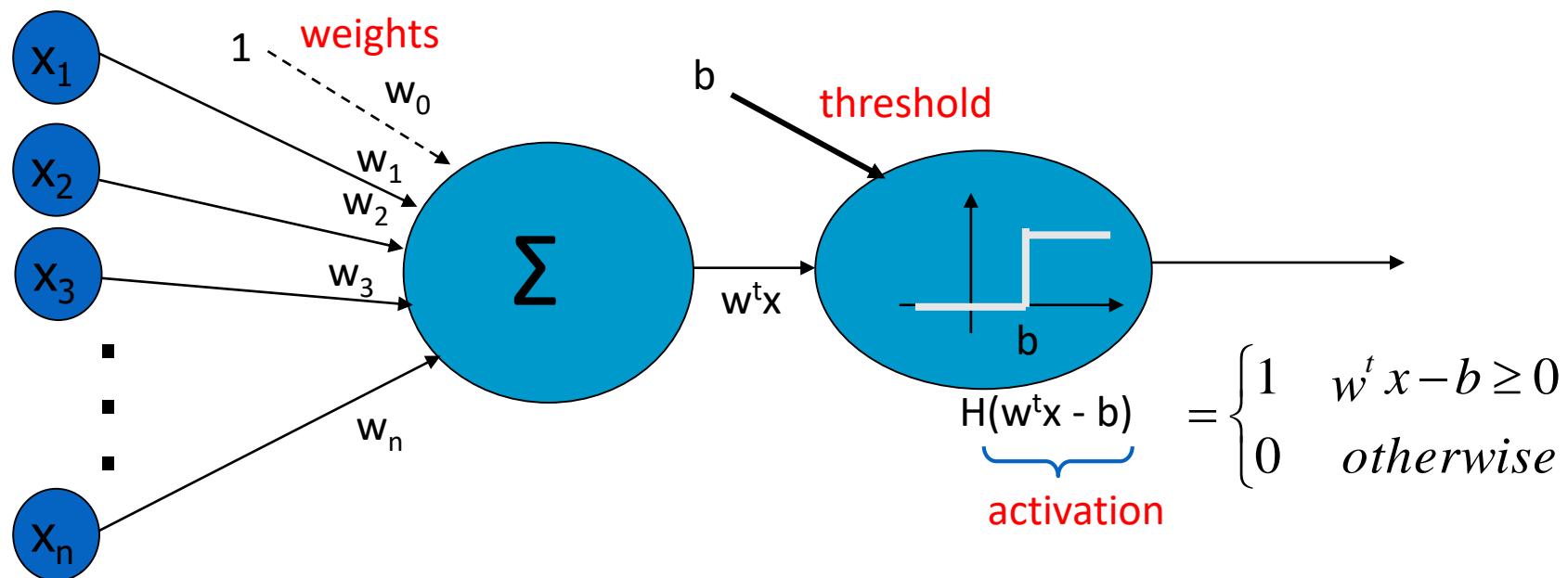
Artificial Neuron



Definition

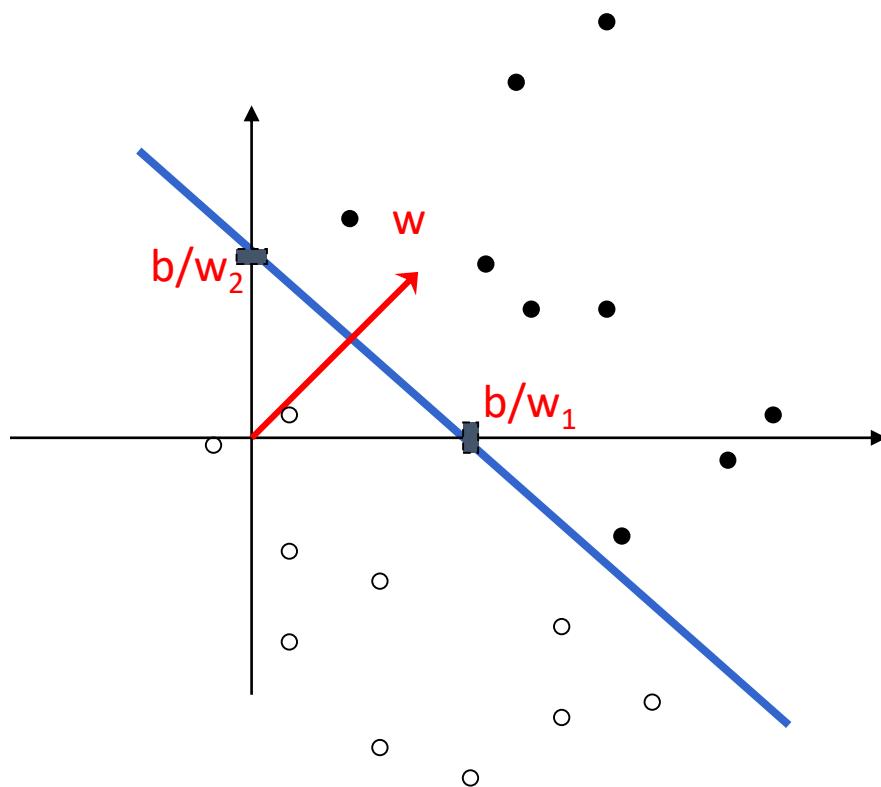
the basic model : *an artificial neuron*

Warren McCulloch & Walter Pitts (1943)



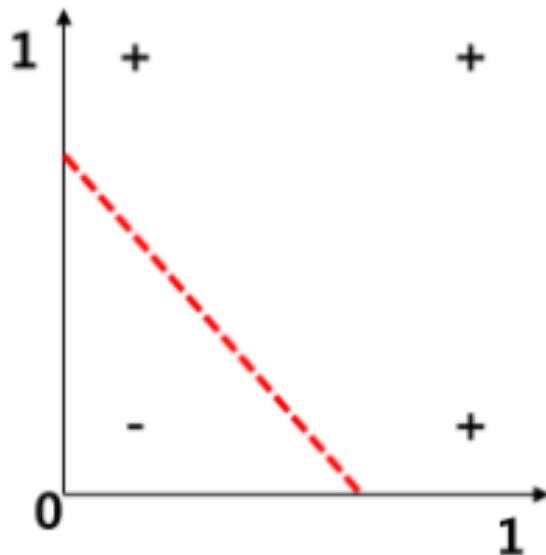
Function

geometry: linear separation boundary

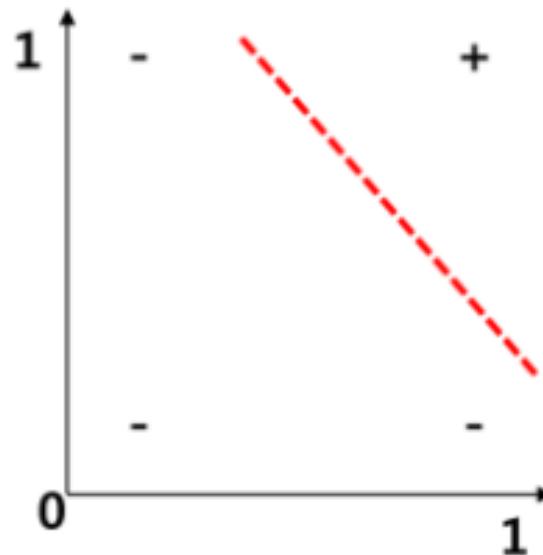


Multi-layer perceptron

OR



AND



XOR



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

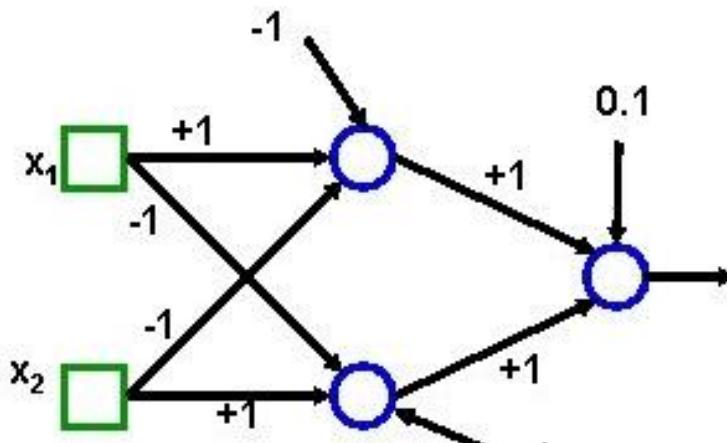
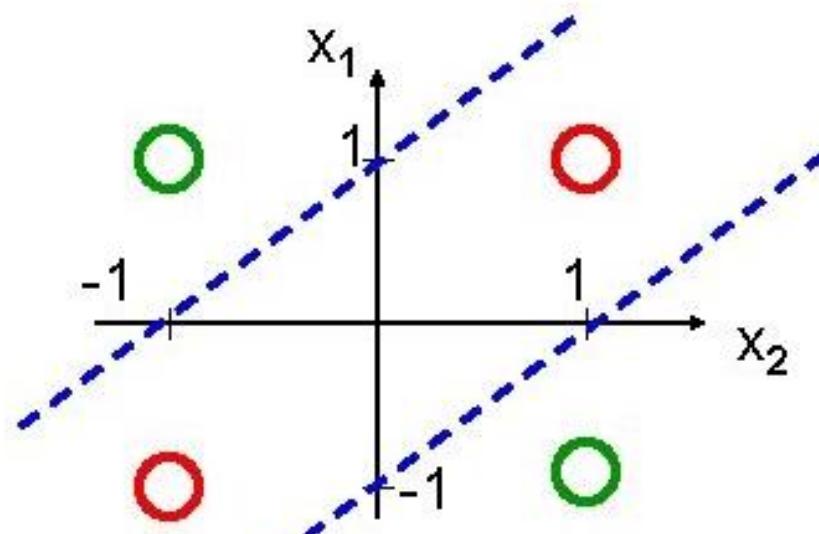
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Multi-layer perceptron

A solution for the XOR problem

x_1	x_2	$x_1 \text{ XOR } x_2$
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1

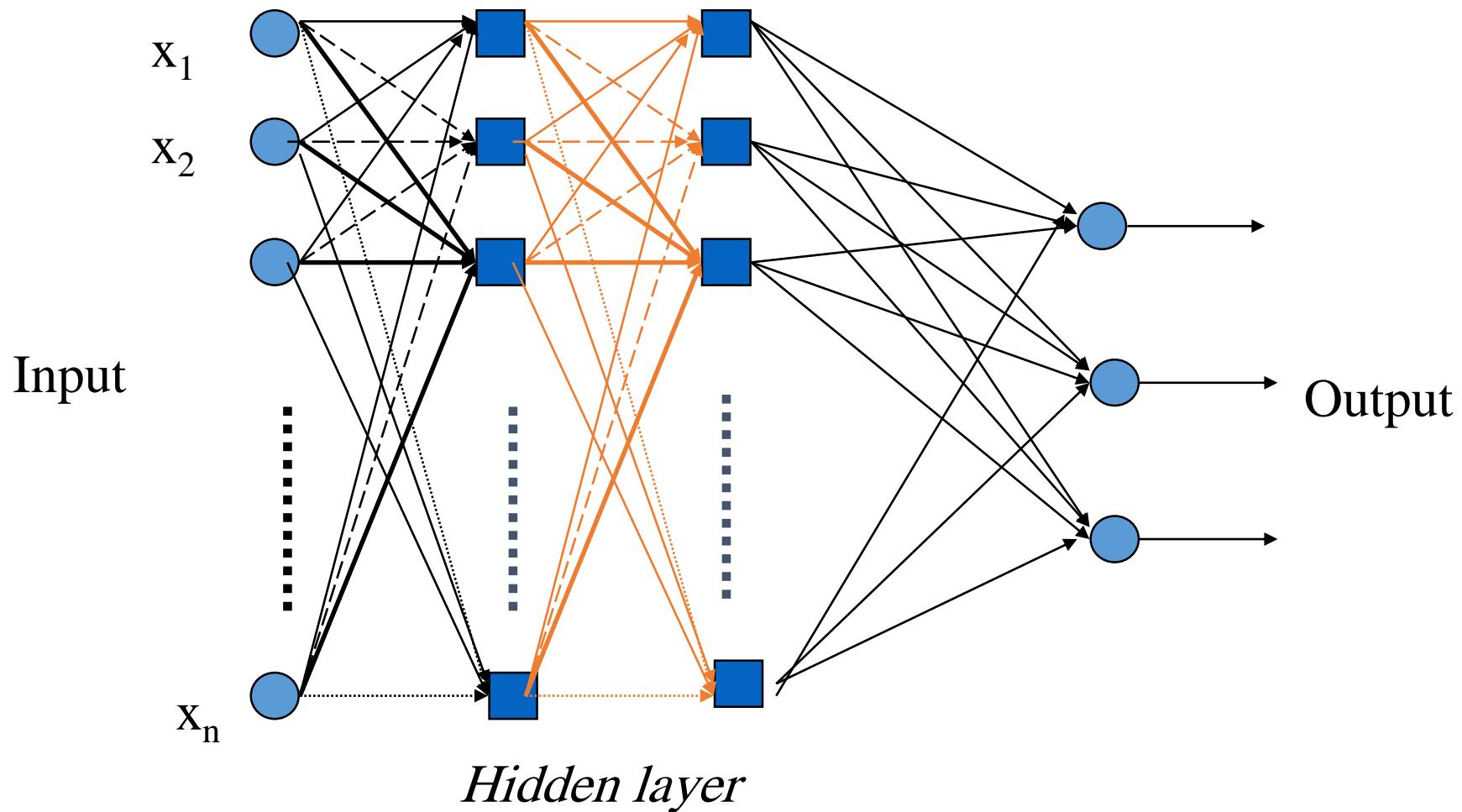


$$\varphi(v) = \begin{cases} 1 & \text{if } v > 0 \\ -1 & \text{if } v \leq 0 \end{cases}$$

φ is the sign function.

Multi-layer perceptron

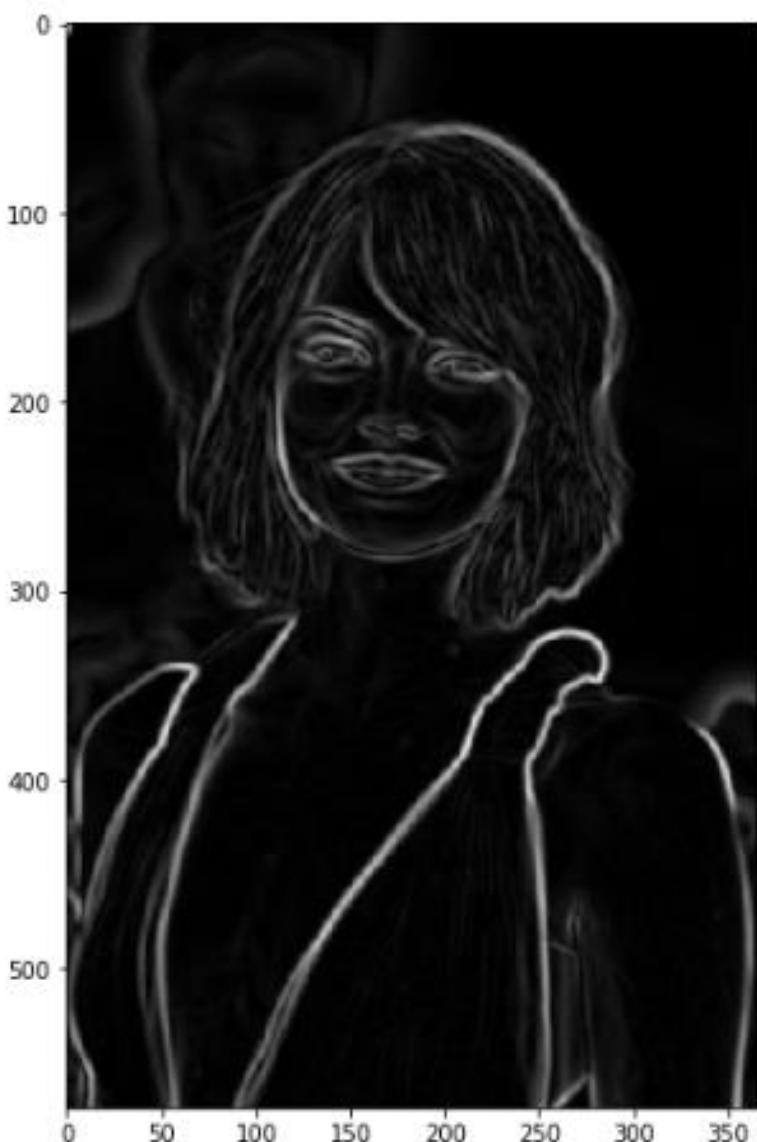
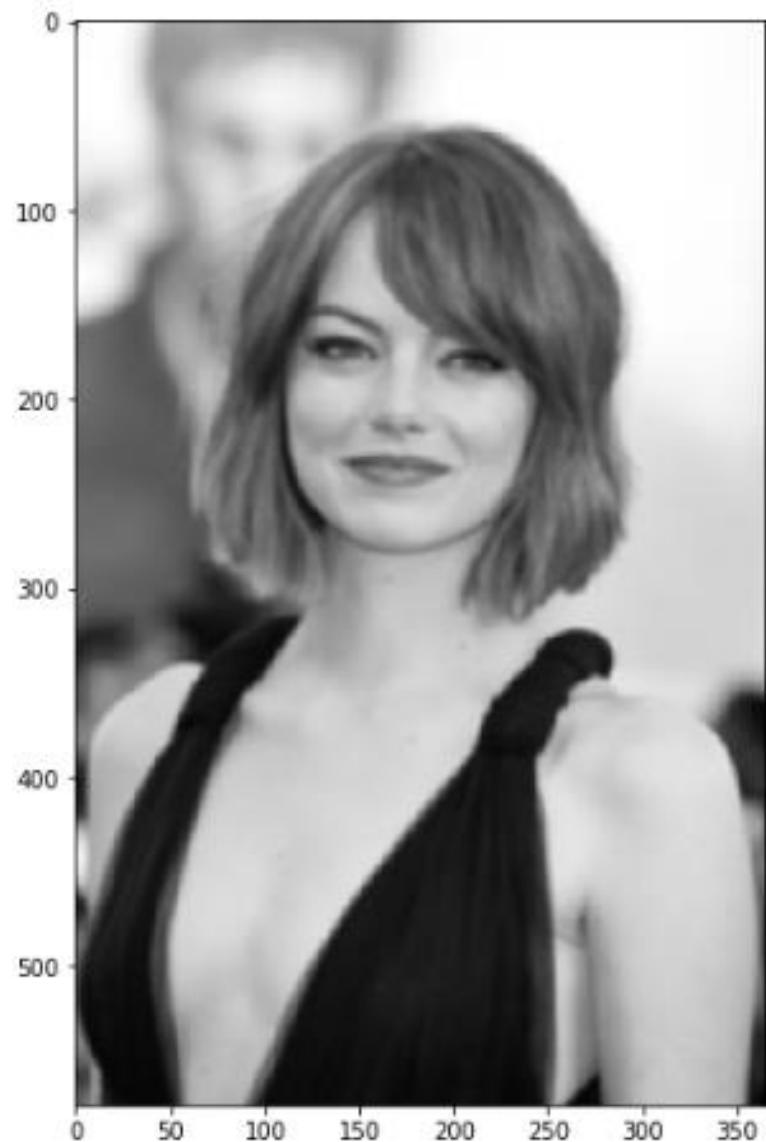
A four-layer network:

