EXERCISE: Object Oriented Programming

AI VIET NAM
@aivietnam.edu.vn

Content

- **Review**
  - Iterator and Generator
  - Xml and Json in Python
  - Long/Short Focal Length Image
  - Basic Image Processing with OpenCV
    - Read & Write image
    - Color Conversions
    - Resize and Crop Image
    - Draw line
  - SIFT algorithm

Content

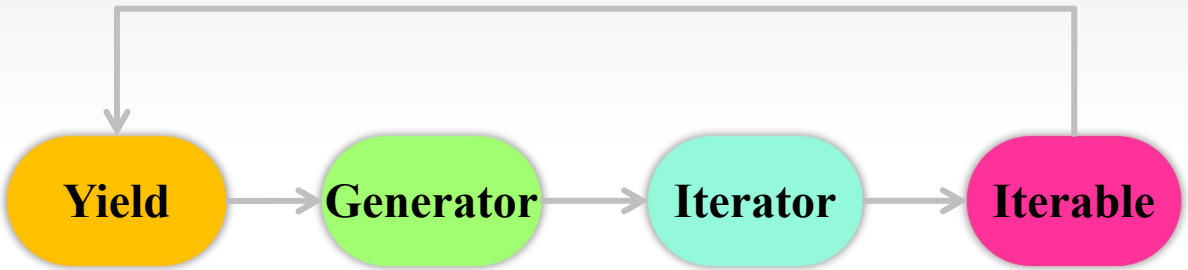
- **Exercise1: PASCAL VOC and CreateML Annotation Format**
  - Extract information from PASCAL VOC and CreatML annotation files for the Object Detection task
- **Exercise2: Google Books API**
  - Search, extract, and save necessary book information with the Google Books API
- **Exercise3: Compensation for Different Focal Length Images**
  - Compensate the short focal length image to achieve approximately the same result as the long focal length image

Content

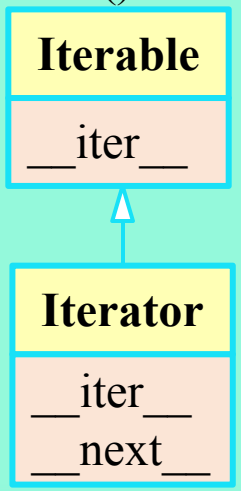
- **Review**
  - Iterator and Generator
  - Xml and Json in Python
  - Long/Short Focal Length Image
  - Basic Image Processing with OpenCV
    - Read & Write image
    - Color Conversions
    - Resize and Crop Image
    - Draw line
  - SIFT algorithm

Review

Iterator and Generator



Iterator: An iterable object with a next() method is an iterator




```
def __iter__(self):
    return self
```

Generator: Python provides a generator to create your iterator function. A special type of function which does not return a single value, instead, it returns an iterator object with a sequence of values. Do not need to create return list all at once, just need to **remember** its **creation rules**

```
def my_generator(max_num):
    n = 0
    while n <= max_num:
        yield n
        n += 1
```

Iterable: An iterable is an object that can return an iterator (to return all of its elements)
Ex: list, strings, ...

```
mylist = ['a', 'b', 'c']
for i in mylist:
    print(i)
```

Yield: You can think of yield as a return, but what it returns is a generator

```
def test():
    print("First")
    yield 1
    print("Second")
    yield 2
    print("Third")
    yield 3

```

https://python.plainenglish.io/python-yield-iterator-and-generator-introduction-15be182f6135

Review

```python
class AdditionValueSeries:
    '''
    Comment ...
    '''

    def __init__(self, value, max=0):
        self.value = value
        self.max = max

    def __iter__(self):
        self.n = 0
        return self

    def __next__(self):
        if self.n <= self.max:
            result = self.n + self.value
            self.n += 1

            return result
        else:
            raise StopIteration
```

```python
# Generator

def addValueSeries(value, max=0):
    n = 0
    while n <= max:
        yield n + value
        n += 1
```

```python
# create an object
numbers = AdditionValueSeries(5, 2)

# create an iterator
a_iter = iter(numbers)

# get iterator elements
print(next(a_iter))
print(next(a_iter))
print(next(a_iter))
```

```
5
6
7
```

```python
numbers = addValueSeries(5, 2)

# create an iterable
a_iter = iter(numbers)

# get iterator elements
print(next(a_iter))
print(next(a_iter))
print(next(a_iter))
```

```
5
6
7
```

Iterator and Generator

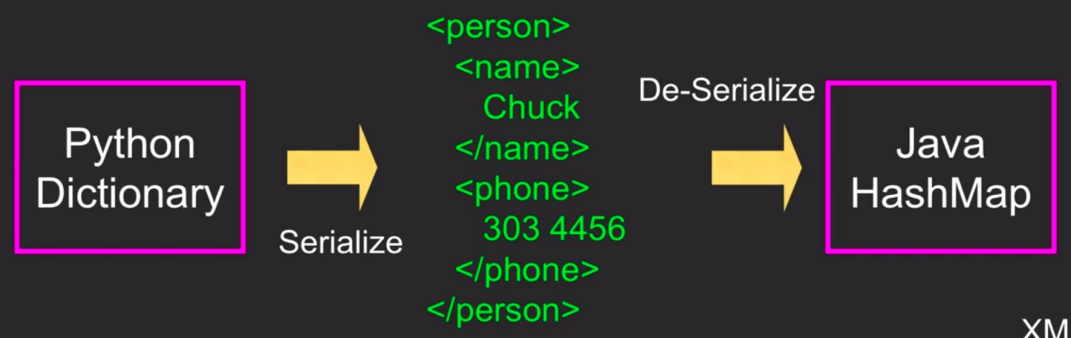
Review

Xml and Json in Python

Sending Data across the "Net"

a.k.a. "Wire Protocol" - What we send on the "wire"

Agreeing on a "Wire Format"



