

- ✓ OpenCV Tutorial
- OpenCV Tutorial

OpenCV Installation

Read & Save Images

Basic Operation On Images

OpenCV Resize Image

OpenCV Image Rotation

OpenCV Image Drawing Functions

OpenCV Blob Detection

Canny Edge Detection

OpenCV Gaussian Blur

OpenCV Image Filters

OpenCV Image Threshold

OpenCV Contours

OpenCV Mouse Event

OpenCV Template Matching

OpenCV Erosion & Dilation

OpenCV Video Capture

Face Recognition & Face Detection

Limitations in Face Detection

## OpenCV Blob Detection

Blob stands for **Binary Large Object** and refers to the connected pixel in the binary image. The term "Large" focuses on the object of a specific size, and that other "small" binary objects are usually noise. There are three processes regarding BLOB analysis.

### BLOB extraction

Blob extraction means to separate the BLOBs (objects) in a binary image. A BLOB contains a group of connected pixels. We can determine whether two pixels are connected or not by the connectivity, i.e., which pixels is neighbor of another pixel. There are two types of connectivity. The **8-connectivity** and the **4-connectivity**. The 8-connectivity is far better than 4-connectivity.

### BLOB representation

BLOB representation is simply means that convert the BLOB into a few representative numbers. After the BLOB extraction, the next step is to classify the several BLOBs. There are two steps in the BLOB representation process. In the first step, each BLOB is denoted by several characteristics, and the second step is to apply some matching methods that compare the features of each BLOB.

### BLOB classification

Here we determine the type of BLOB, for example, given BLOB is a circle or not. Here the question is how to define which BLOBs are circle and which are not based on their features that we described earlier. For this purpose, generally we need to make a prototype model of the object we are looking for.

```
import cv2
import numpy as np;

img = cv2.imread(r"filename", cv2.IMREAD_GRAYSCALE)
# Set up the detector with default parameters.
detector = cv2.SimpleBlobDetector()

# Detecting blobs.
keypoints = detector.detect(img)
# Draw detected blobs as red circles.
# cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS ensures the size of the circle corresponds to the size of blob
im_with_keypoints = cv2.drawKeypoints(img, keypoints, np.array([[]]), (0, 0, 255),
                                     cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

# Show keypoints
cv2.imshow("Keypoints", im_with_keypoints)
cv2.waitKey(0)
```

## How to perform Background Subtraction?

Background subtraction is widely used to generating a foreground mask. The binary images contain the pixels which belong to moving objects in the scene. Background subtraction calculates the foreground mask and performs the subtraction between the current frame and background model.

There are two main steps in Background modeling

- Background Initialization-** In this step, an initial model of the background is computed.
- Background Update-** In this step, that model is updated that adapt the possible change in the scene.

### Manual subtraction from the first frame

First, we import the libraries and load the video. Next, we take the first frame of the video, convert it into grayscale, and apply the Gaussian Blur to remove some noise. We use the while loop, so we load frame one by one. After doing this, we get the core part of the background of the subtraction where we calculate the absolute difference between the first frame and the current frame.

#### Example-1

```
import cv2
import numpy as np
cap = cv2.VideoCapture(0)

first_frame = cap.read()
first_gray = cv2.cvtColor(first_frame,)
first_gray_col = cv2.GaussianBlur(first_gray, (5, 5), 0)

while True:
    frame = cap.read()
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray_frame = cv2.GaussianBlur(gray_frame, (5, 5), 0)

    difference = cv2.absdiff(first_gray, gray_frame)
    difference = cv2.threshold(difference, 25, 255, cv2.THRESH_BINARY)
```

### Subtraction using Subtractor MOG2

OpenCV provides the subtractor MOG2 which is effective than the manual mode. The Subtractor MOG2 has the benefit of working with the frame history. The syntax is as follows:

```
cv2.createBackgroundSubtractorMOG2(history, varThreshold, detectShadow)
```

The first argument, history is the number of the last frame(by default 120).