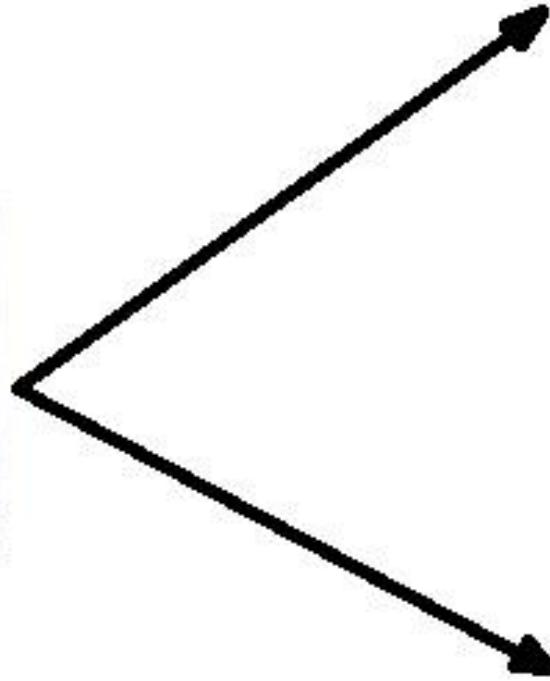


- Some patterns are much smaller than the whole image

Can represent a small region with fewer parameters







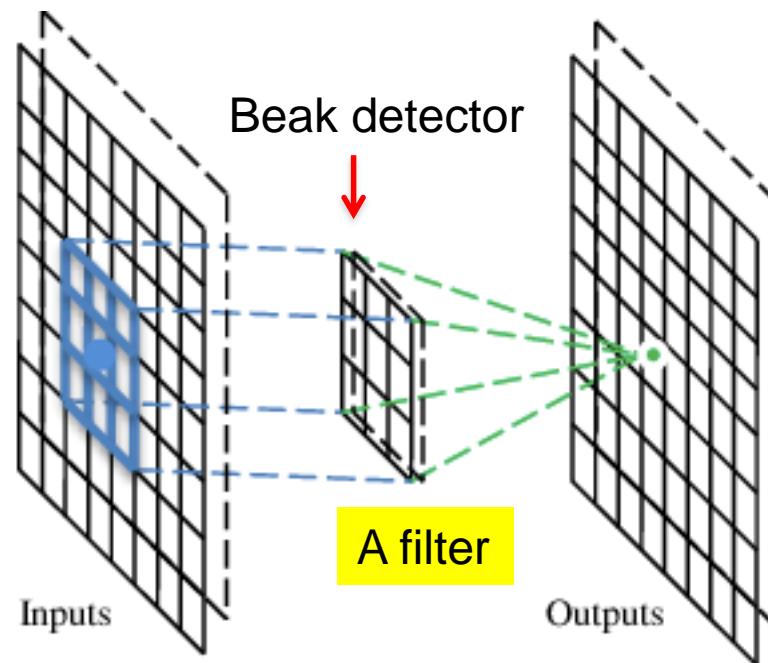
Vertical edges



Horizontal edges

A convolutional layer

A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.



Convolution

These are the network parameters to be learned.

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3).

Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

Average Pooling

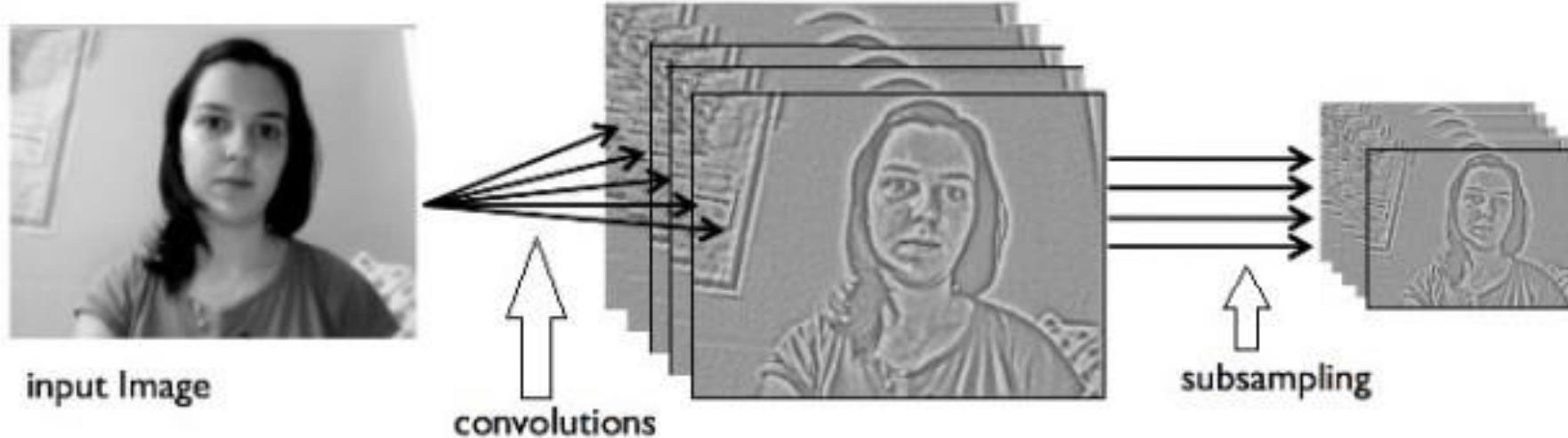
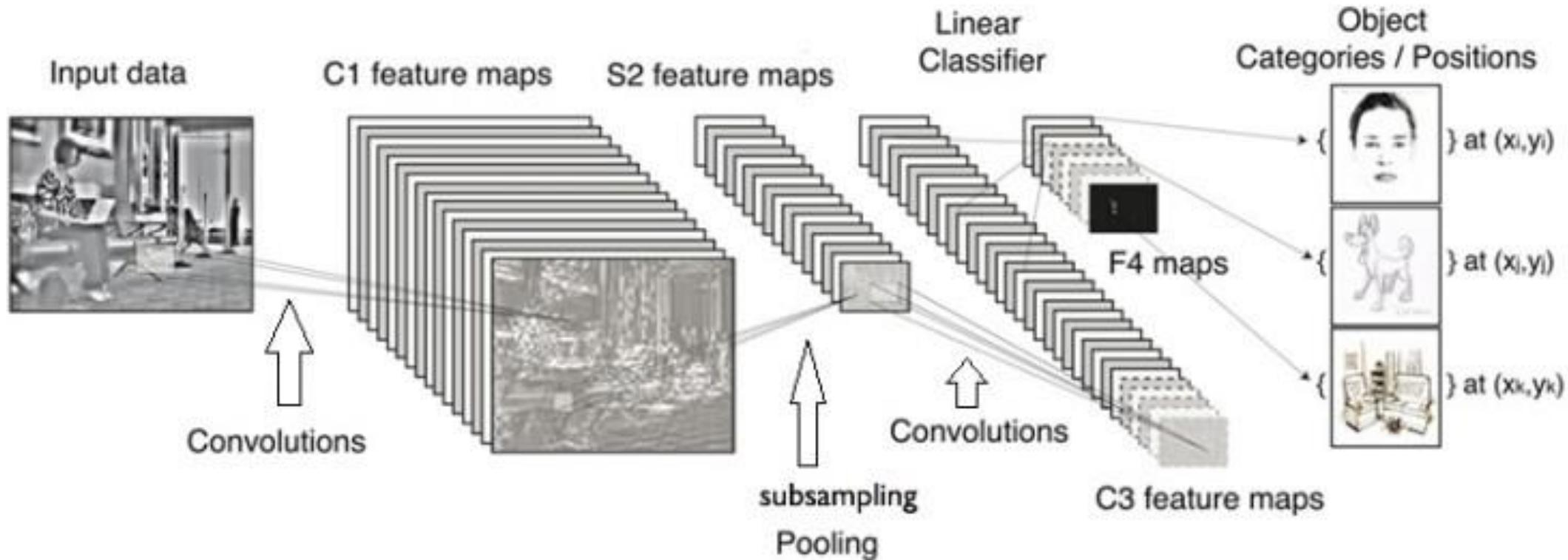
31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

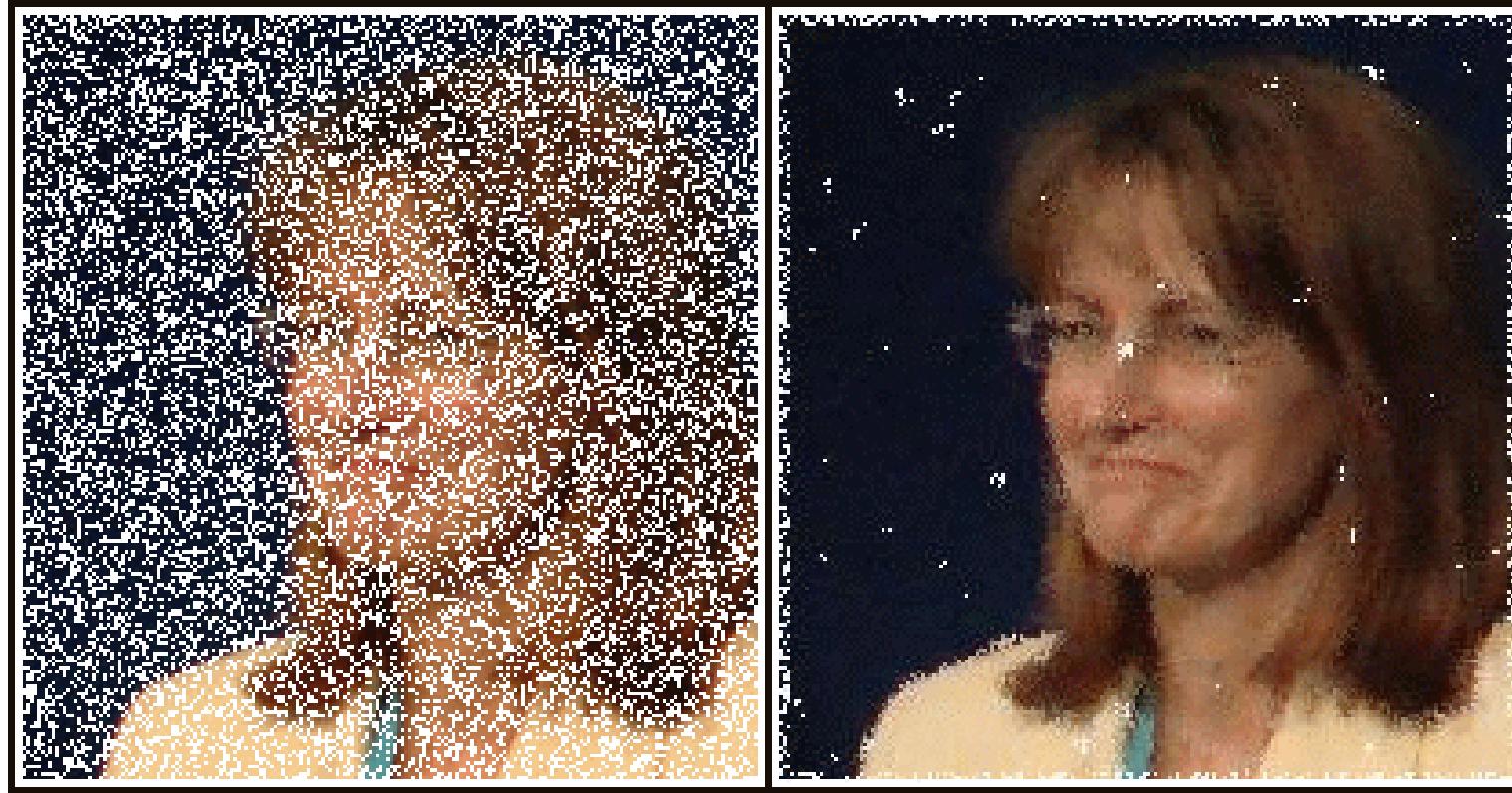
2×2
pool size

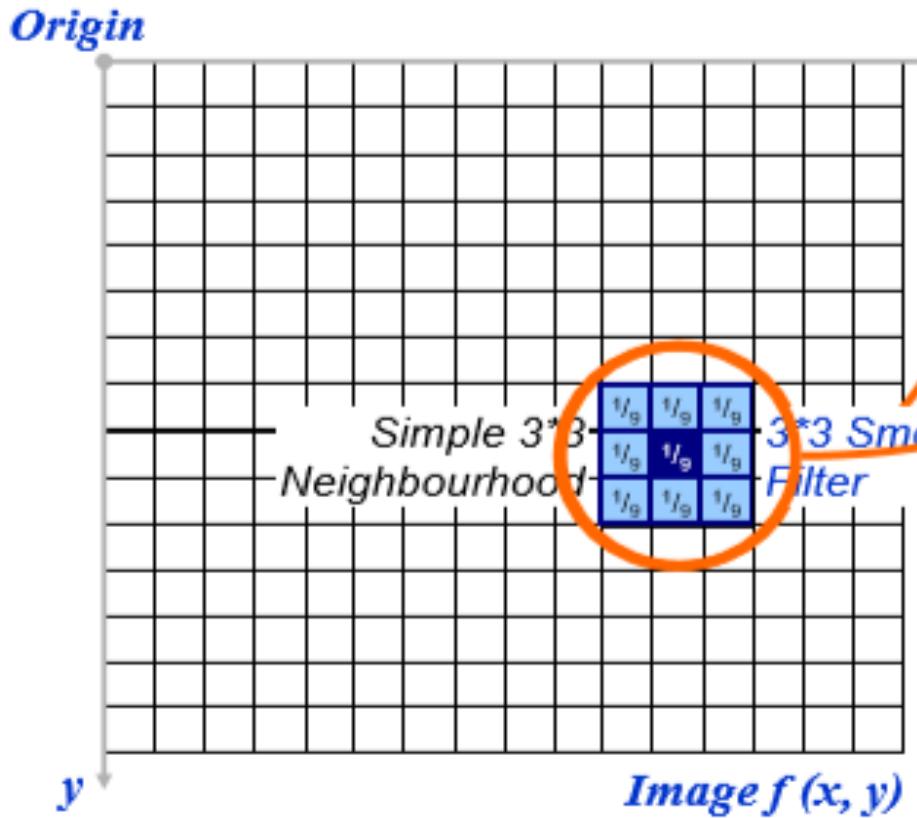
100	184
12	45

2×2
pool size

36	80
12	15







104	100	108
99	106	98
95	90	85

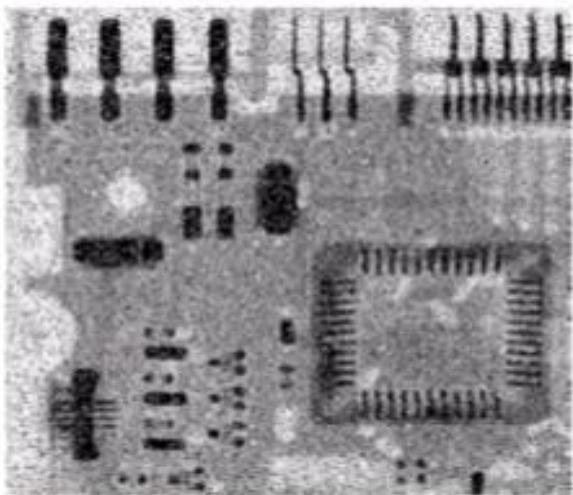
پیکسل های تصویر
اصلی

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

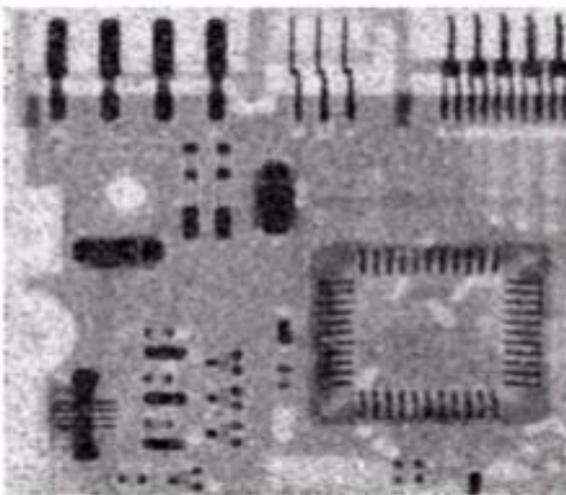
فیلتر

$$\begin{aligned}
 e = & \frac{1}{9} * 106 + \\
 & \frac{1}{9} * 104 + \frac{1}{9} * 100 + \frac{1}{9} * 108 + \\
 & \frac{1}{9} * 99 + \frac{1}{9} * 98 + \\
 & \frac{1}{9} * 95 + \frac{1}{9} * 90 + \frac{1}{9} * 85 \\
 = & 98.3333
 \end{aligned}$$

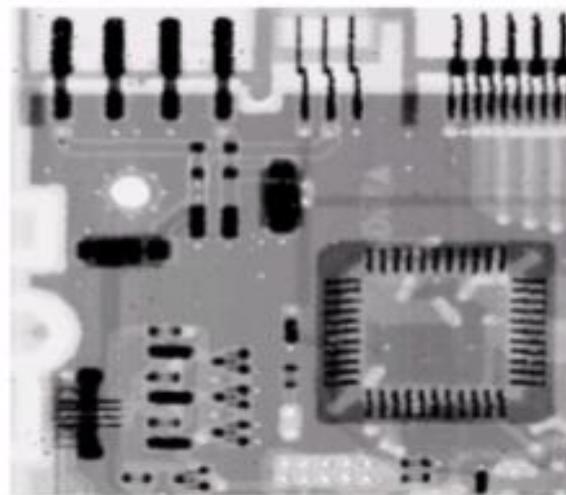
فرایند فوق برای هر پیکسل در تصویر اصلی تولید تصویر هموار شده انجام می شود.