XML Basics

- Start Tag
- End Tag
- Text Content
- Attribute
- Self Closing Tag

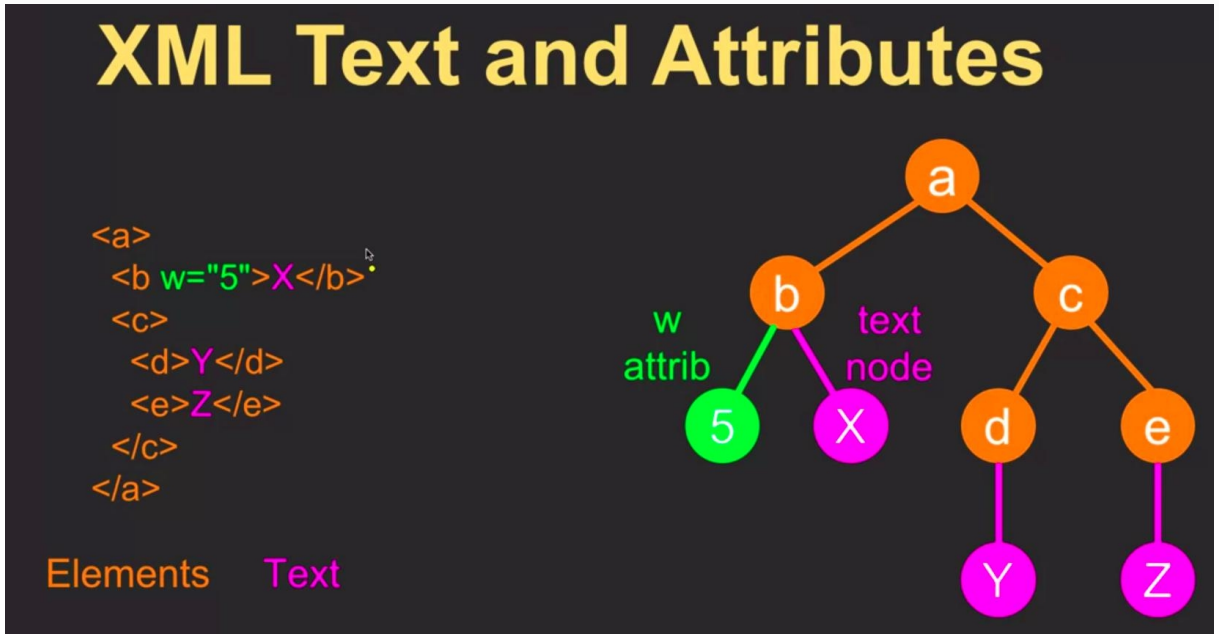
```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```

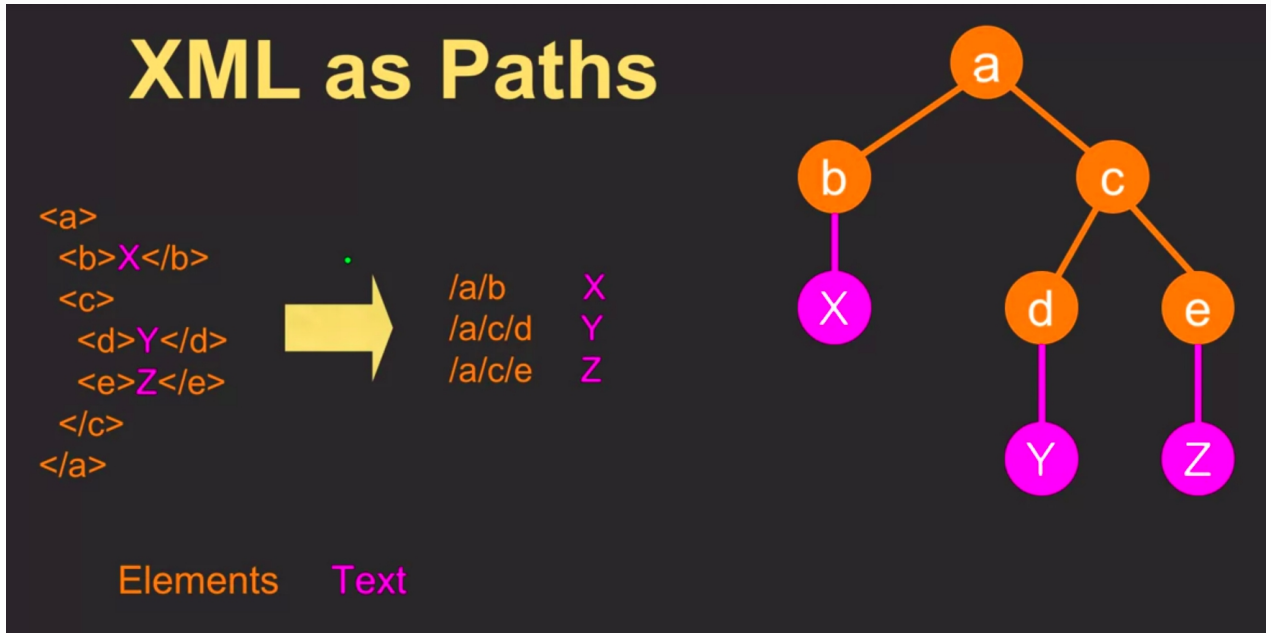
XML Terminology

- Tags indicate the beginning and ending of elements
- Attributes - Keyword/value pairs on the opening tag of XML
- Serialize / De-Serialize - Convert data in one program into a common format that can be stored and/or transmitted between systems in a programming language-independent manner

Review

Xml and Json in Python





Review

Xml and Json in Python

A key-value pair

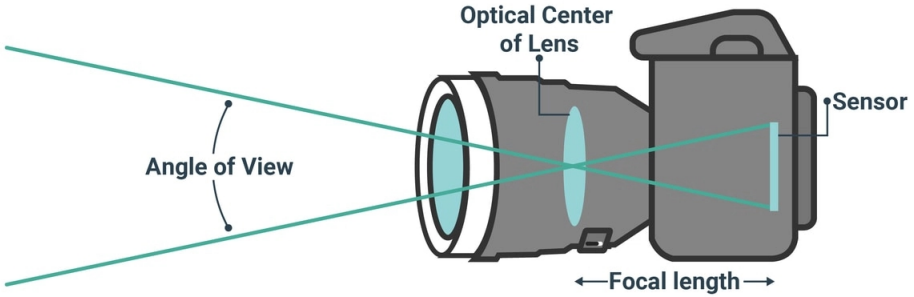
Object

Array of objects

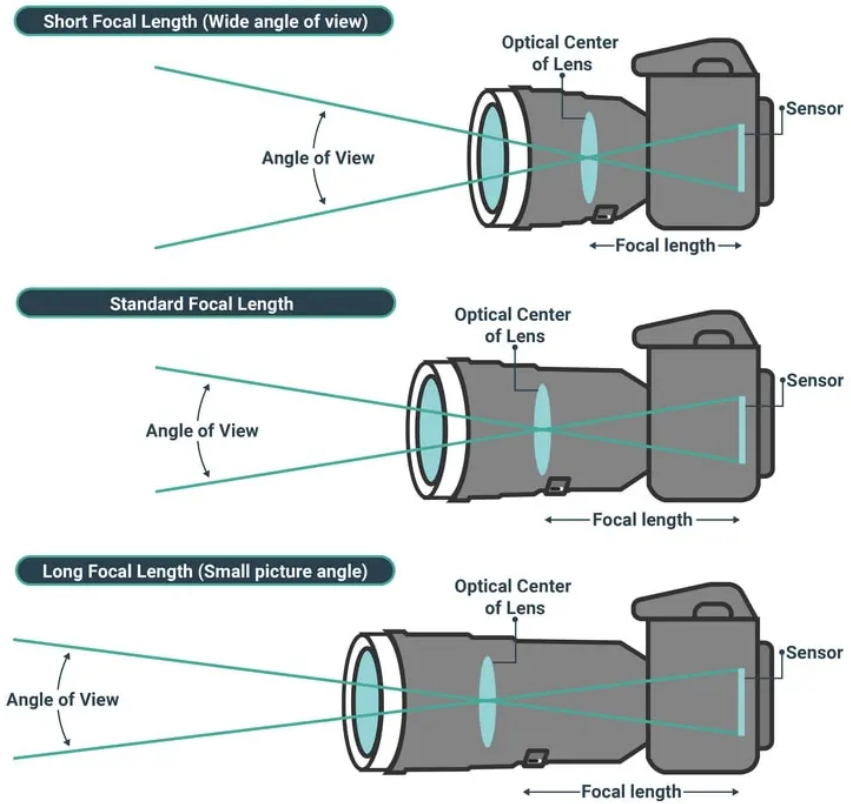
```
{
    "author": "Surbhi Kakar",
    "title": "Java Programming",
    "genre": "Computer",
    "price": "30.0",
    "publish_date": {
        "day": 1,
        "month": 8,
        "year": 2010
    },
    "publisher": "Dream Tech Press",
    "description": "..."
}
```

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        {
            "type": "home",
            "number": "212 555-1234"
        },
        {
            "type": "fax",
            "number": "646 555-4567"
        }
    ]
}
```

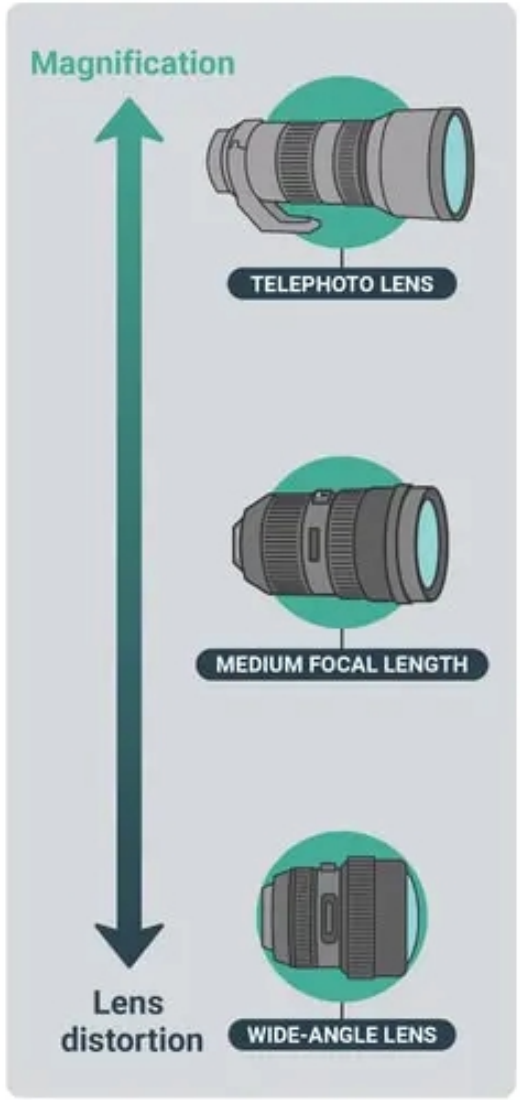
WHAT IS FOCAL LENGTH IN PHOTOGRAPHY?