The second argument, a **varThreshold** is the value that used when evaluating the difference to extract the background. A lower threshold will find more variation with the advantage of a noisier image.

The third argument, **detectShadows** is the functions of the algorithm which can remove the shadow if enabled.

#### Example-2:

```
import cv2
import numpy as np
cap = cv2.VideoCapture("filename")

subtractor = cv2.createBackgroundSubtractorMOG2(history=100, varThreshold=50, detectShadows=True)

while True:
    _, frame = cap.read()
    mask = subtractor.apply(frame)
    cv2.imshow("Frame", frame)
    cv2.imshow("mask", mask)
    key = cv2.waitKey(30)
    if key == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

In the above code, The **cv2.VideoCapture("filename")** accepts the full path included the file where the **cv2.createBackgroundSubtractorMOG2()** will exclude the background from the video file.

Next Topic

Canny Edge Detection

← prev

next →

## Help Others, Please Share



### Join Javatpoint Test Series

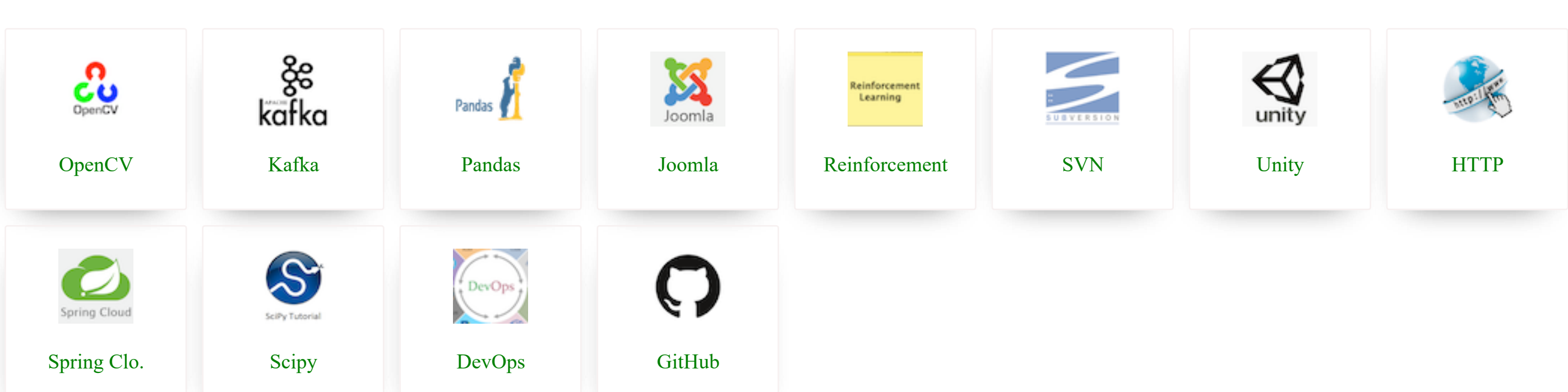
Placement Papers  
TCS  
HCL  
Infosys  
IBM  
Accenture

AMCAT  
eLitmas  
Java  
Python  
C Programming  
Networking

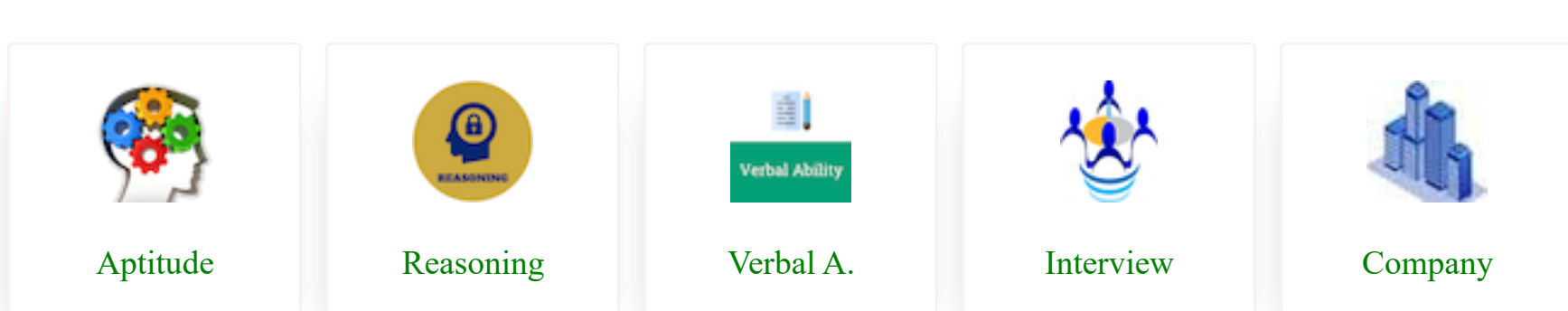
Bank PO/Clerk  
UPSSSC  
Government Exams  
SSC  
Civil Services  
SBI

GATE  
NEET  
CAT  
Railway  
CTET  
IIT JEE

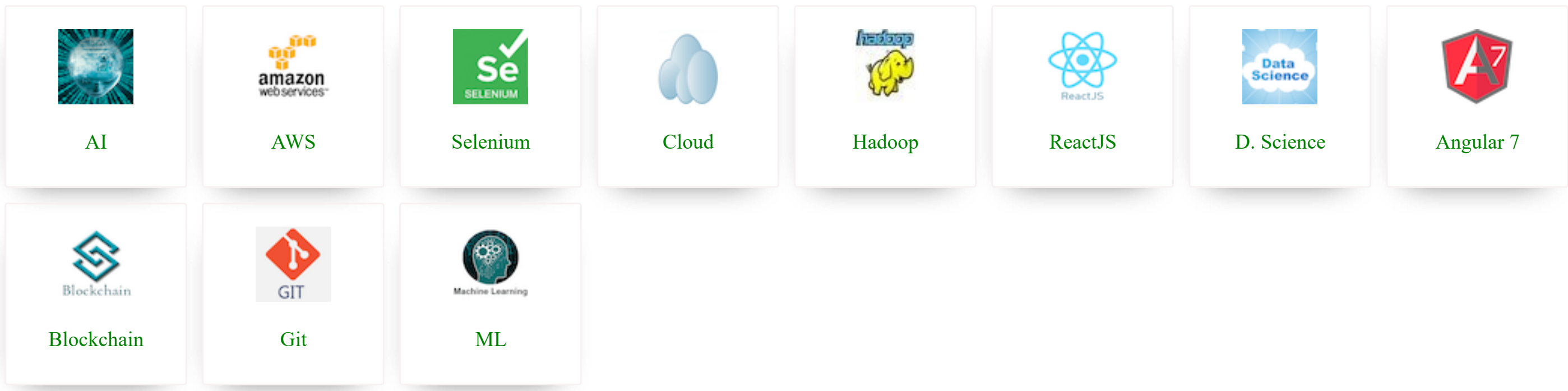
## Learn Latest Tutorials



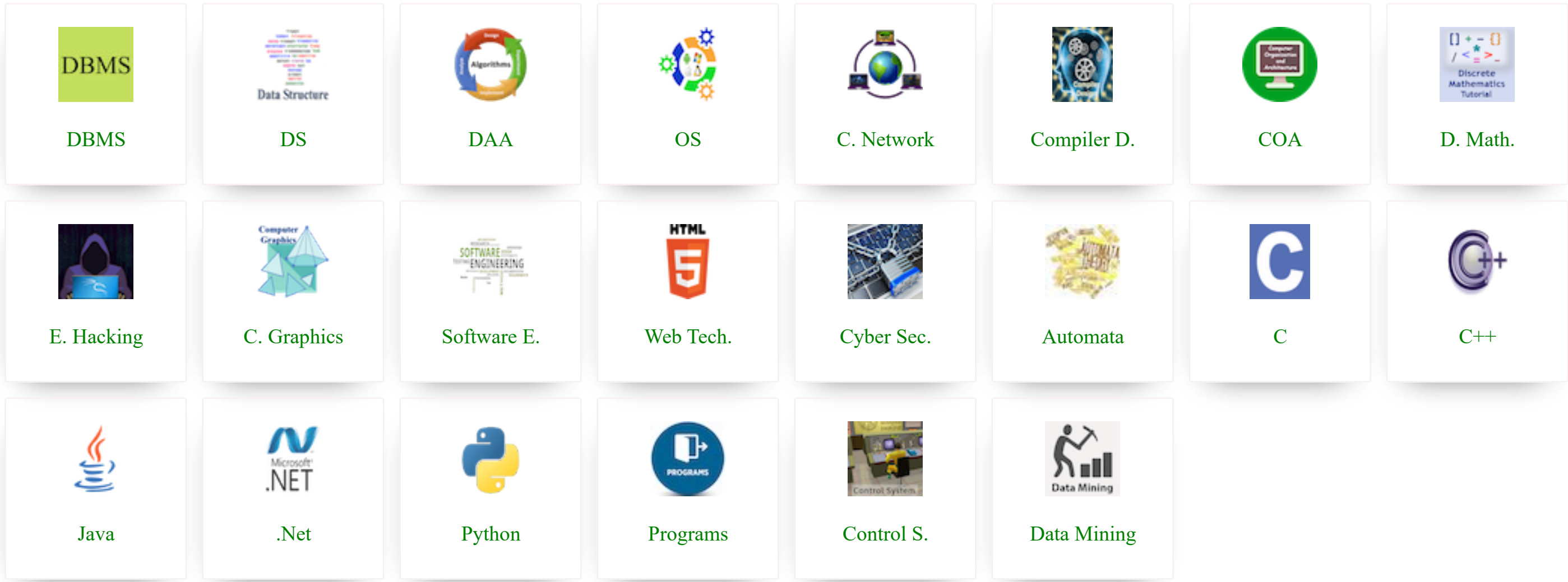
## Preparation



## Trending Technologies



## B.Tech / MCA



## Javatpoint Services

JavaTpoint offers too many high quality services. Mail us on [hr@javatpoint.com](mailto:hr@javatpoint.com), to get more information about given services.

- Website Designing
- Website Development
- Java Development
- PHP Development
- WordPress
- Graphic Designing
- Logo
- Digital Marketing
- On Page and Off Page SEO
- PPC
- Content Development
- Corporate Training
- Classroom and Online Training
- Data Entry

## Training For College Campus

JavaTpoint offers college campus training on Core Java, Advance Java, .Net, Android, Hadoop, PHP, Web Technology and Python. Please mail your requirement at [hr@javatpoint.com](mailto:hr@javatpoint.com).

Duration: 1 week to 2 week

Like/Subscribe us for latest updates or newsletter



### LEARN TUTORIALS

Learn Java  
Learn Data Structures  
Learn C Programming  
Learn C++ Tutorial  
Learn C# Tutorial  
Learn PHP Tutorial  
Learn HTML Tutorial  
Learn JavaScript Tutorial  
Learn jQuery Tutorial  
Learn Spring Tutorial

### OUR WEBSITES

Javatpoint.com  
Hindi100.com  
Lyricsin.com  
Quoteperson.com  
Jobandplacement.com

### OUR SERVICES

Website Development  
Android Development  
Website Designing  
Digital Marketing  
Summer Training  
Industrial Training  
College Campus Training

### CONTACT

Address: G-13, 2nd Floor, Sec-3  
Noida, UP, 201301, India  
Contact No: 0120-4256464, 9990449935  
Contact Us  
Subscribe Us  
Privacy Policy  
Sitemap