تصویر اصلی با نویز



تصویر بعد از فیلتر میانگین

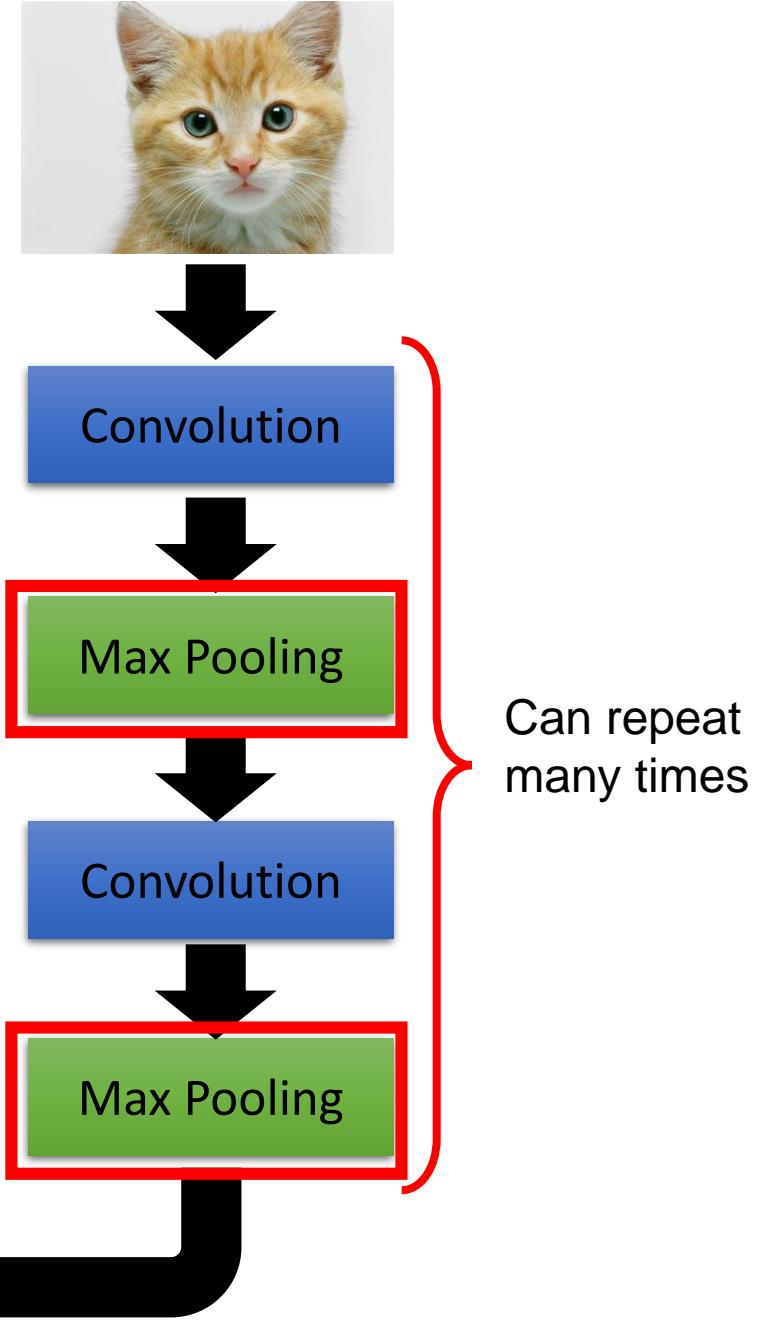
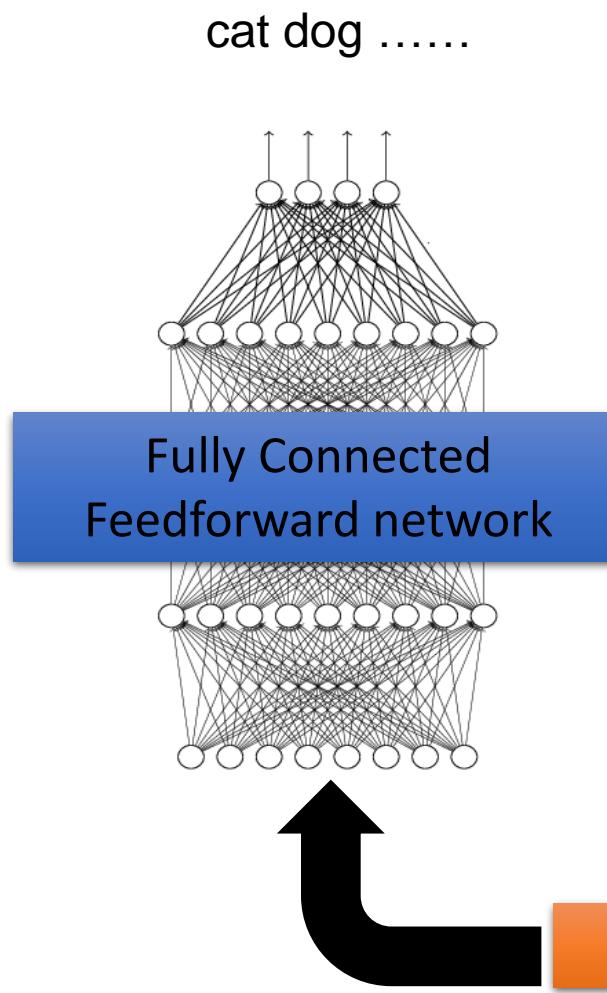


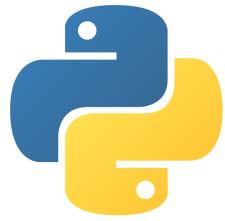
تصویر بعد از فیلتر میانه

برای حذف نویز از فیلتر استفاده می شود.

برخی مواقع فیلتر میانه از فیلتر میانگین کارآیی در حذف نویز دارد.

The whole CNN

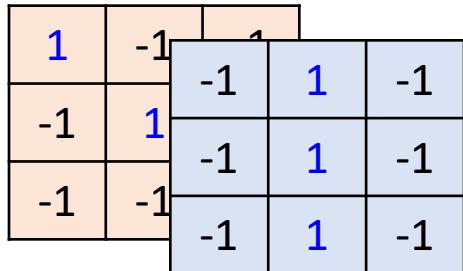




CNN in Keras

Only modified the *network structure* and *input format* (*vector -> 3-D tensor*)

```
model2.add( Convolution2D( 25, 3, 3,  
                           input_shape=(28, 28, 1)) )
```

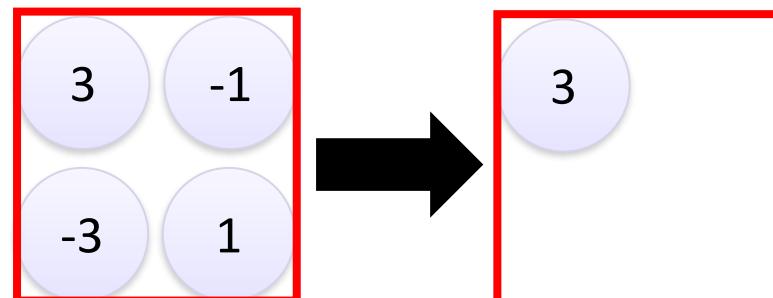


Input_shape = (28 , 28 , 1)

28 x 28 pixels

There are
25 3x3
filters.
1: black/white, 3: RGB

```
model2.add(MaxPooling2D( (2, 2) ))
```



input
↓

Convolution

↓

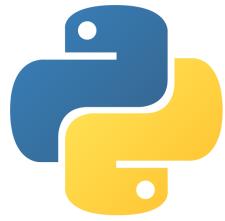
Max Pooling

↓

Convolution

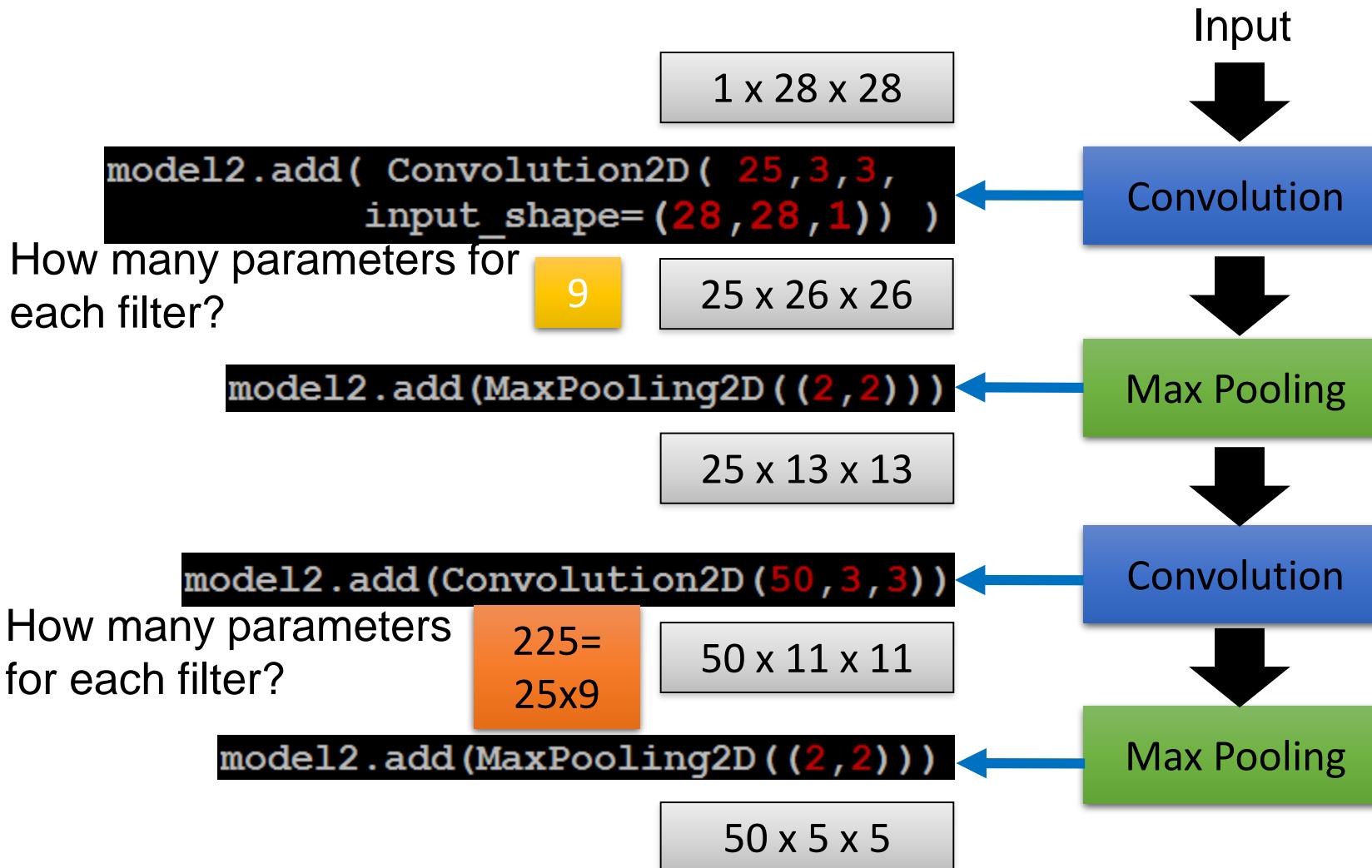
↓

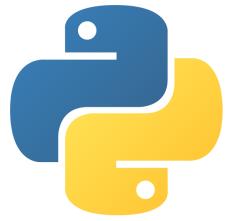
Max Pooling



CNN in Keras

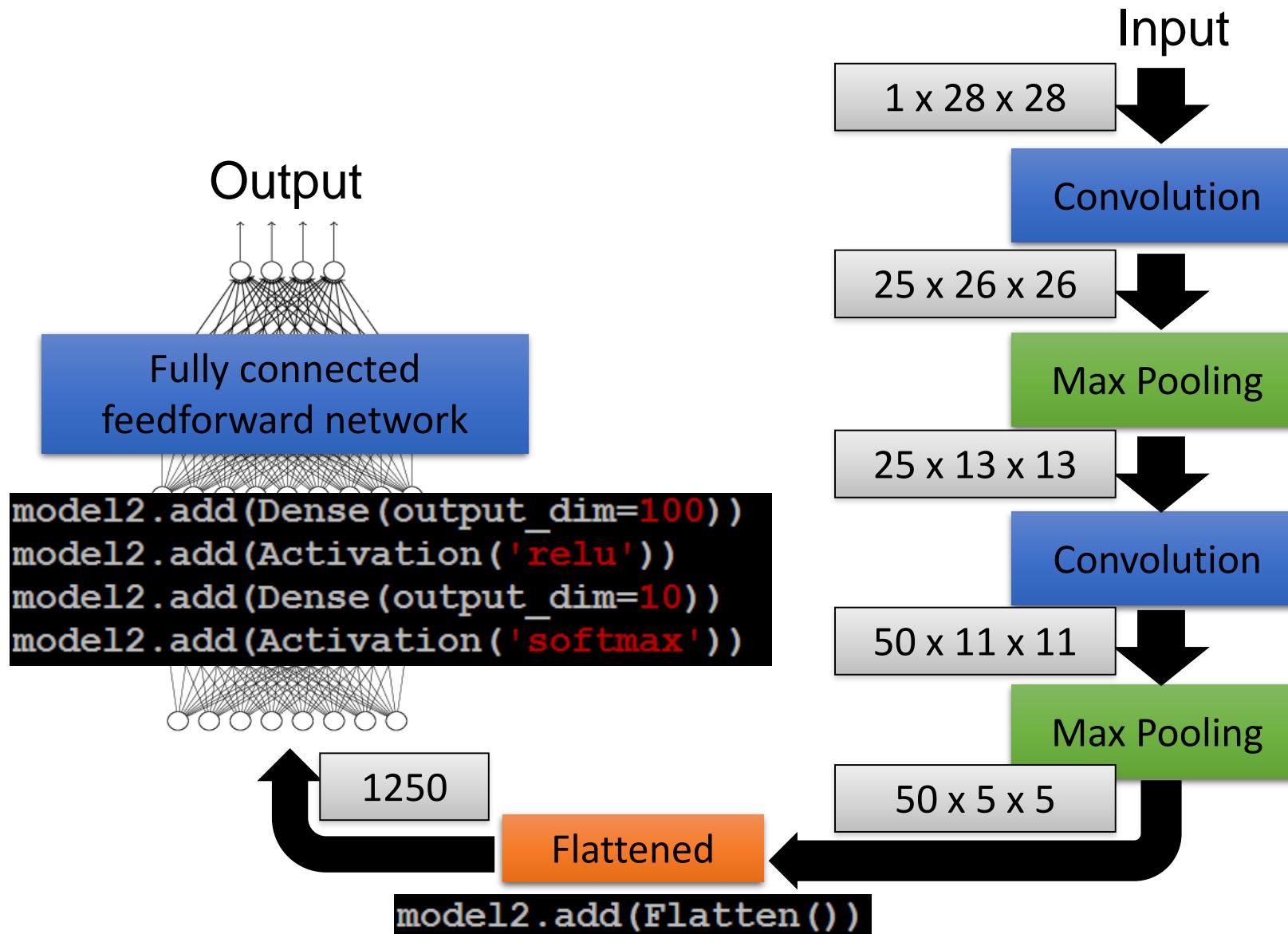
Only modified the *network structure* and *input format* (vector -> 3-D array)





CNN in Keras

Only modified the *network structure* and *input format* (*vector -> 3-D array*)





A. I.

TensorFlow

NumPy

pandas

K Keras

SciPy

scikit
learn

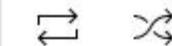
OpenCV

matplotlib

Stanford university machine learning Andrew Ng

A video player interface showing a lecture by Andrew Ng. The video frame shows a man in a light blue shirt speaking. A black overlay at the bottom left of the video frame contains the text "Welcome to CS229 Machine Learning.". The video player has a dark grey header bar with a play button icon, a progress bar showing "0:04 / 1:15:19", and a set of control icons (rewind, forward, volume, etc.). Below the video frame is a dark grey footer bar with the text "Lecture 1 - Stanford CS229: Machine Learning - Andrew Ng (Autumn 2018)" and "1,109,061 views • 17 Apr 2020". To the right of the video frame is a vertical toolbar with icons for "i", "↔", "⤓", and "≡+".