WHAT IS A SHORT/LONG/STANDARD FOCAL LENGTH?



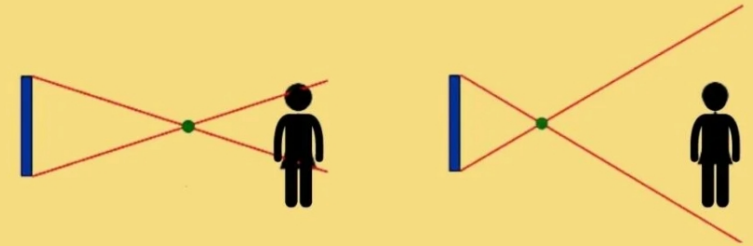
HOW DOES FOCAL LENGTH AFFECT AN IMAGE?



https://capturetheatlas.com/what-is-focal-length/

Review

Long/Short Focal Length Image



Long Focal Length

Short Focal Length

Review

Basic Image Processing with OpenCV

◆ imread()

```
Mat cv::imread ( const String &  filename,
                 int             flags = IMREAD_COLOR
               )
```

Python:

```
cv.imread( filename[, flags] ) -> retval
```

```
1  img = cv2.imread("flower.png", cv2.IMREAD_COLOR)
```

Loads an image from a file.

The function imread loads an image from the specified file and returns it. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format), the function returns an empty matrix

Note:
The function determines the type of an image by the content, not by the file extension.
In the case of color images, the decoded images will have the channels stored in **B G R order.**

Currently, the following file formats are supported:

- Windows bitmaps - *.bmp, *.dib (always supported)
- JPEG files - *.jpeg, *.jpg, *.jpe (see the *Note* section)
- JPEG 2000 files - *.jp2 (see the *Note* section)
- Portable Network Graphics - *.png (see the *Note* section)
- WebP - *.webp (see the *Note* section)
- Portable image format - *.pbm, *.pgm, *.ppm *.pxm, *.pnm (always supported)
- PFM files - *.pfm (see the *Note* section)
- Sun rasters - *.sr, *.ras (always supported)
- TIFF files - *.tiff, *.tif (see the *Note* section)
- OpenEXR Image files - *.exr (see the *Note* section)
- Radiance HDR - *.hdr, *.pic (always supported)
- Raster and Vector geospatial data supported by GDAL (see the *Note* section)

https://docs.opencv.org/4.5.5/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56

Review

Basic Image Processing with OpenCV

◆ imwrite()

```
bool cv::imwrite ( const String &          filename,
                   InputArray              img,
                   const std::vector< int > & params = std::vector< int >()
                 )
```

Python:

```
cv.imwrite( filename, img[, params] ) -> retval
```

Parameters

filename Name of the file.

img (**Mat** or vector of **Mat**) Image or Images to be saved.

Saves an image to a specified file.

The function imwrite saves the image to the specified file. The image format is chosen based on the filename extension (see cv::imread for the list of extensions). In general, only 8-bit single-channel or 3-channel (with 'BGR' channel order)

If the image format is not supported, the image will be converted to 8-bit unsigned (CV_8U) and saved that way.

https://docs.opencv.org/4.5.5/d4/da8/group__imgcodecs.html#gabbc7ef1aa2edfaa87772f1202d67e0ce

Review

Basic Image Processing with OpenCV

◆ cvtColor()

```
void cv::cvtColor ( InputArray  src,
                    OutputArray dst,
                    int         code,
                    int         dstCn = 0
                  )
```

Python:

```
cv.cvtColor( src, code[, dst[, dstCn]] ) -> dst
```

```
1  img_RGB = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2RGB)
2  img_GRAY = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2GRAY)
3  img_HSV = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2HSV)
```

Converts an image from one color space to another.

The function converts an input image from one color space to another. In case of a transformation to-from RGB color space, the order of the channels should be specified explicitly (RGB or BGR). Note that the default color format in OpenCV is often referred to as RGB but it is actually BGR (the bytes are reversed). So the first byte in a standard (24-bit) color image will be an 8-bit Blue component, the second byte will be Green, and the third byte will be Red. The fourth, fifth, and sixth bytes would then be the second pixel (Blue, then Green, then Red), and so on.

https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html#ga397ae87e1288a81d2363b61574eb8cab
https://4k8k.xyz/article/zhang_cherry/88951259

Review

Basic Image Processing with OpenCV

◆resize()

```
void cv::resize ( InputArray  src,
                  OutputArray dst,
                  Size        dsize,
                  double      fx = 0 ,
                  double      fy = 0 ,
                  int         interpolation = INTER_LINEAR
                )
```

Python:

```
cv.resize( src, dsize[, dst[, fx[, fy[, interpolation]]]] ) -> dst
```

```
7  rez_img = cv.resize(img, (width, heigh))
```

Resizes an image.

The function resize resizes the image src down to or up to the specified size. Note that the initial dst type or size are not taken into account. Instead, the size and type are derived from the src,dsize,fx, and fy. If you want to resize src so that it fits the pre-created dst, you may call the function as follows

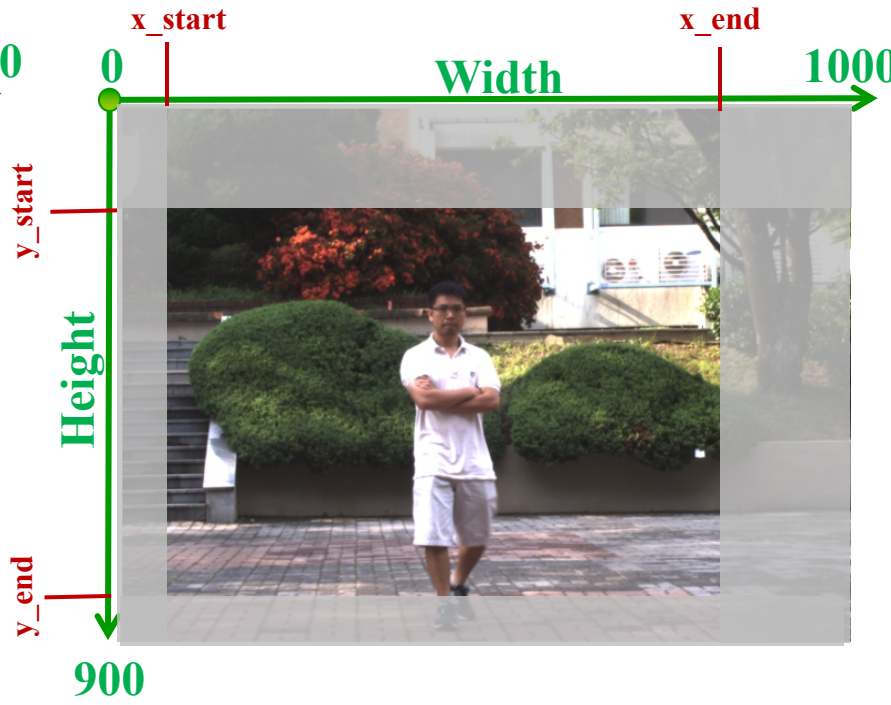
https://docs.opencv.org/4.5.5/da/d54/group__imgproc__transform.html#ga47a974309e9102f5f08231edc7e7529d

Review

Basic Image Processing with OpenCV

```
1  crop_img = img[y_start:y_end, x_start:x_end,:]
```

x_start = 100
x_end = 300
y_start = 220
y_end = 100



Review

line()

```
void cv::line ( InputOutputArray  img,
                Point             pt1,
                Point             pt2,
                const Scalar &    color,
                int               thickness = 1 ,
                int               lineType = LINE_8 ,
                int               shift = 0
              )
```

Python:

```
cv.line( img, pt1, pt2, color[, thickness[, lineType[, shift]]] ) -> img
```

Parameters

img	Image.
pt1	First point of the line segment.
pt2	Second point of the line segment.
color	Line color.
thickness	Line thickness.
lineType	Type of the line. See **LineTypes**.
shift	Number of fractional bits in the point coordinates.

Draws a line segment connecting two points.

The function line draws the line segment between pt1 and pt2 points in the image. The line is clipped by the image boundaries. For non-antialiased lines with integer coordinates, the 8-connected or 4-connected Bresenham algorithm is used. Thick lines are drawn with rounding endings. Antialiased lines are drawn using Gaussian filtering.

Review

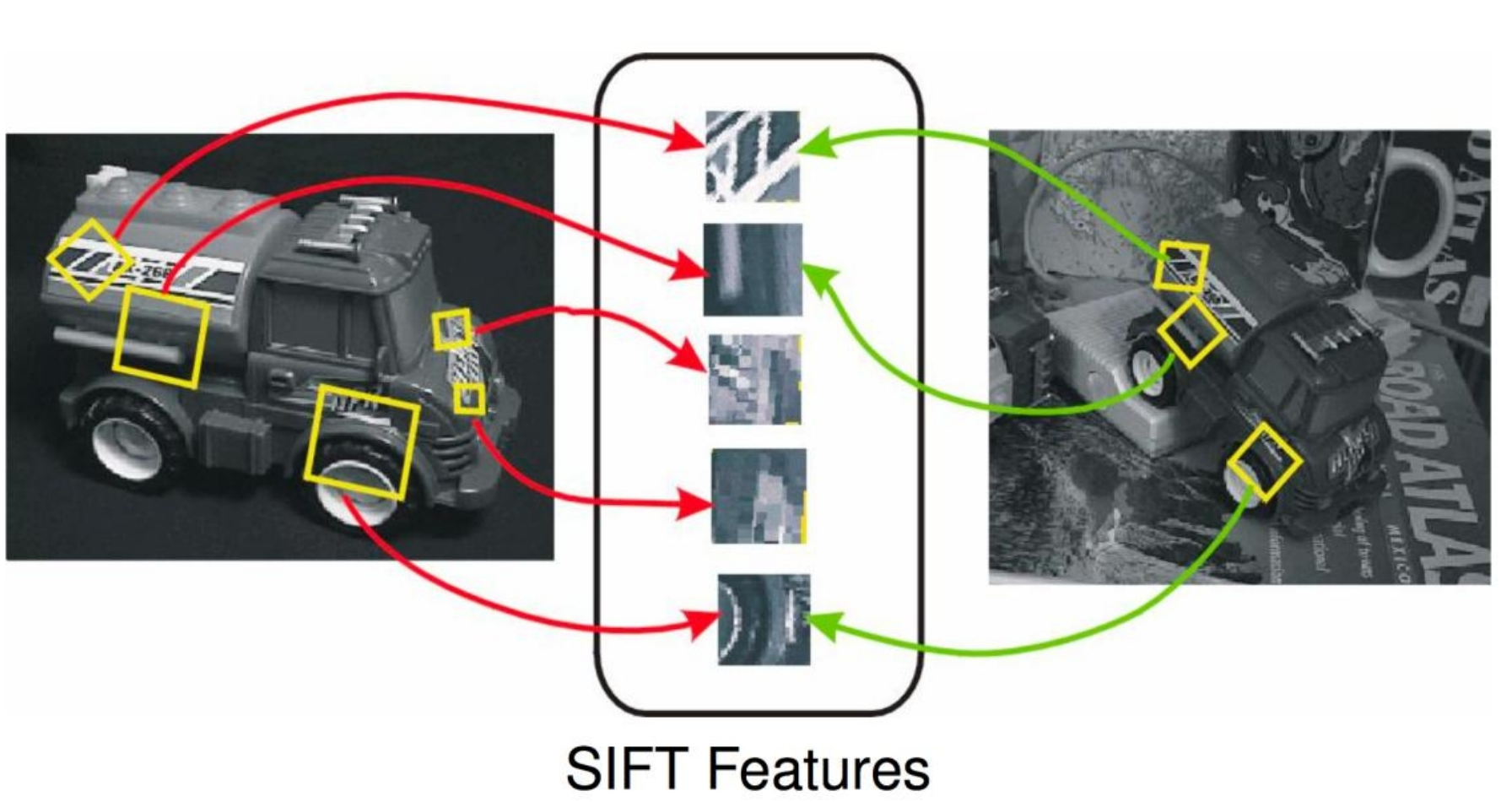
```
final_img = cv.line(final_img, pt1=(0, 100), pt2=(width, 100), color=(0,255,0), thickness=1)
final_img = cv.line(final_img, pt1=(0, 200), pt2=(width, 200), color=(0,255,0), thickness=1)
```

⋮ ⋮ ⋮

```
final_img = cv.line(final_img, pt1=(0, 900), pt2=(width, 900), color=(0,255,0), thickness=1)
```