Stanford CS229: Machine Learning | Autumn 2018
stanfordonline - 1/20



▶ Lecture 1 - Stanford CS229:
Machine Learning - Andrew N...
stanfordonline
1:15:20

2 Lecture 2 - Linear Regression
and Gradient Descent | ...
stanfordonline
1:18:17

3 Lecture 3 - Locally Weighted &
Logistic Regression | Stanfor...
stanfordonline
1:19:34

4 Lecture 4 - Perceptron &
Generalized Linear Model | ...
stanfordonline
1:22:02

5 Lecture 5 - GDA & Naive Bayes
| Stanford CS229: Machine...
stanfordonline
1:18:52

Lecture 6 - Support Vector
Machines | Stanford CS229...

All

Machine learning

Media theories

PI

Machine Learning Andrew Ng



Even among machine learning practitioners there isn't a well accepted definition of Machine Learning what is and what isn't machine learning?

◀ ▶ ⏪ ⏩ 🔊 0:08 / 7:14



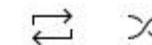
Lecture 1.1 – What Is Machine Learning – [Machine Learning | Andrew Ng]

1,435,991 views • 7 Oct 2016

Like 9.4K Dislike Share Save ...

Machine Learning – Andrew Ng, Stanford University

Artificial Intelligence - All in One - 1/112



Lecture 1.1 – What Is Machine Learning – [Machine Learnin...

Artificial Intelligence - All in One



Lecture 1.2 – Supervised Learning – [Machine Learnin...

Artificial Intelligence - All in One



Lecture 1.3 – Unsupervised Learning – [Machine Learnin...

Artificial Intelligence - All in One



Lecture 2.1 – Linear Regression With One Variable...

Artificial Intelligence - All in One

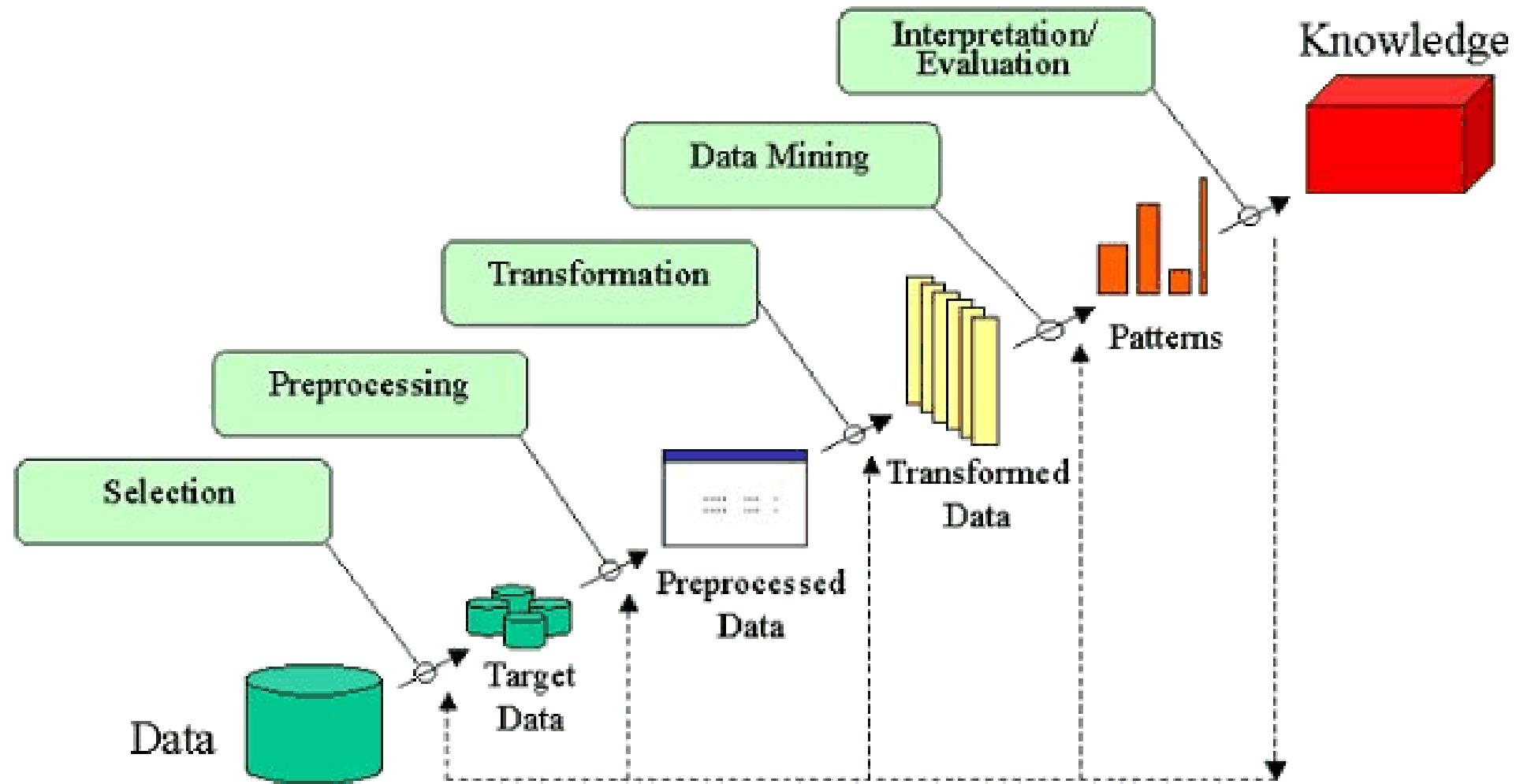


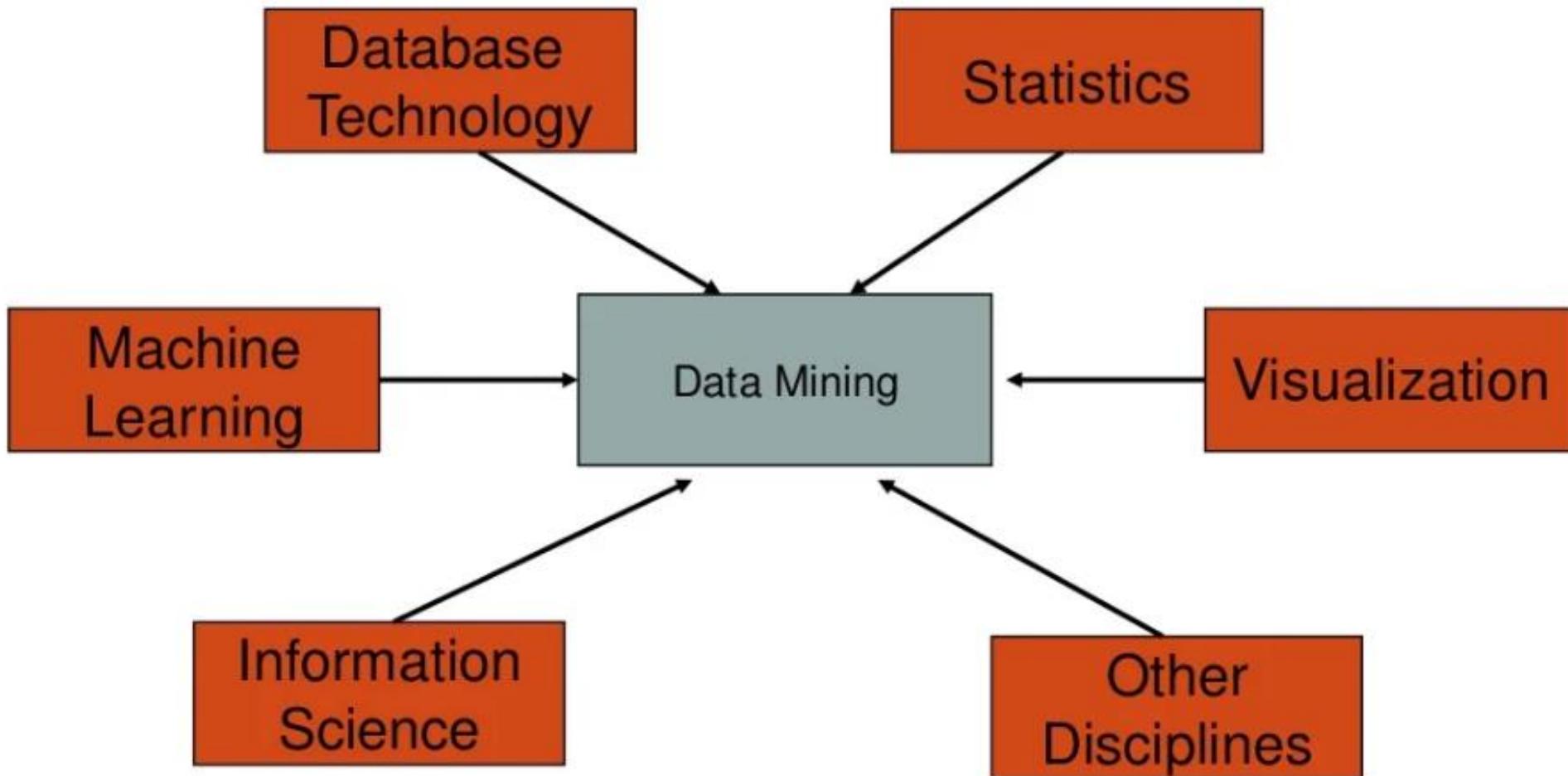
Lecture 2.2 – Linear Regression With One Variable...

Artificial Intelligence - All in One

Lecture 2.3 – Linear

Data Mining KDD : knowledge discovery and data mining







Data Mining Algorithms

Regression	Classification	Clustering	Outlier Detection	Reinforcement Learning
Linear Regression	Naive Bayes Classification	KMeans Clustering	Statistical Method	Value-Based
Polynomial Regression	KNearest Neigbor Classification	Hierarchinal Clustering	Proximity-Based Method	Policy-Based
Lasso Regression	Random Forest Classification	Density-Based Clustering	cluster-Based Method	Model-Based
Logisctic Regression	Support Vector Machine	Grid-Based Clustering		
	Neural Networks			

Data Mining Techniques

Anomaly Detection

Association Learning

Clustering Analysis

Classification Analysis

Regression Analysis

Choice Modeling

Rule Induction

Neural Networks

Data Mining Algorithms

1. C4.5 classifier Algorithm
2. K-means
3. Support vector machines (SVM)
4. Apriori
5. Expectation-Maximization (EM)
6. PageRank
7. AdaBoost
8. KNN
9. Navie Bayes
10. CART

Data Mining Models : for example Association rule

- Ideas come from the market basket analysis (MBA)
 - Let's go shopping!

Milk, eggs, sugar,
bread



Milk, eggs, cereal,
bread

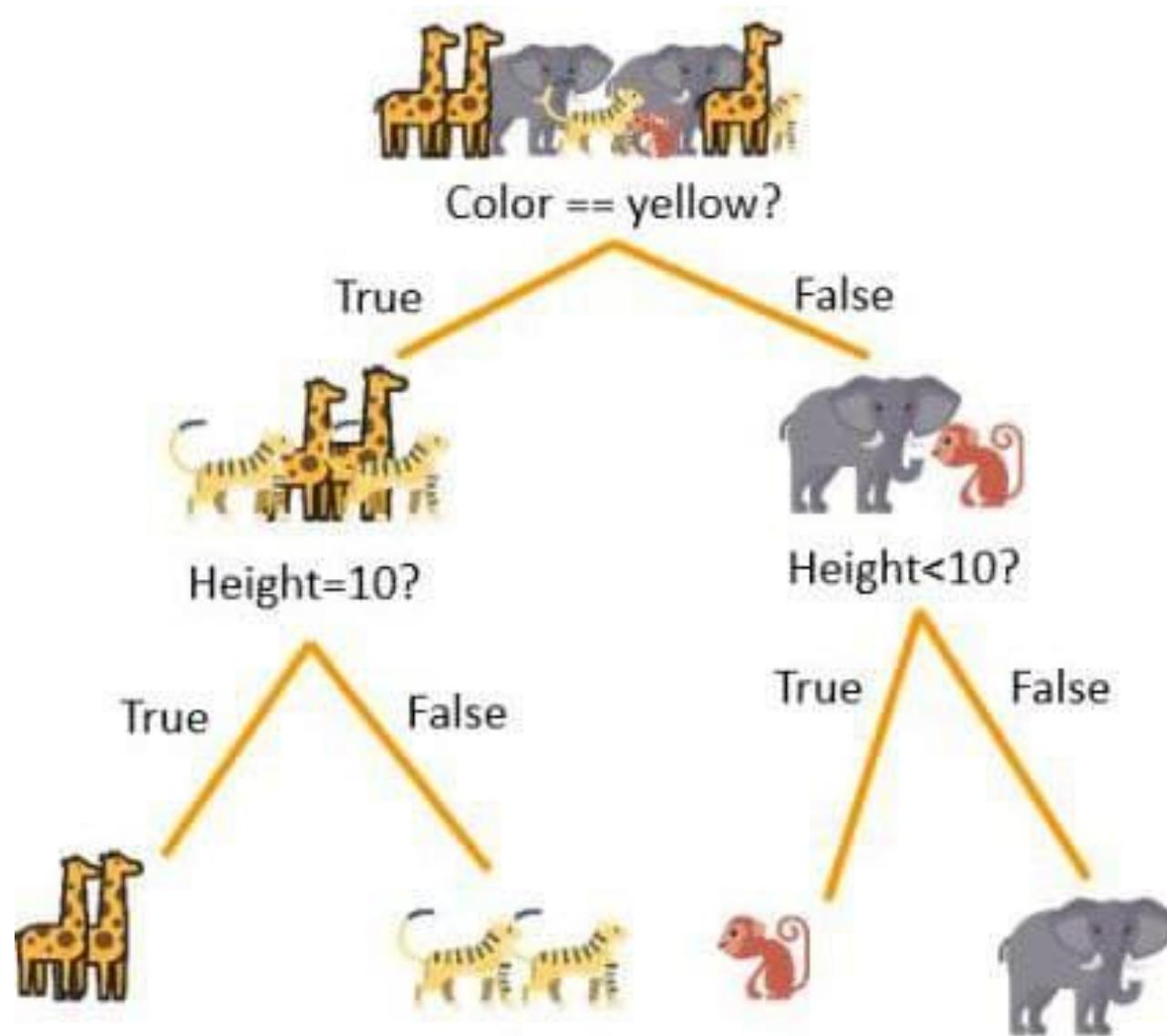


Eggs, sugar

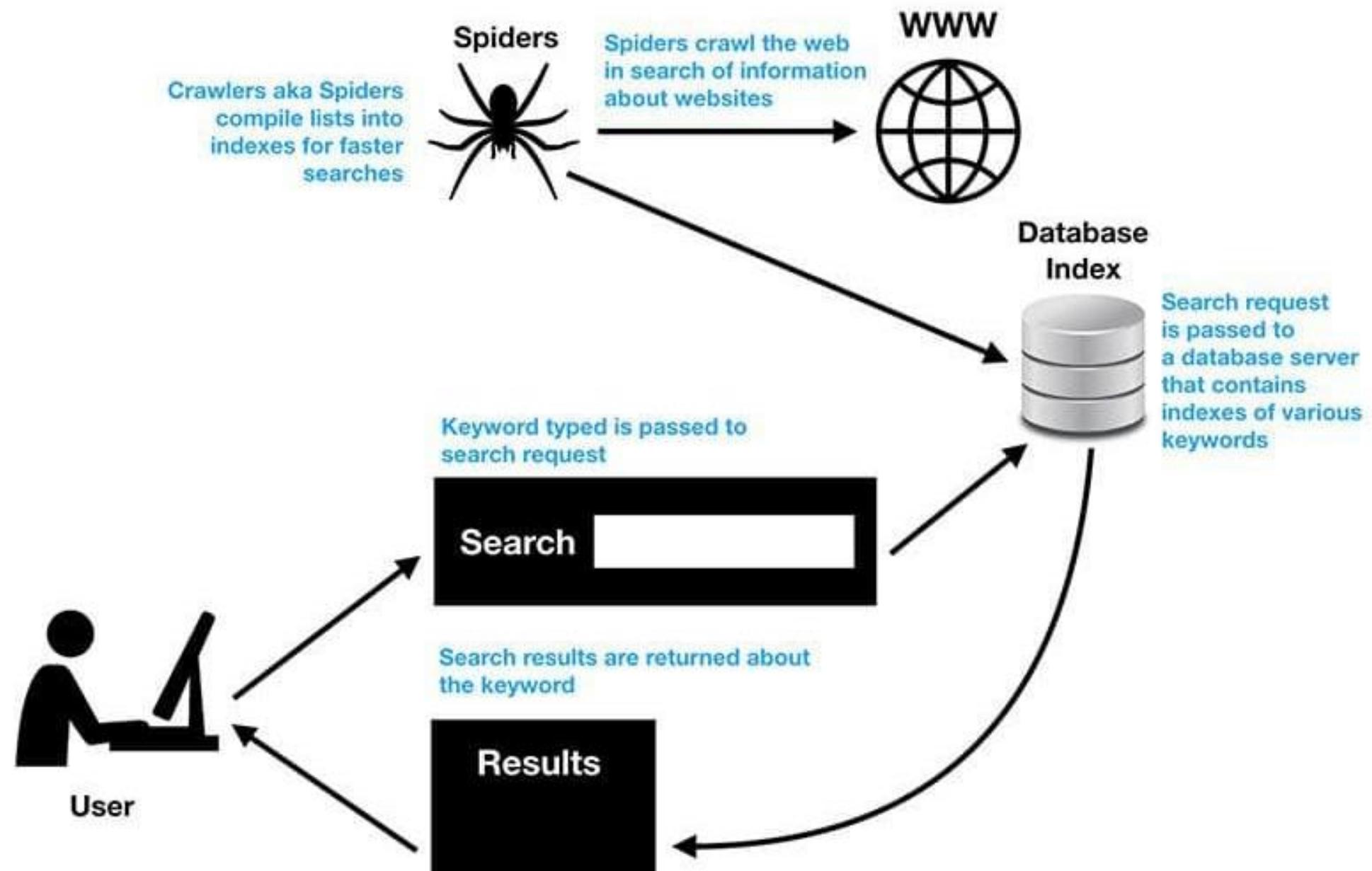


- What do my customer buy? Which product are bought together?
- **Aim:** Find associations and correlations between the different items that customers place in their shopping basket