Review

SIFT algorithm with OpenCV



https://www.cs.ubc.ca/~lsigal/425_2020W1/Lecture18.pdf

Review

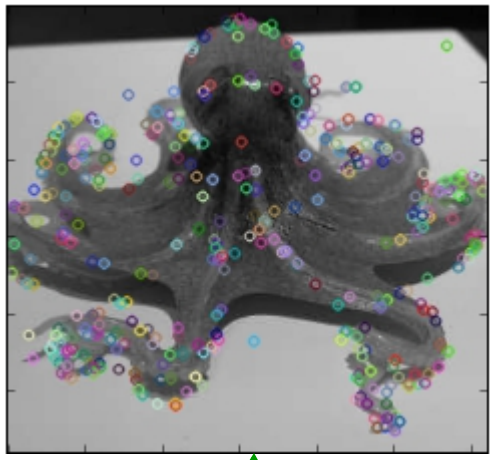
SIFT algorithm

Scale-space Extrema Detection

Keypoint Localization

Orientation Assignment

Keypoint Descriptor

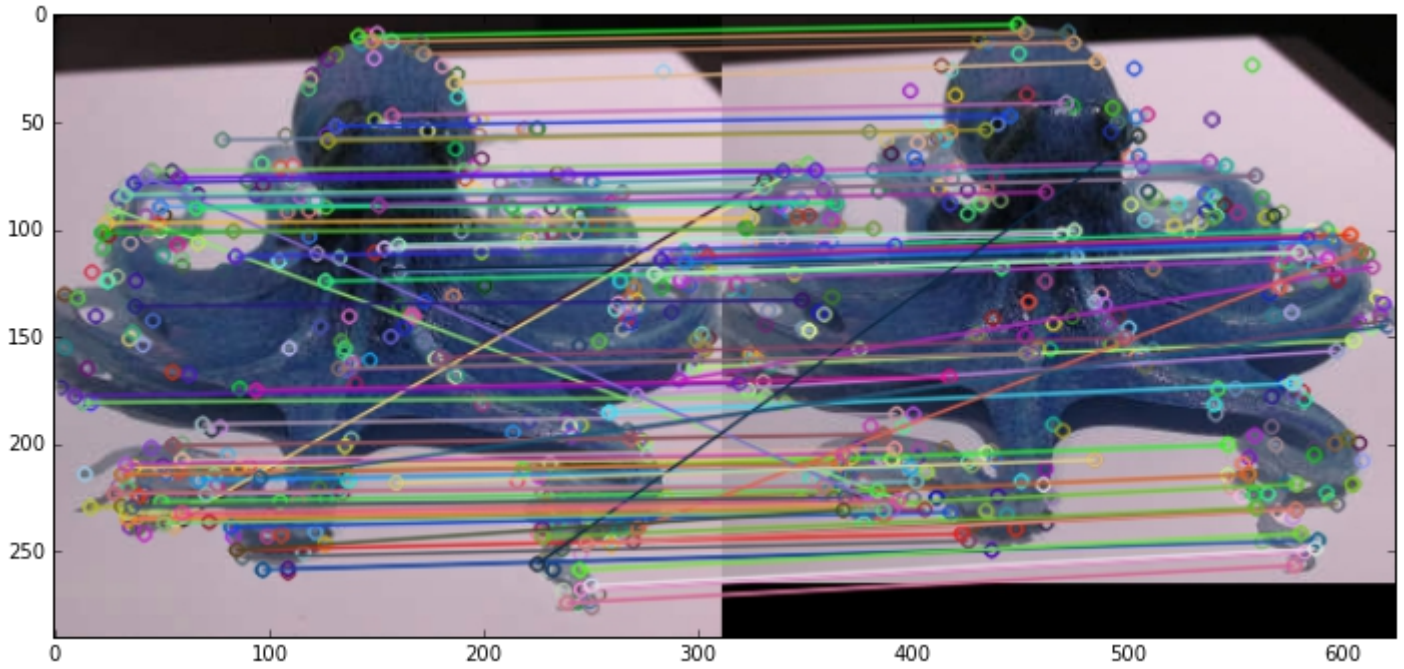


https://ianlondon.github.io/blog/how-to-sift-opencv/

Review

SIFT algorithm

Keypoint Matching



https://ianlondon.github.io/blog/how-to-sift-opencv/

Content

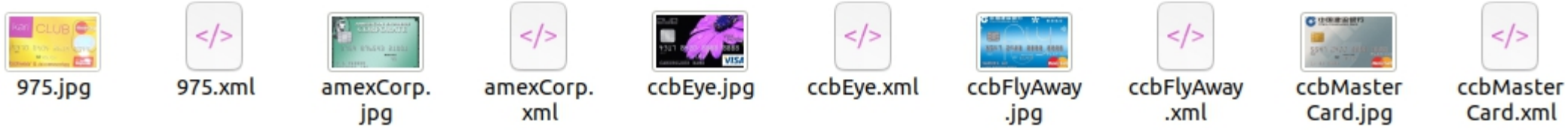
- **Exercise1: PASCAL VOC and CreateML Annotation Format**
  - Extract information from PASCAL VOC and CreatML annotation files for the Object Detection task
- **Exercise2: Google Books API**
  - Search, extract, and save necessary book information with the Google Books API
- **Exercise3: Compensation for Different Focal Length Images**
  - Compensate the short focal length image to achieve approximately the same result as the long focal length image

Exercise1: PASCAL VOC and CreateML Annotation Format

Viết các hàm đọc và lấy thông tin từ tất cả các file xml hoặc json trong folder theo yêu cầu sau: (một số configuration và data đã được định nghĩa trước và đính kèm vào link file notebook bên dưới)

(a) Viết hàm readxml(path) đọc toàn bộ file xml và lấy các thông tin sau "filename", và thông tin của các "object" bao gồm "name", "xmin", "ymin", "xmax", "ymax" và trả về một dictionary với mỗi item gồm key là tên đường dẫn ảnh, value là một list chứa thông tin của các "object"

- **Input**: path đường dẫn file chứa ảnh (.jpg) và file xml (.xml).
- **Output**: một dictionary
  - key: đường dẫn đến file ảnh
  - value: list với mỗi element là 'name' và tuple(xmin, ymin, xmax, ymax)



Data khi download về 1 (a)

Exercise1: PASCAL VOC and CreateML Annotation Format

```
[ ]  1 !gdown --id 1B6YL5voMkc9YqPp9wKx06KOTuGRiuXQM
```

```
[ ]  1 !gdown --id 1eBdw81oNtef9vVnDnowwFFZtFBHMGXSE
```

```
[ ]  1 !unzip /content/data_P1.zip
```

Chuẩn bị data

```
[ ]  1 import os
     2 import json
     3 import xml.etree.ElementTree as ET
     4 from P1 import draw_one_img
```

```
[ ]  1 path1a = "/content/data_P1/a"
     2 path1b = "/content/data_P1/b"
```

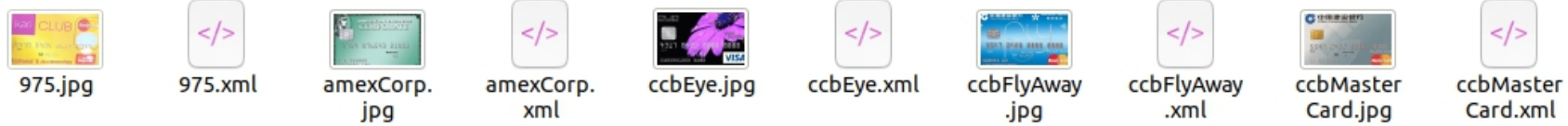
Import module/library định nghĩa đường dẫn cho câu (a), (b)

Exercise1: PASCAL VOC and CreateML Annotation Format

```xml
<annotation>
    <folder>d</folder>
    <filename>975.jpg</filename>
    <path>/home/khoa/Khoa/AI2022/Week2/Data/
        Credit-debit card numbers detection_xml/
        d/975.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>828</width>
        <height>522</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>5</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>59</xmin>
            <ymin>277</ymin>
            <xmax>98</xmax>
            <ymax>329</ymax>
        </bndbox>
    </object>
    <object>
        <name>3</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>102</xmin>
            <ymin>280</ymin>
            <xmax>131</xmax>
```

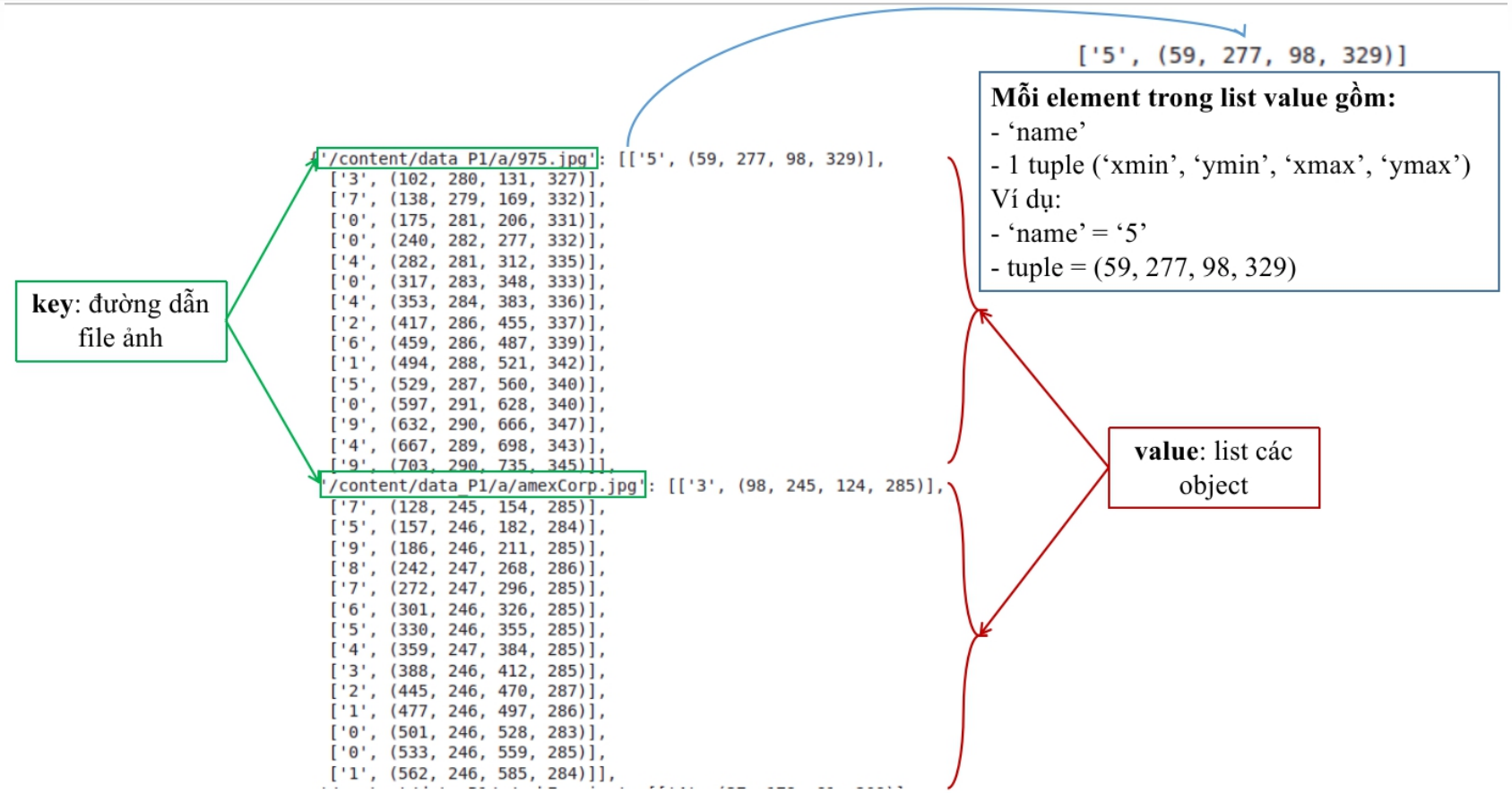
```xml
<annotation>
    <folder>d</folder>
    <filename>975.jpg</filename>
    <path>/home/khoa/Khoa/AI2022/Week2/Data/
        Credit-debit card numbers detection_xml/
        d/975.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>828</width>
        <height>522</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>5</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>59</xmin>
            <ymin>277</ymin>
            <xmax>98</xmax>
            <ymax>329</ymax>
        </bndbox>
    </object>
    <object>
        <name>3</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>102</xmin>
            <ymin>280</ymin>
            <xmax>131</xmax>
```