Data Mining Models : for example Decision tree



Data Mining Models : for example Search Engines

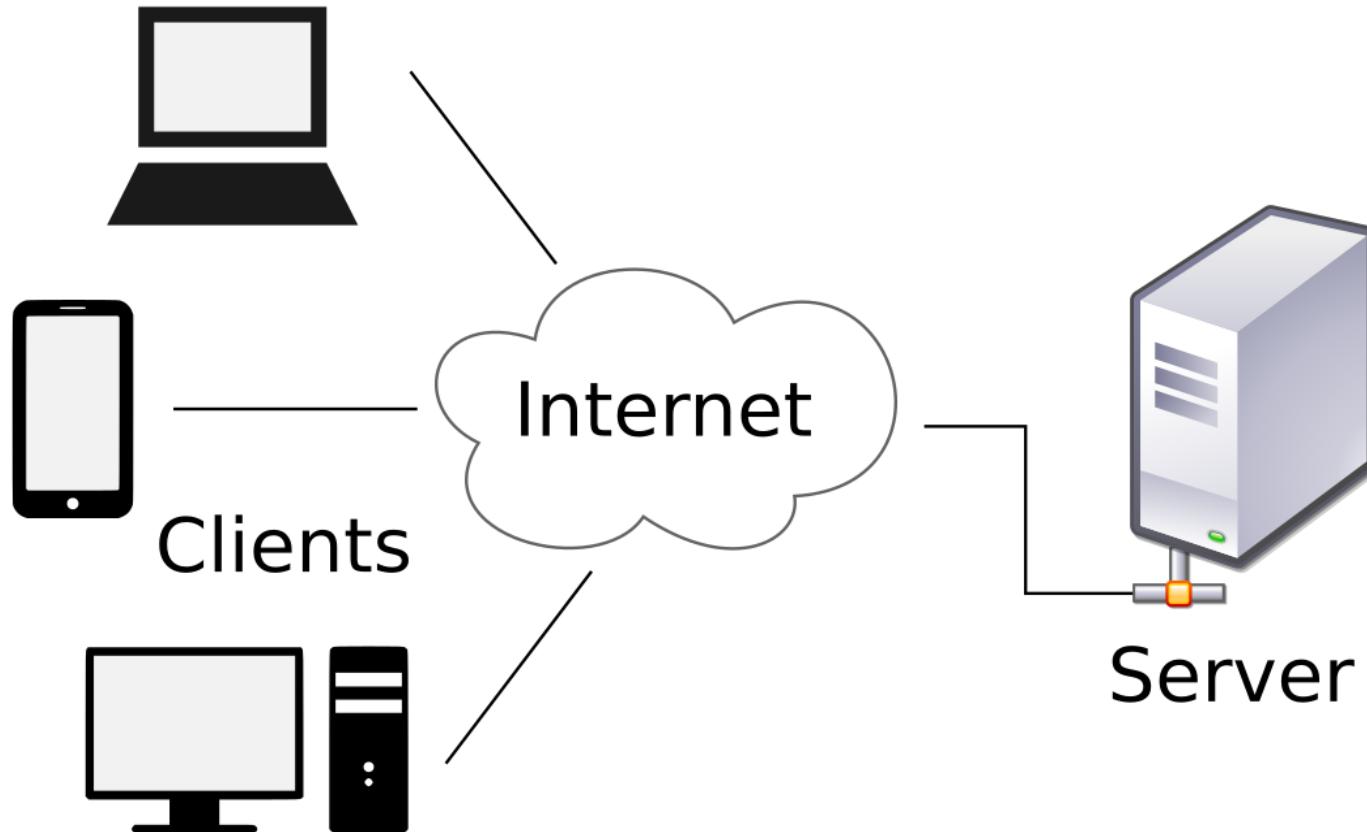


Python Web Development



django

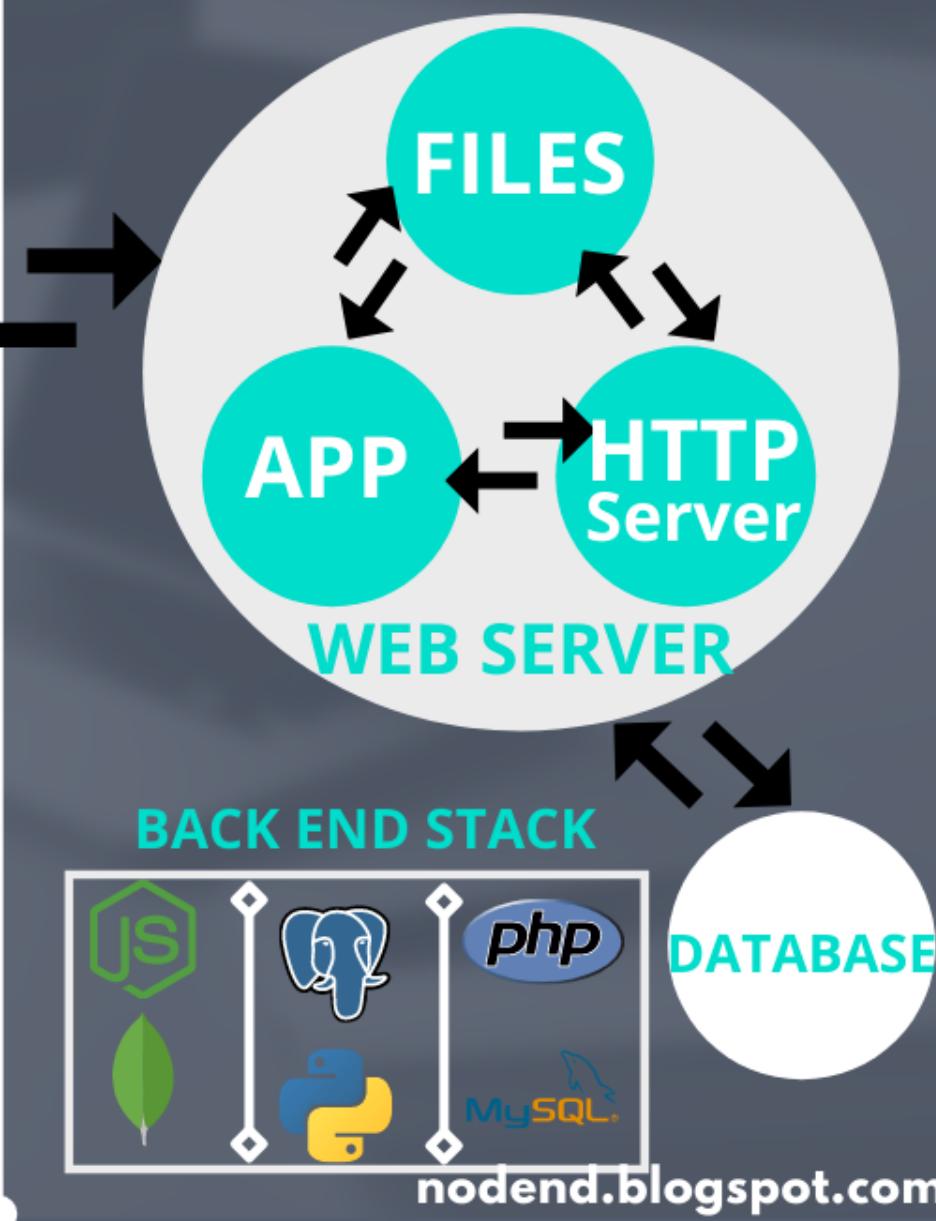
Client Server Architecture



FRONT END



BACK END



django



Python Web Frameworks

WEB2PY



SANIC
FRAMEWORK





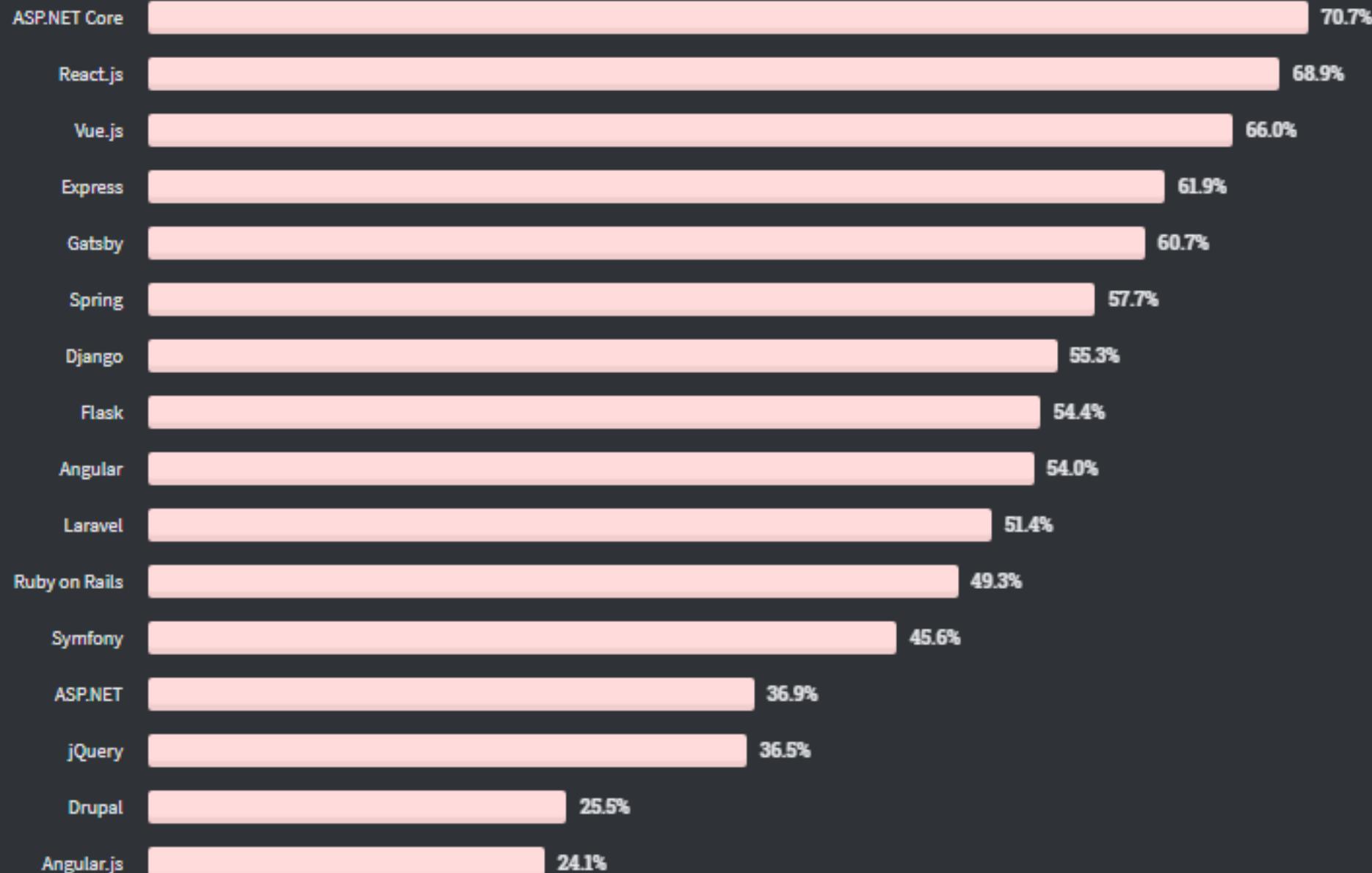
The 7 Backend Frameworks with the most stars on GitHub

3 Feb 2021





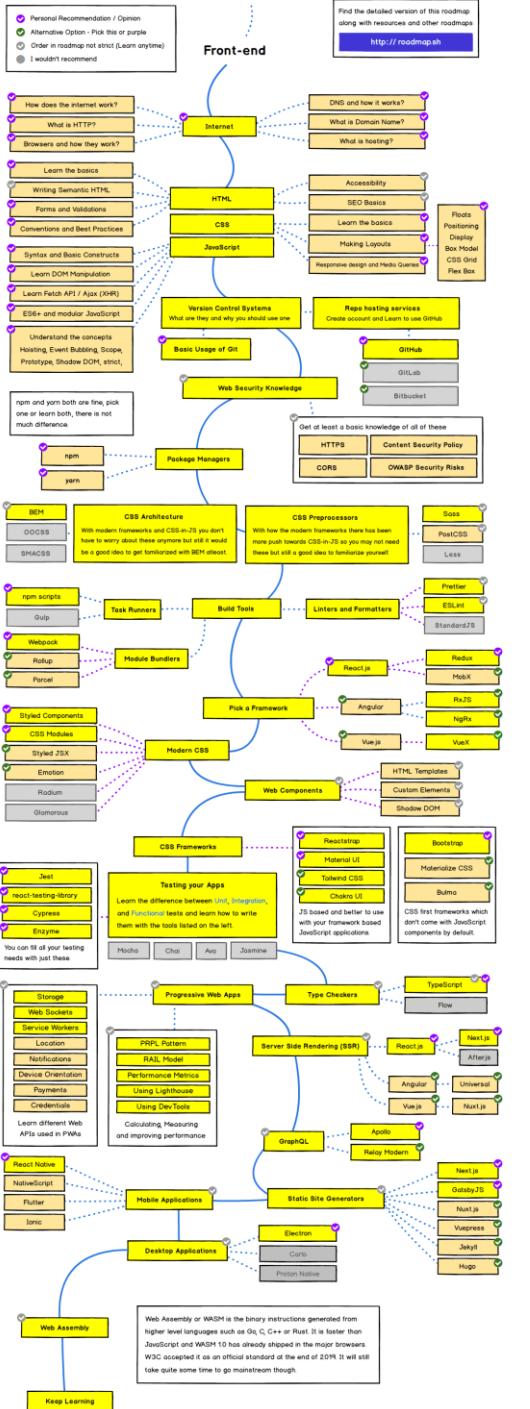
Stack overflow survey



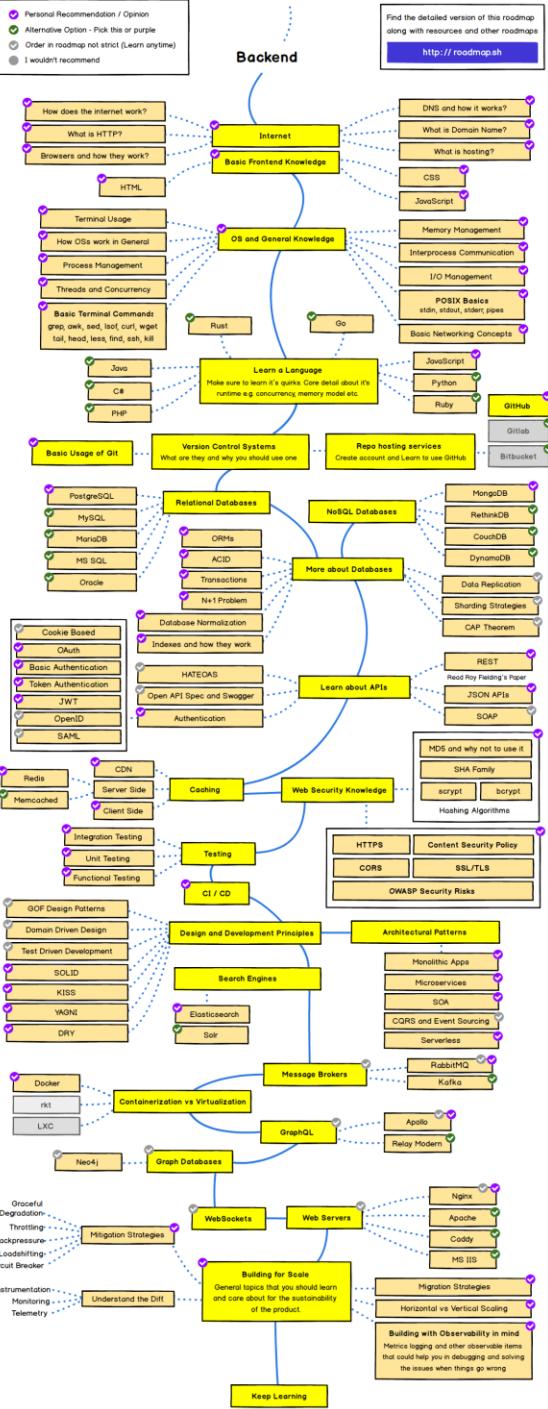
Web application frameworks components

- URL Routing
- Request and Response Objects
- Template Engine
- Development Web Server
- Database Object Relational Mapper
- Form Validation

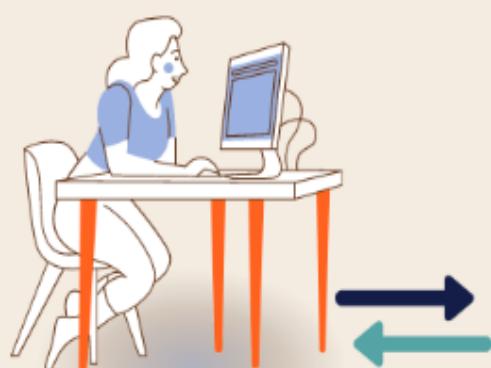
Front-end Roadmap



Back-end Roadmap



MVT pattern



Rose

Requests
the url

Django

dj



Url

<https://geekflare.com/author/daniel>

Recibes a **request**, and returns a
response

View



django



Template



Model

Database



Interacts, with the

- Instagram.
- National Geographic
- Mozilla.
- Spotify.
- Pinterest.
- Disqus.
- Bitbucket.
- Eventbrite.



django Start Project (Web Application)

\$ django-admin startproject mysite

Let's look at what `startproject` created

```
mysite/
  manage.py
  mysite/
    __init__.py
    settings.py
    urls.py
    asgi.py
    wsgi.py
```

- The outer `mysite/` root directory is a container for your project. Its name doesn't matter to Django; you can rename it to anything you like.
- `manage.py`: A command-line utility that lets you interact with this Django project in various ways. You can read all the details about `manage.py` in [django-admin](#) and [manage.py](#).
- The inner `mysite/` directory is the actual Python package for your project. Its name is the Python package name you'll need to use to import anything inside it (e.g. `mysite.urls`).
- `mysite/__init__.py`: An empty file that tells Python that this directory should be considered a Python package. If you're a Python beginner, read [more about packages](#) in the official Python docs.
- `mysite/settings.py`: Settings/configuration for this Django project. [Django settings](#) will tell you all about how settings work.
- `mysite/urls.py`: The URL declarations for this Django project; a "table of contents" of your Django-powered site. You can read more about URLs in [URL dispatcher](#).
- `mysite/asgi.py`: An entry-point for ASGI-compatible web servers to serve your project. See [How to deploy with ASGI](#) for more details.
- `mysite/wsgi.py`: An entry-point for WSGI-compatible web servers to serve your project. See [How to deploy with WSGI](#) for more details.

```
$ python manage.py runserver
```

django

[View release notes for Django 3.1](#)



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



Django Documentation
Topics, references, & how-to's



Tutorial: A Polling App
Get started with Django



Django Community
Connect, get help, or contribute

django Create multiple Apps in web application

```
$ python manage.py startapp polls
```

That'll create a directory **polls**, which is laid out like this:

```
polls/
    __init__.py
    admin.py
    apps.py
    migrations/
        __init__.py
    models.py
    tests.py
    views.py
```

django View : Route Controller

polls/views.py

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, world. You're at the polls index.")
```

django URLs

mysite/urls.py

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('polls/', include('polls.urls')),
    path('admin/', admin.site.urls),
]
```

django Models : Data Base Table (Using Django-ORM)

polls/models.py

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

django Installed app in settings.py

```
mysite/settings.py
```

```
INSTALLED_APPS = [  
    'polls.apps.PollsConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

Now Django knows to include the **polls** app. Let's run another command:

```
$ python manage.py makemigrations polls
```

Django-ORM makemigration on database SQL Code for our Table

```
BEGIN;  
--  
-- Create model Question  
--  
CREATE TABLE "polls_question" (  
    "id" serial NOT NULL PRIMARY KEY,  
    "question_text" varchar(200) NOT NULL,  
    "pub_date" timestamp with time zone NOT NULL  
);  
--  
-- Create model Choice  
--  
CREATE TABLE "polls_choice" (  
    "id" serial NOT NULL PRIMARY KEY,  
    "choice_text" varchar(200) NOT NULL,  
    "votes" integer NOT NULL,  
    "question_id" integer NOT NULL  
);  
ALTER TABLE "polls_choice"  
    ADD CONSTRAINT "polls_choice_question_id_c5b4b260_fk_polls_question_id"  
        FOREIGN KEY ("question_id")  
        REFERENCES "polls_question" ("id")  
        DEFERRABLE INITIALLY DEFERRED;  
CREATE INDEX "polls_choice_question_id_c5b4b260" ON "polls_choice" ("question_id");  
COMMIT;
```

django Migrate : Create Model and SQL for our Table

```
$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, polls, sessions
Running migrations:
  Rendering model states... DONE
  Applying polls.0001_initial... OK
```

The [migrate](#) command takes all the migrations that haven't been applied (Django tracks which ones are applied using a special table in your database called [django_migrations](#)) and runs them against your database - essentially, synchronizing the changes you made to your models with the schema in the database.

Migrations are very powerful and let you change your models over time, as you develop your project, without the need to delete your database or tables and make new ones - it specializes in upgrading your database live, without losing data. We'll cover them in more depth in a later part of the tutorial, but for now, remember the three-step guide to making model changes:

django Template.html

polls/templates/polls/index.html

```
{% if latest_question_list %}  
    <ul>  
        {% for question in latest_question_list %}  
            <li><a href="/polls/{{ question.id }}/">{{ question.question_text }}</a></li>  
        {% endfor %}  
    </ul>  
{% else %}  
    <p>No polls are available.</p>  
{% endif %}
```

django Admin panel

PP Select invite to chan x

localhost:8000/admin/academy/invite/

Django administration

Welcome, JohnDoe. Change password / Log out

Home > Academy > Invites

Select invite to change

Action: ----- ▾ Go 0 of 100 selected

<input type="checkbox"/>	Name	Branch	Gender	Date of birth	Race
<input type="checkbox"/>	Meredith Wechter	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	Beth Swofford	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	Melanie Ramsayer	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	David Pringle	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	Joel Lubin	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	David Kramer	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	Paul Christopher Hook	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	Robert Hohman	Associates	Unknown	(None)	Unknown
<input type="checkbox"/>	Joe Funicello	Associates	Unknown	(None)	Unknown

Add invite +

Filter

By branch

- All
- Actor
- Associates
- Casting directors
- Cinematographers
- Costume designers
- Designers
- Directors
- Documentary
- Executives
- Film editors
- Makeup Artists and
- Hairstylists
- Members-At-Large
- Music
- Producers



شروع کا، با میکرو فریم و ک قد، تمند و جذاب فلساک

- Micro Framework
- Simple but extensible core
- Werkzeug WSGI toolkit
- Jinja2 template engine
- Provides a simple template for web development
- No ORM
- No Form validation
- Supports extensions





Base on WSGI and Jinja2



WSGI



Template Engine

companies using flask

- Uber
- Red Hat
- Netflix
- Reddit
- Pinterest
- Linkedin
- Mozilla
- Hotjar
- Nginx
- Airbnb
- Lyft
- Mozilla
- MIT



Red Hat



RackSpace



Airbnb



UBER



Netflix



Lyft



Reddit



Flask Features

- Built in development server and debugger
- Integrated unit testing support
- RESTful request dispatching
- Uses Jinja2 template engine
- Support for secure cookies (client side sessions)
- 100% WSGI 1.0 compliant
- Unicode based
- Extensively documented



```
PyGram
├── app.py
├── requirements.txt
├── dependencies.txt
├── routes.txt
├── README.md
├── .env
├── .gitignore
├── Documentation
└── migrations
    └── PyGram
        ├── __init__.py
        ├── controller.py
        ├── models.py
        ├── config.py
        ├── mod_direct_message
        │   ├── __init__.py
        │   └── controller.py
        ├── mod_follow
        │   ├── __init__.py
        │   └── controller.py
        ├── mod_post
        │   ├── __init__.py
        │   └── controller.py
        ├── mod_profile
        │   ├── __init__.py
        │   └── controller.py
        ├── mod_search
        │   ├── __init__.py
        │   └── controller.py
        ├── mod_setting
        │   ├── __init__.py
        │   └── controller.py
        ├── mod_user_page
        │   ├── __init__.py
        │   └── controller.py
        └── static
            ├── Site-Screen
            ├── img
            ├── styles
            │   └── all of css files..
            └── upload
                ├── dir_of_user1
                ├── dir_of_user2
                └── dir_of_user3
    └── Templates
        └── all of html files..
    └── utils
        ├── check_auth.py
        ├── registration.py
        └── upload.py
```

یک ساختار پیشنهادی : ساختاری که در پروژه شبکه اجتماعی پایگرام استفاده شده است.

<https://github.com/M-Taghizadeh/PyGram>

Simple web application with Flask 😊

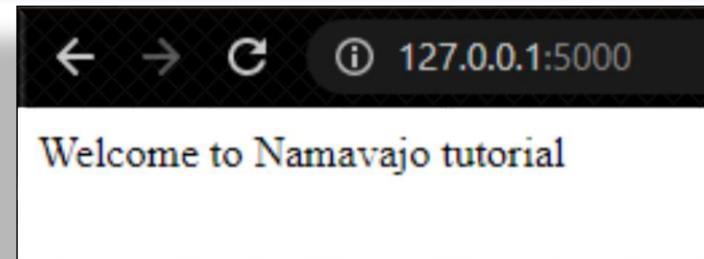


```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Welcome to Namavajo tutorial"

if __name__ == "__main__":
    app.run()
```



福 挑战 Ninja Template engine



```
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/hello")
@app.route("/hello/<name>")
def hello(name):
    return render_template('hello.html', name=name)
```

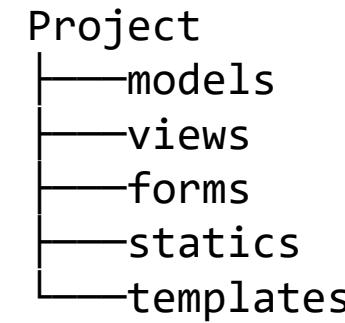
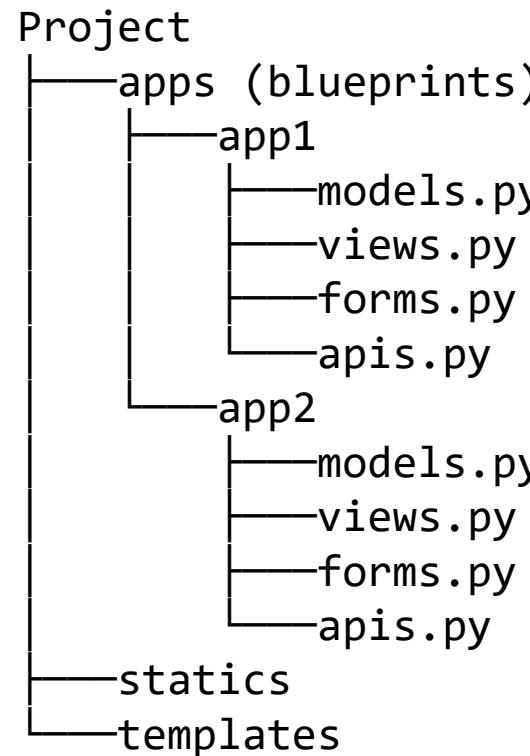
app.py