Exercise1: PASCAL VOC and CreateML Annotation Format




```
[ ]   1 path1a = "/content/data_P1/a"
      2 path1b = "/content/data_P1/b"
```

Data sau khi download 1(a)




```
<annotation>
    <folder>d</folder>
    <filename>975.jpg</filename>
    <path>/home/khoa/Khoa/AI2022/Week2/Data/
        Credit-debit card numbers detection_xml/
        d/975.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>828</width>
        <height>522</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>5</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>59</xmin>
            <ymin>277</ymin>
            <xmax>98</xmax>
            <ymax>329</ymax>
        </bndbox>
    </object>
    <object>
        <name>3</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>102</xmin>
            <ymin>280</ymin>
            <xmax>131</xmax>
```

Output 1(a)

Exercise1: PASCAL VOC and CreateML Annotation Format

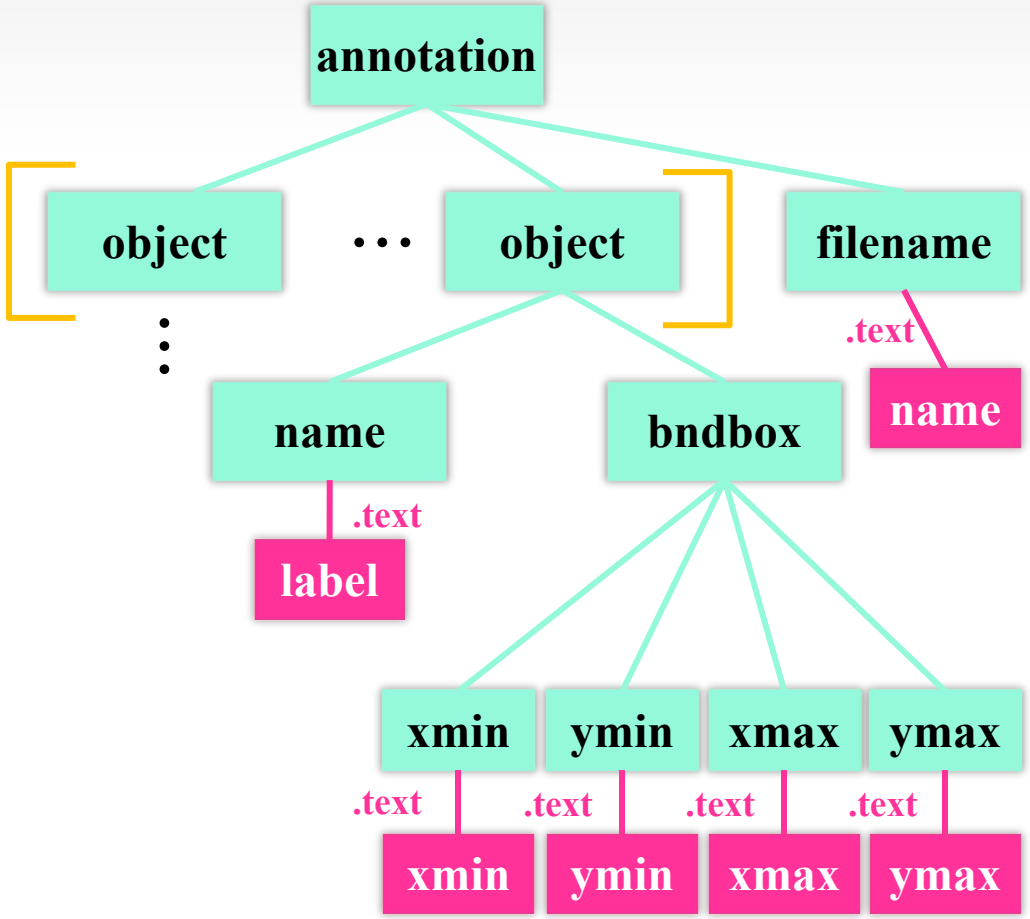
```python
# Examples 1a
res = readxml(path1a)
it = iter(res.items())

img_path, data = next(it)
draw_one_img(img_path=img_path,
             data=data,
             is_from_xml=True)
...
# Only 5 times
img_path, data = next(it)
```

```python
draw_one_img(img_path=img_path,
             data=data,
             is_from_xml=True)