```
<!doctype html>
<head>
    <title>Hello from Flask</title>
</head>
<body>
    {% if name %}
        <h1>Hello {{ name }}</h1>
    {% else %}
        <h1>Hello World</h1>
    {% endif %}
</body>
</html>
```

Hello.html

Flask architectural pattern

یکی از دلایل جذابیت میکروفیمور ک محبوب فلسک ، داشتن آزادی عمل برنامه نویس در انتخاب هر نوع الگو برای ساختار توسعه پروژه است 😊



Comparison between Django and Flask

The Django logo consists of the word "django" in a white, lowercase, sans-serif font, centered within a dark teal rectangular background.

Type of Framework

Full Stack Web Framework

WSGI framework

Flexibility

Feature-packed

Full flexibility

ORM Usage

Built-in ORM

SQLAlchemy is used

Design

Batteries-included

Minimalistic design

Working Style

Monolithic

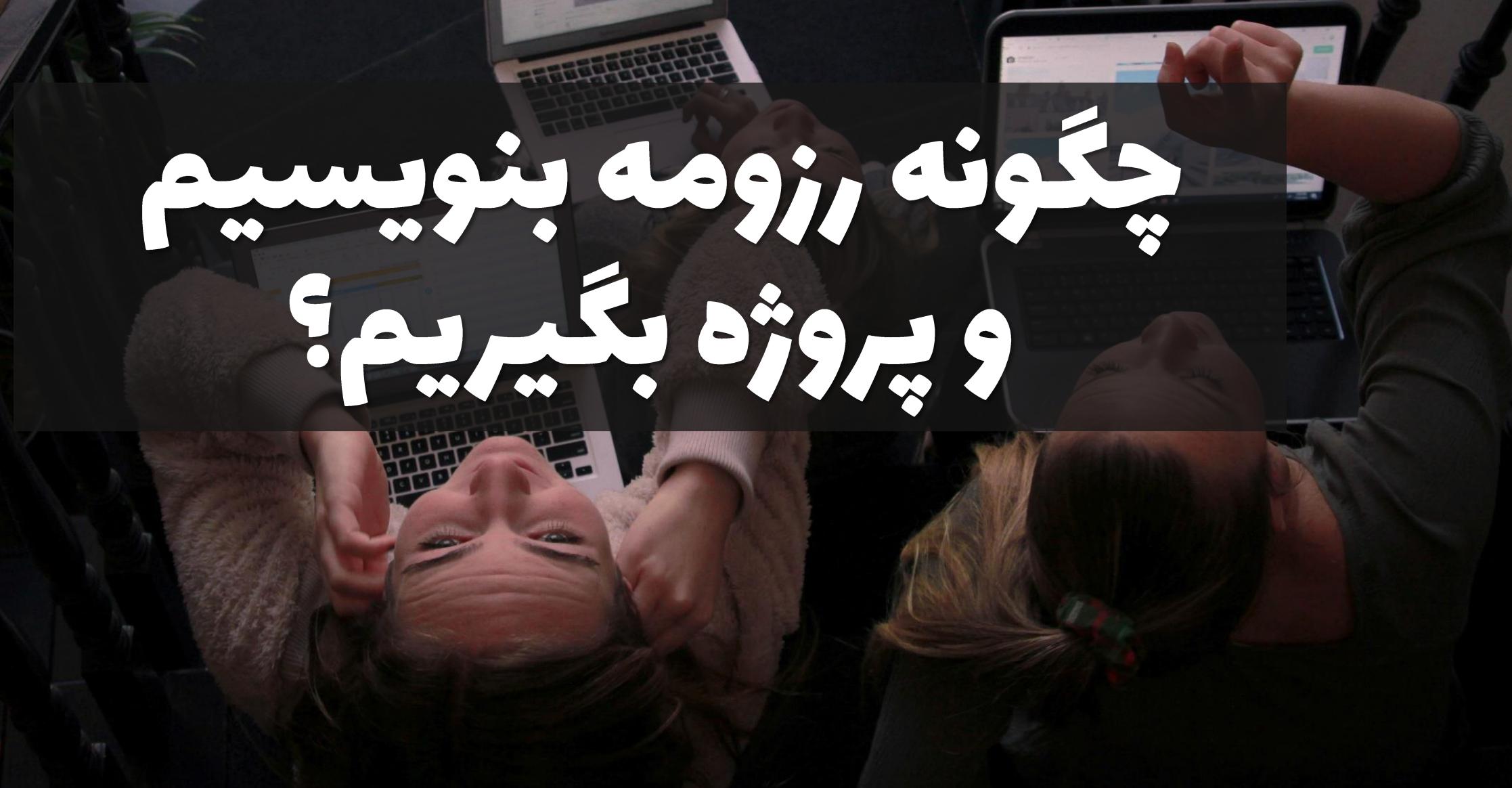
Diversified

Applications

Spotify, Instagram,
Dropbox, Disqus, etc

Linkedin, Pinterest,
Reddit, Uber, etc

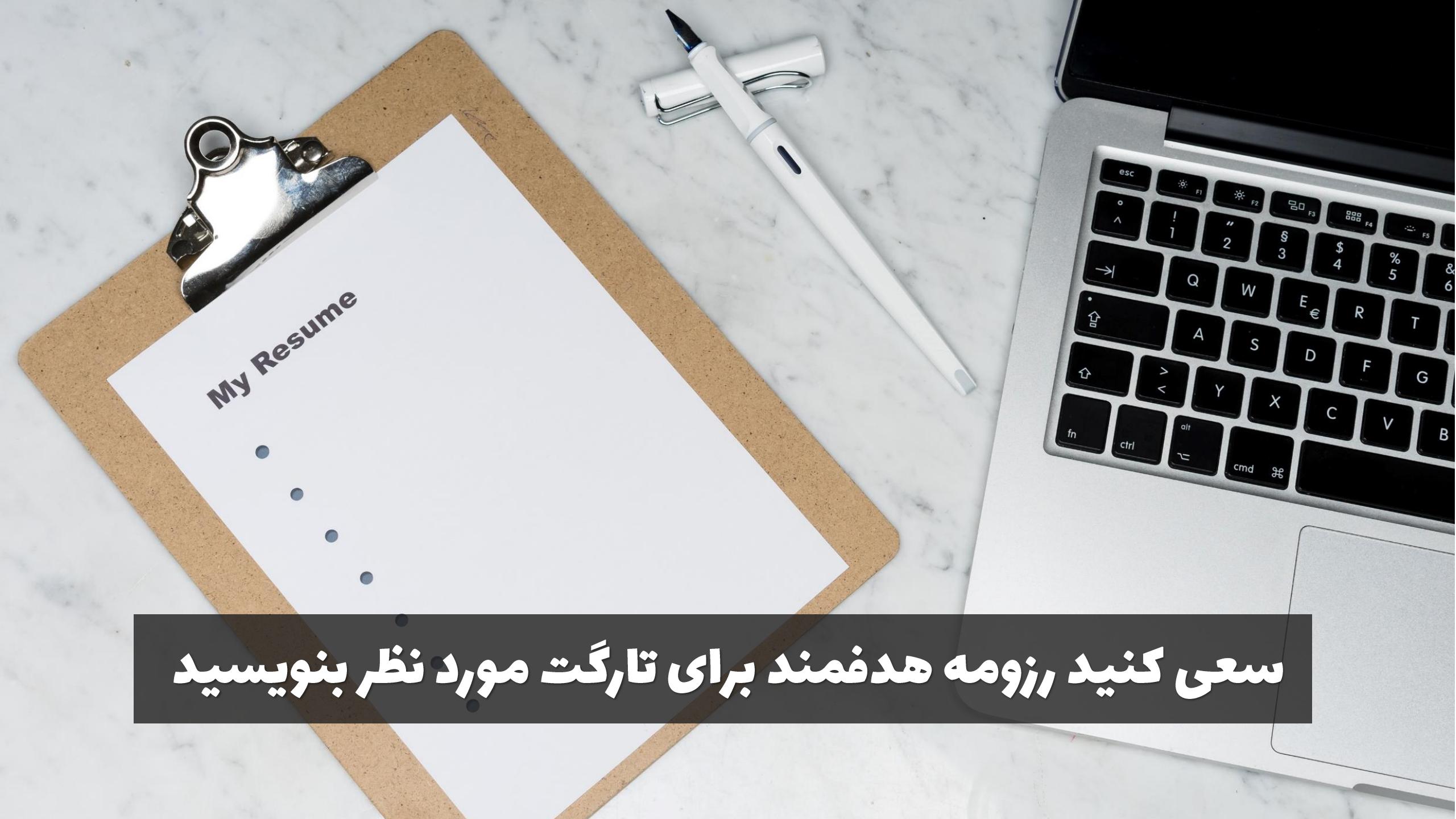
چگونه از ووده بنویسیم
و پژوهش بگیریم؟



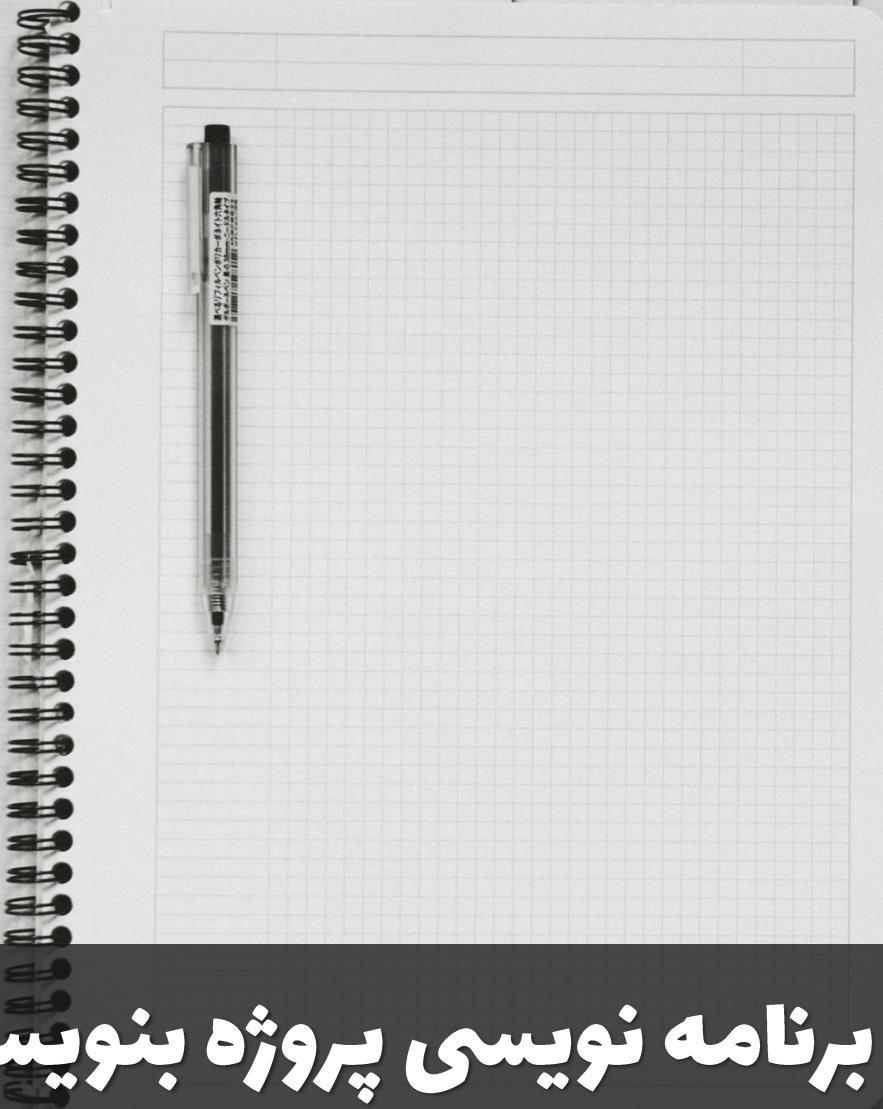
اولین قدم: خودتون وو در لینکدین به آدم ها معرفی کنید



سعی کنید ازومه هدفمند برای تارگت موبد نظر بنویسید



چطوری در اول مسیر برنامه نویسی پروژه بنویسیم؟

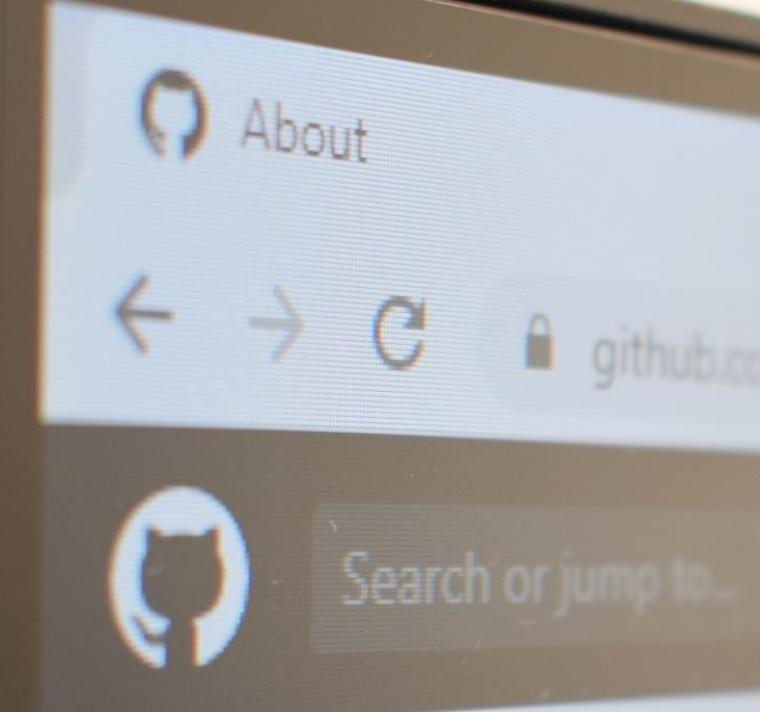


PROJECTS

بهر تین راهکار، در ابتدای مسیر آموزش برنامه نویسی پروژه های تمرینی تعریف کنید طوریکه دانشتون رو نشون بده و همینطور دوره های پروژه محور ببینید که در انتهای چند پروژه داشته باشید



گیتیاب رو برای پرزنست کردن مهارتتون و همکاری در
پروژه های متعدد باز به شدت جدی بگیرید

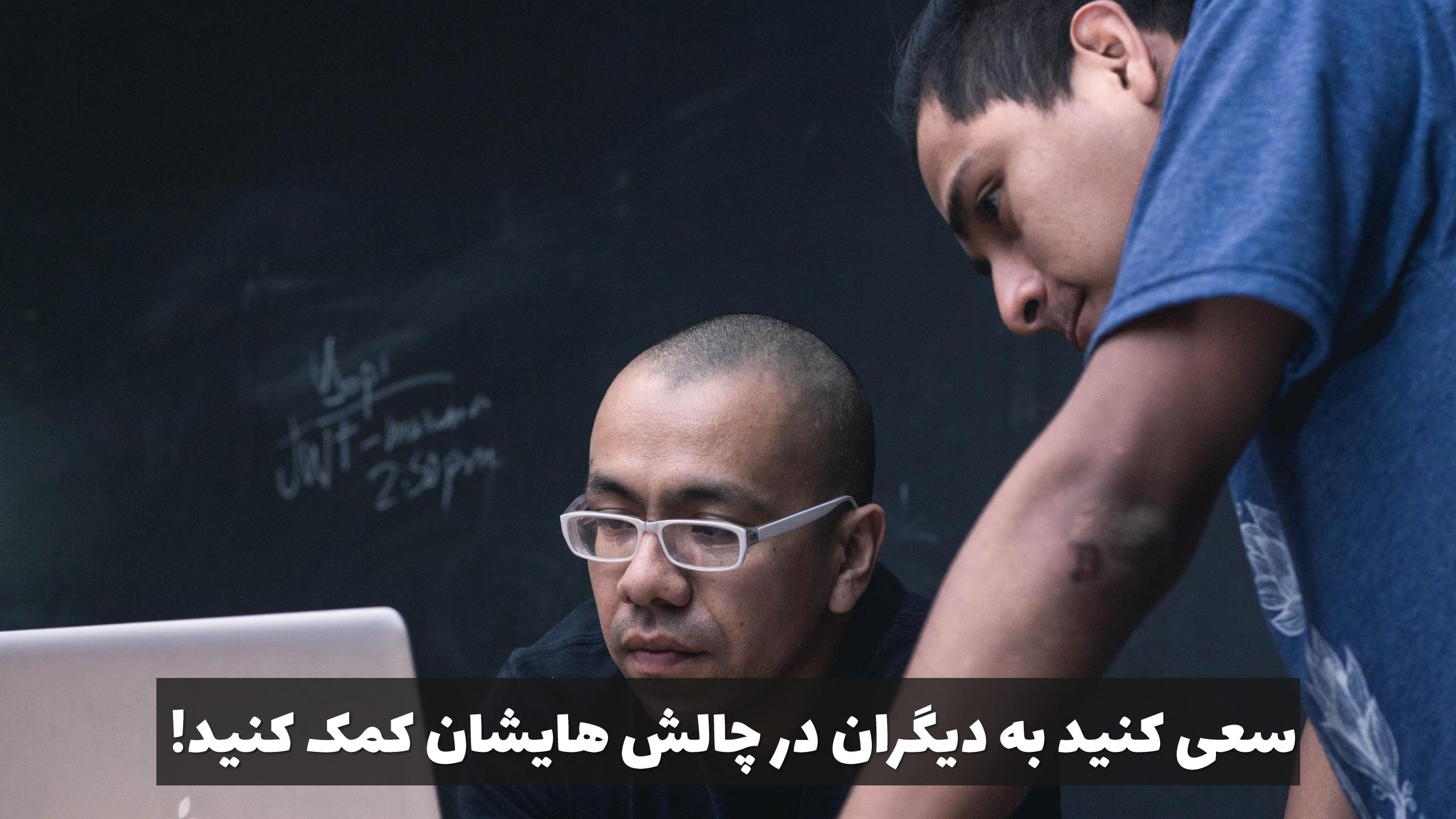


دوره جامع گیت و گیت‌هاپ برای برنامه نویسان حرفه ای

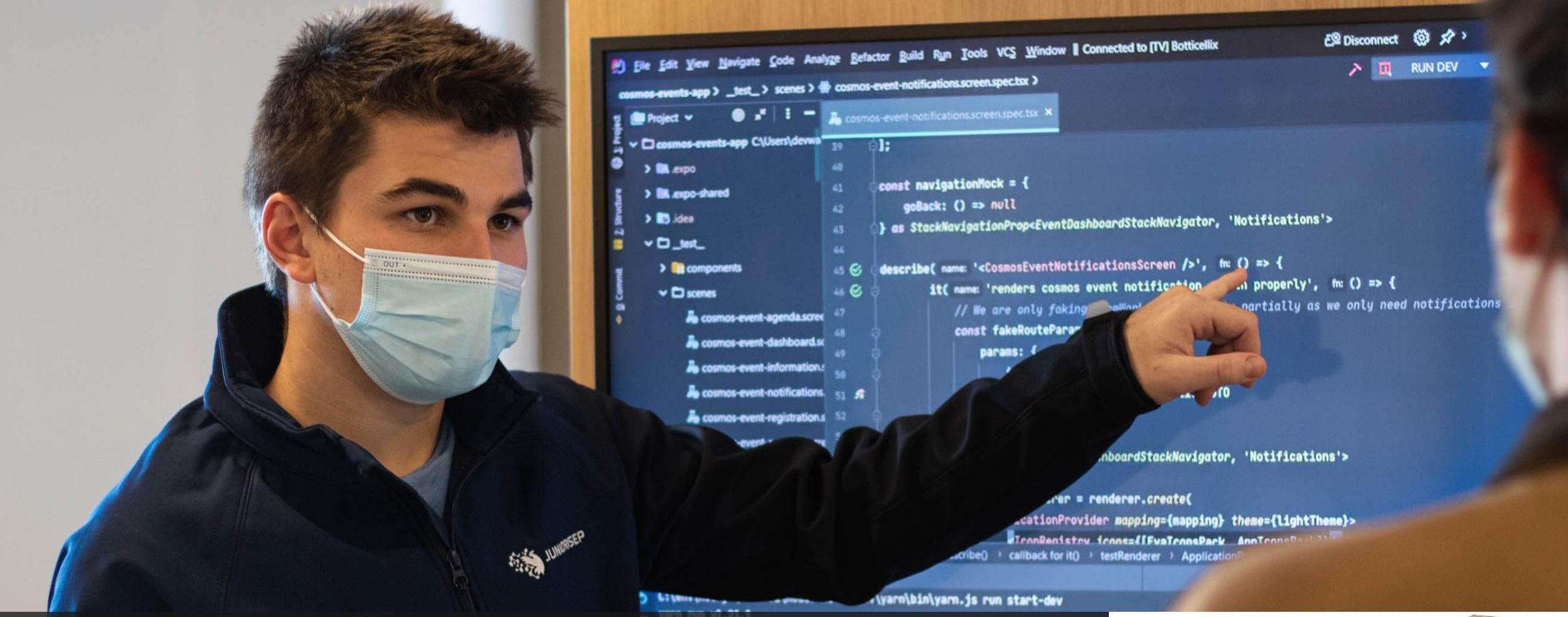
<https://www.daneshjooyar.com/learn-git-and-github/>



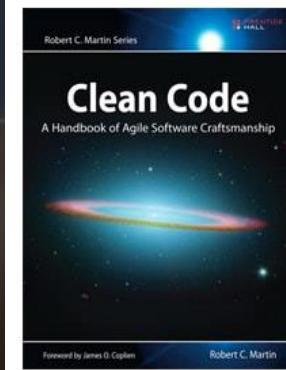
چگونه اعتماد آدم‌ها را برای همکاری شغلی جلب کنیم؟



سعی کنید به دیگران در چالش هایشان کمک کنید!



هم همیشه در حال یادگیری باشید هم به دیگران کنید
بترین اهکار برای پیشرفت دانش و مهارت فنی برنامه نویس
ها، افزایش دانش جمعی و کمک کامپیونیتی برنامه نویس ها
به یکدیگر است





شبکه های اجتماعی بستری به شدت مناسب برای پرزنست
کردن دانش ها و مهارت های شما هستند

User groups

ecture



Product Marketing



> _?



5 posts

Product Team Onboarding

Start here if you're new!

onboarding product

Kat Monroe

testing

Create new Article

info-sec

REPUTATION

1,280

NEXT TAG BADGE



BADGES

• 75 • 121

NEWEST

NEXT BADGE

• Explainer



How do I setup my local dev

Asked 10 days ago Active 2 days ago Viewed 11 times

Can someone give me an overview of the b
involved to run on my non-Windows OS?
9

People asked

Crystal Najera Shaun Beck



add a reaction



Answer from a subject matter expert

63

✓



کامیونیتی های برنامه نویسی و حتی جدید بگیرید

12 posts

Engineering Tech Stack

Documentation and discussions for the latest

engineering docs

asked Feb 11 at 9:23

144 answers

Eliot Godfrey

PR review champ

devops

چگونه برنامه نویس خوبی شویم؟

نکاتی در جهت یادگیری بہتر و با پروفومنس بیشتر برنامه نویسی

[Donate](#)

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

Browse the docs online or download a copy of your own.

Python's documentation, tutorials, and other resources.

Get started here, or scroll down for documentation releases by version.

[Python Docs](#)

See also [Documentation Releases by Version](#)

Docs

Audio/Visual Talks

Beginner's Guide

Developer's Guide

FAQ

Non-English Docs

PEP Index

Python Books

Python Essays

Python's standard documentation: download, browse or watch a tutorial.

Get started below, or visit the Documentation page to browse by version.

[Python Docs](#)

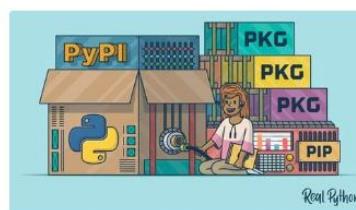
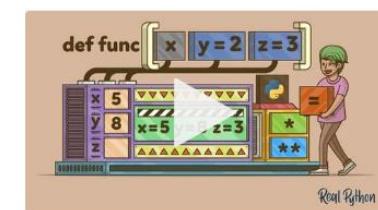
Beginner

Moderate

Advanced

General

سایت های مناسب برای افزایش دانش برنامه نویسی پایتون



Defining Python Functions
Optional Arguments

Real Python

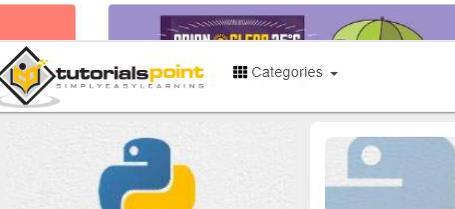
Feb 08, 2022 basics python



Python's len() Function



Draw the Mandelbrot Set



Using Python's pip to Manage Your Projects' Dependencies

Feb 02, 2022 basics tools



learnpython.org Home About Certify More Languages ▾

Python Java HTML Go C C++ JavaScript PHP Shell C# Perl Ruby

Get started learning Python with DataCamp's free [Intro to Python tutorial](#). Learn DataCamp challenges and watching videos by expert instructors. [Start Now!](#)

Ready to take the test? Head onto [LearnX](#) and get your Python Certification!

This site is generously supported by DataCamp. DataCamp offers online interactive Python courses for **575,000** other learners and get started learning Python for data science today!

Welcome

LearnPython

learn the Python

Welcome to the LearnPython website! Whether you are an experienced or beginner programmer, we have something for everyone.

You are welcome to join our group on [Facebook](#) for questions, discussions and updates.

After you complete the tutorials, you can get certified at [LearnX](#) and add your certification to your LinkedIn profile.

Just click on the chapter you wish to begin from, and follow the instructions. Good luck!

Learn the Basics

Python Tutorial

Python HOME

Python Intro

Python Get Started

Python Syntax

Python Comments

Python Variables

Python Data Types

Python Numbers

Python Casting

Python Strings

Python Booleans

Python Operators

Python Lists

Python Introduction

◀ Previous

What is Python?

Python is a popular programming language

It is used for:

- web development (server-side)
- software development,
- mathematics,
- system scripting.

released in 1991.
W3Schools

on do?

used on a server to create web applications.
used alongside software to create workflows.
connect to database systems. It can also read and modify files.
used to handle big data and perform complex mathematics.
used for rapid prototyping, or for production-ready software development.

GeeksforGeeks

Home Topic-wise Practice C++ Java Python Competitive Programming Machine Learning

on Programming Language

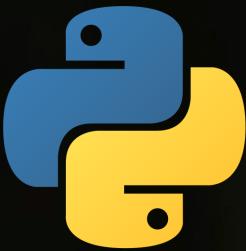
dated : 04 Sep, 2021

a high-level, general-purpose and a very popular programming language. Python programming is designed to be highly readable using English keywords frequently where possible. It allows programmers to express concepts in fewer lines of code than some other languages. Python has a large standard library and a large number of third-party libraries available for download. Machine learning applications, scientific computing, web development, and more are built using Python. GeeksforGeeks



PYTHON

INPUT/OUTPUT
PANDAS
DOP OPERATORS
DATA TYPES
EXCEPTION
JAVA
JQUERY
C++



Python Podcasts

Talk Python to Me focuses on the people and organizations coding on Python. Each episode features a different guest interviewee to talk about his or her work

Podcast.__init__ is another regular podcast that presents stories about Python and interviews "with the people who make it great"

The Real Python Podcast is a weekly podcast with interviews, coding tips, and conversation with guests from the Python community

Teaching Python is a podcast by two teachers about their adventures teaching middle school computer science, problem solving, handling failure, frustration, and success with teaching Python programming

Django Riffs is a podcast for learning web application development in Python using the Django web framework

Python Bytes is a new podcast from the creators of the above mentioned "Talk Python to Me" and "Test and Code Podcast"



پروفومنس برنامه ها: چالش پیش روی برنامه نویسان پایتون

Benchmarks

Python 3 versus C gcc fastest programs

vs C vs Go vs Java vs JavaScript

Always look at the source code.

These are only the fastest programs. Do some of them use manually vectorized SIMD? Look at the other programs.

They may seem more-like a *fair* comparison to you.

regex-redux

source	secs	mem	gz	busy	cpu load
<u>Python 3</u>	1.36	111,852	1403	2.64	32% 40% 33% 88%
<u>C gcc</u>	0.80	152,172	1397	2.01	52% 99% 48% 53%

pidigits

source	secs	mem	gz	busy	cpu load
<u>Python 3</u>	1.28	12,024	567	1.29	0% 1% 100% 0%
<u>C gcc</u>	0.59	2,444	1090	2.37	100% 100% 100% 98%

reverse-complement

source	secs	mem	gz	busy	cpu load
<u>Python 3</u>	7.20	1,005,184	814	10.75	20% 53% 48% 29%
<u>C gcc</u>	0.86	712,208	820	1.27	99% 28% 1% 19%

k-nucleotide

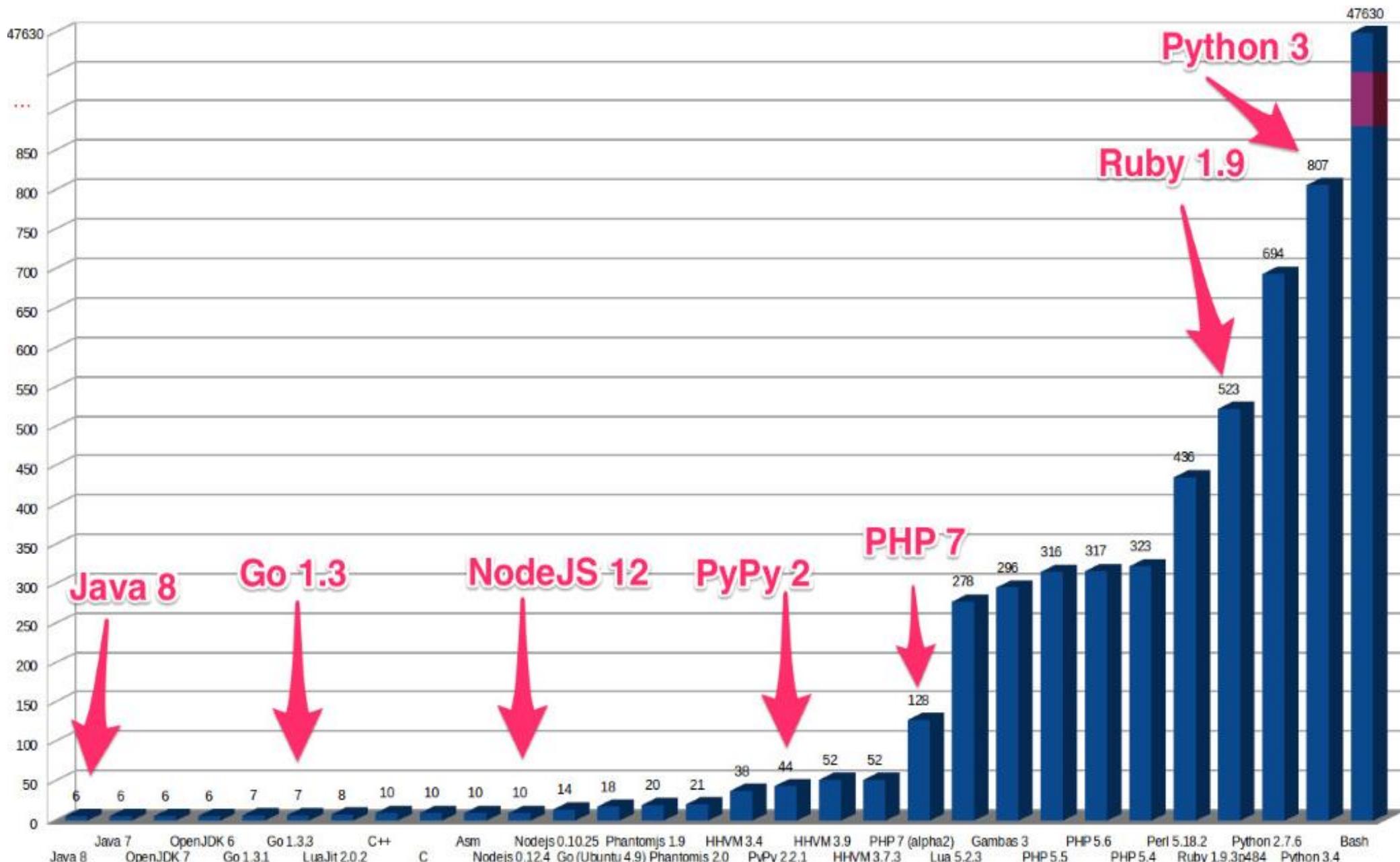
source	secs	mem	gz	busy	cpu load
<u>Python 3</u>	46.28	241,108	1967	176.42	94% 97% 95% 96%
<u>C gcc</u>	3.72	130,260	1506	12.07	100% 89% 78% 57%

binary-trees

source	secs	mem	gz	busy	cpu load
<u>Python 3</u>	48.03	462,732	472	174.44	89% 97% 88% 89%
<u>C gcc</u>	1.54	168,832	809	4.35	60% 67% 57% 100%

The Computer Language Benchmarks Game. Which programming language is fastest?

<https://benchmarksgame-team.pages.debian.net/benchmarksgame/index.html>



Python performance against other languages

<https://belitsoft.com/java-development-services/java-vs-python-tried-and-true-vs-modern-and-new>

Why is Python So Slow?

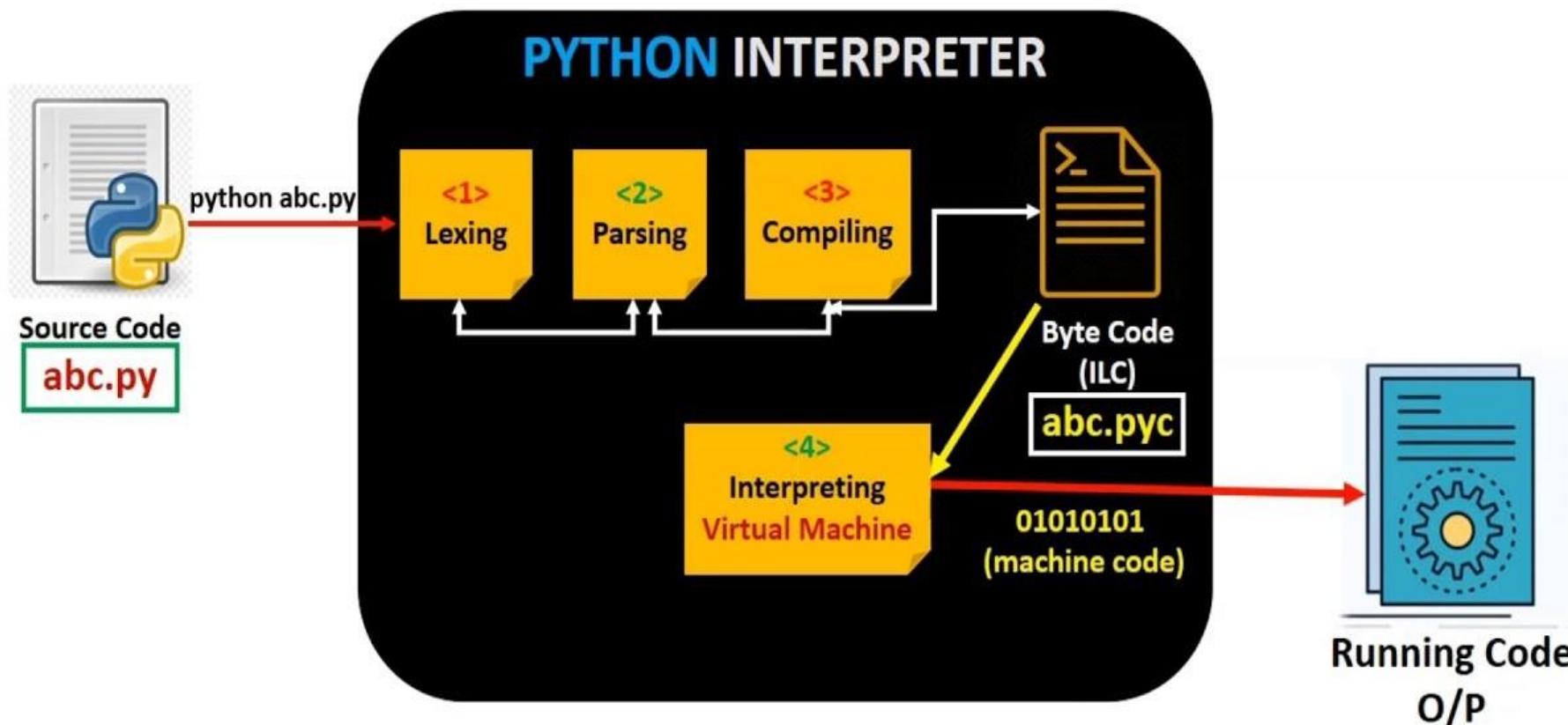
It's the GIL

It's because its an interpreted language

It's because its a dynamically typed language

Why is Python So Slow?

Simulating Python Interpreter



Why is Python So Slow?