```

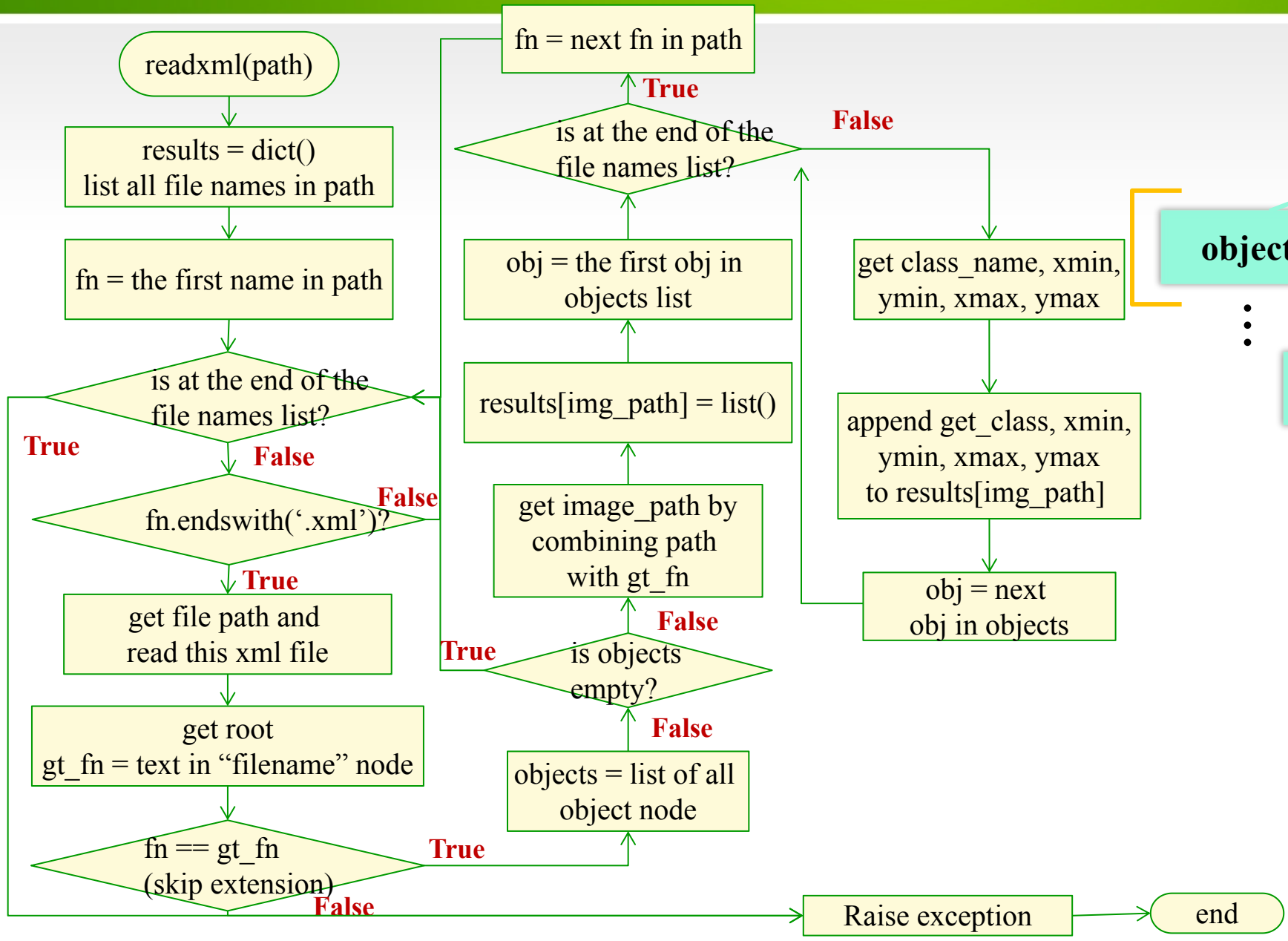
Hình 4: Kết quả 1a

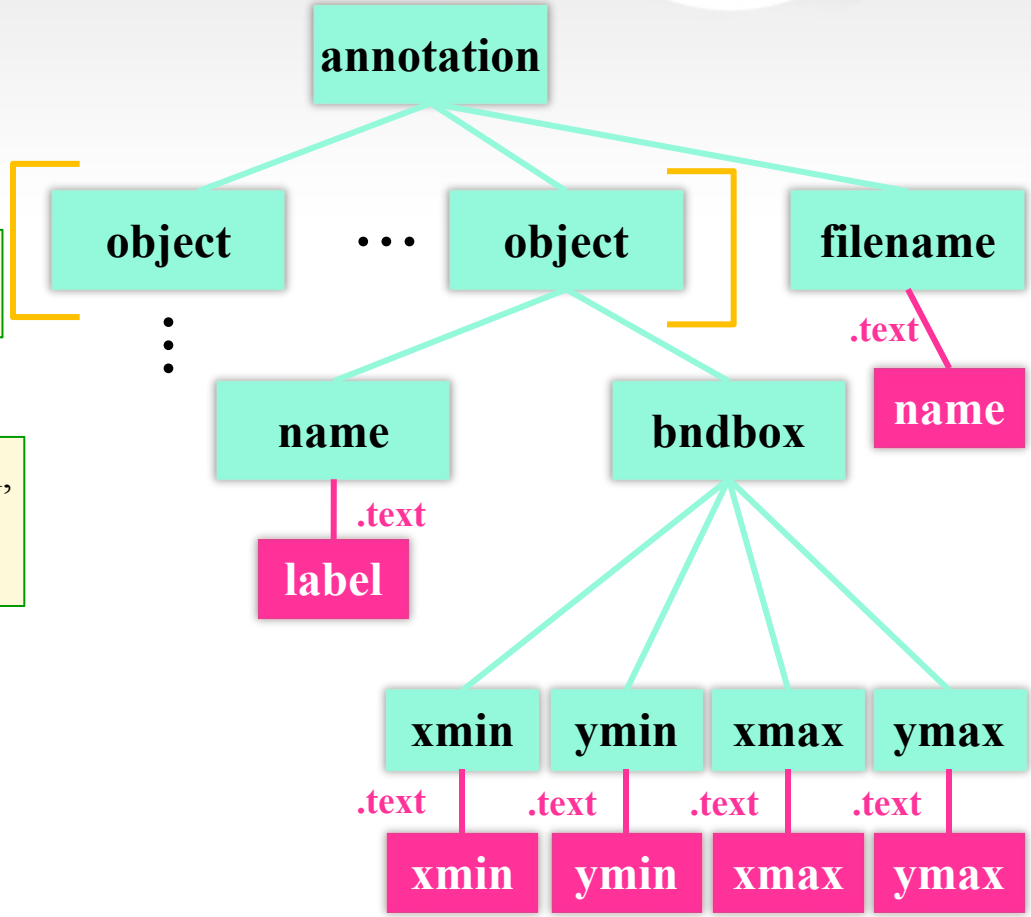
Exercise1: PASCAL VOC and CreateML Annotation Format

```xml
<annotation>
    <folder>d</folder>
    <filename>975.jpg</filename>
    <path>/home/khoa/Khoa/AI2022/Week2/Data/
        Credit-debit card numbers detection_xml/
        d/975.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>828</width>
        <height>522</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>5</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>59</xmin>
            <ymin>277</ymin>
            <xmax>98</xmax>
            <ymax>329</ymax>
        </bndbox>
    </object>
    <object>
        <name>3</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>102</xmin>
            <ymin>280</ymin>
            <xmax>131</xmax>
```



Exercise1: PASCAL VOC and CreateML Annotation Format





Exercise1: PASCAL VOC and CreateML Annotation Format

Viết các hàm đọc và lấy thông tin từ tất cả các file xml hoặc json trong folder theo yêu cầu sau: (một số configuration và data đã được định nghĩa trước và đính kèm vào link file notebook bên dưới)

(b) Viết hàm readjon(path) đọc toàn bộ file json và lấy các thông tin sau "imagePath", và thông tin của "shapes" bao gồm "label", và "points". Sau đó trả về một dictionary với mỗi item gồm key là tên đường dẫn ảnh, value là một list chứa thông tin của các "shapes"

Input: Đường dẫn file chứa ảnh (.jpg) và file json (.json).

Output: một dictionary
– key: path đường dẫn đến file ảnh
– value: list với mỗi element là 'label' và list points

Data khi download về 1 (b)

Exercise1: PASCAL VOC and CreateML Annotation Format


```
{
  "version": "3.16.7",
  "flags": {},
  "shapes": [
    {
      "label": "T",
      "line_color": null,
      "fill_color": null,
      "points": [
        [
          256.4102564102564,
          160.47008547008548
        ],
        [
          281.62393162393164,
          145.51282051282053
        ],
        [
          296.5811965811966,
          173.0769230769231
        ],
        [
          269.6581196581197,
          186.75213675213675
        ]
      ],
      "shape_type": "polygon",
      "flags": {}
    },
    {
      "label": "H",
      "line_color": null,
      "fill_color": null,
      "points": [
```

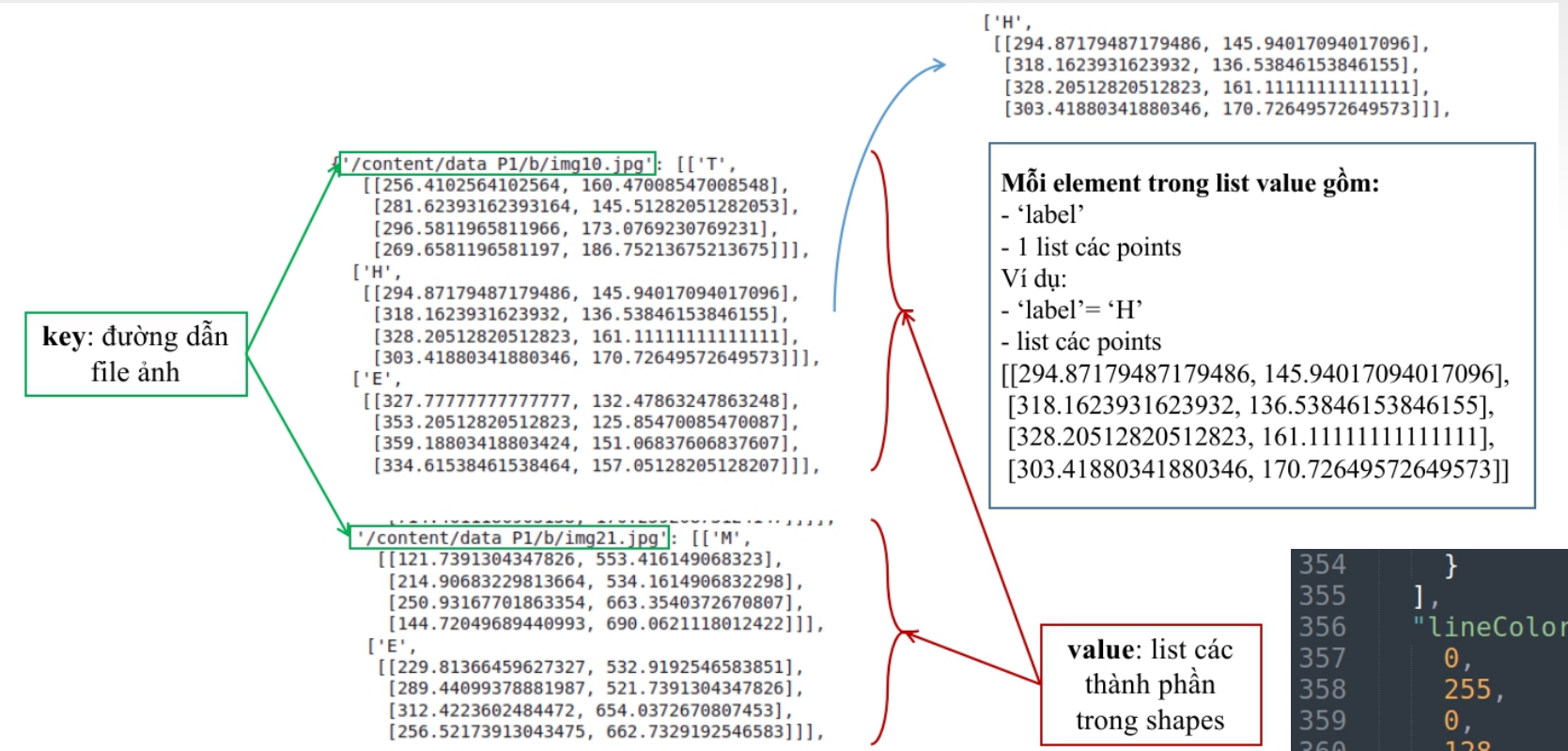
```
    }
  ],
  "lineColor": [
    0,
    255,
    0,
    128
  ],
  "fillColor": [
    255,
    0,
    0,
    128
  ],
  "imagePath": "img10.jpg",
  "imageHeight": 768,
  "imageWidth": 1024
}
```