```
1 # python_functions.py
2 #
3 import math
4
5 def f(x):
6     return math.exp(-(x ** 2))
7
8 def integrate_f(a, b, N):
9     s = 0
10    dx = (b - a) / N
11    for i in range(N):
12        s += f(a + i * dx)
13    return s * dx
14
```

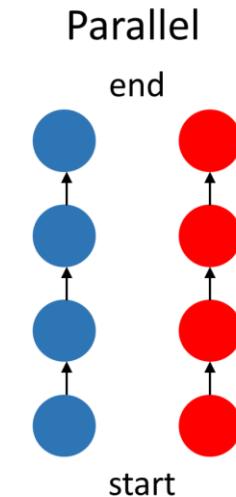
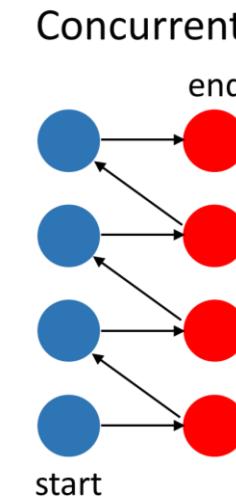
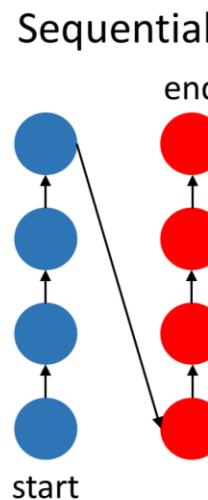
```
1 # cython_functions.pyx
2 #
3 from libc cimport math
4
5 cdef double f(double x):
6     return math.exp(-(x ** 2))
7
8 def integrate_f(double a, double b, int N):
9     cdef double s = 0
10    cdef double dx = (b - a) / N
11    cdef int i
12    for i in range(N):
13        s += f(a + i * dx)
14    return s * dx
15
```

راه حل افزایش پر فورمنس پایتون: افزایش سرعت برنامه های پایتونی با برنامه نویسی موازی و همرونده

Parallel and Concurrent Programming in Python



Threading
Asyncio
Multiprocessing





Q

A



<https://www.youtube.com/c/mohammadtaghizadeh>



<https://www.instagram.com/taghizadeh.me/>



<https://github.com/m-taghizadeh>



<https://www.linkedin.com/in/mtaghizadeh/>



@Tqzdh