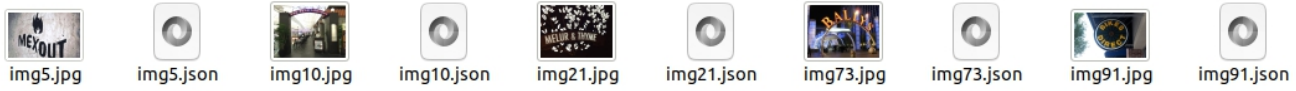


```
{
  "version": "3.16.7",
  "flags": {},
  "shapes": [
    {
      "label": "T",
      "line_color": null,
      "fill_color": null,
      "points": [
        [
          256.4102564102564,
          160.47008547008548
        ],
        [
          281.62393162393164,
          145.51282051282053
        ],
        [
          296.5811965811966,
          173.0769230769231
        ],
        [
          269.6581196581197,
          186.75213675213675
        ]
      ],
      "shape_type": "polygon",
      "flags": {}
    },
    {
      "label": "H",
      "line_color": null,
      "fill_color": null,
      "points": [
```

```
    }
  ],
  "lineColor": [
    0,
    255,
    0,
    128
  ],
  "fillColor": [
    255,
    0,
    0,
    128
  ],
  "imagePath": "img10.jpg",
  "imageHeight": 768,
  "imageWidth": 1024
}
```

Exercise1: PASCAL VOC and CreateML Annotation Format





Data khi download về 1 (b)

```
[ ]   1 path1a = "/content/data_P1/a"
      2 path1b = "/content/data_P1/b"
```

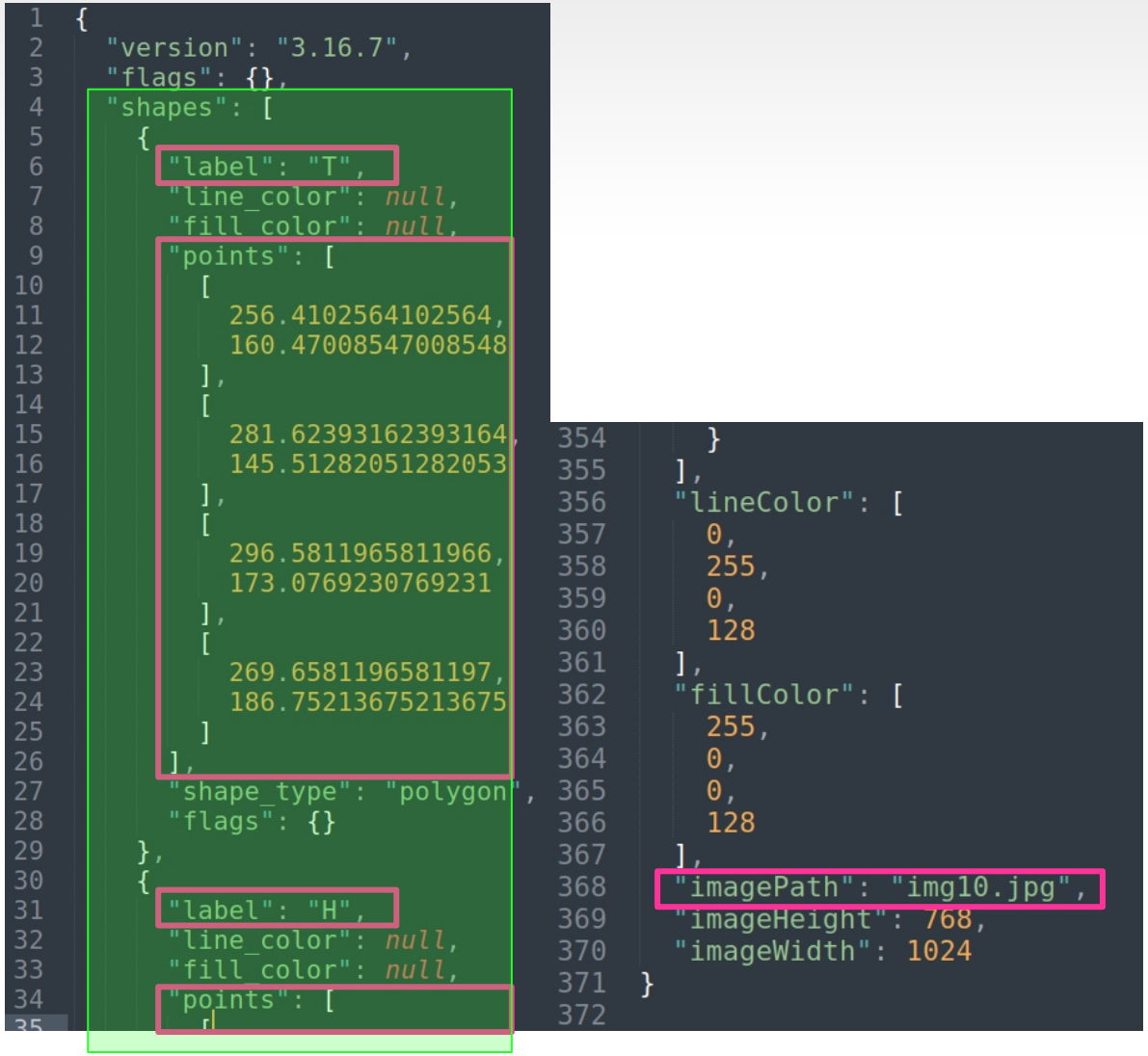

```
{
  "version": "3.16.7",
  "flags": {},
  "shapes": [
    {
      "label": "T",
      "line_color": null,
      "fill_color": null,
      "points": [
        [
          256.4102564102564,
          160.47008547008548
        ],
        [
          281.62393162393164,
          145.51282051282053
        ],
        [
          296.5811965811966,
          173.0769230769231
        ],
        [
          269.6581196581197,
          186.75213675213675
        ]
      ],
      "shape_type": "polygon",
      "flags": {}
    },
    {
      "label": "H",
      "line_color": null,
      "fill_color": null,
      "points": [
        [
      }
    ],
    "lineColor": [
      0,
      255,
      0,
      128
    ],
    "fillColor": [
      255,
      0,
      0,
      128
    ],
    "imagePath": "img10.jpg",
    "imageHeight": 768,
    "imageWidth": 1024
}
```

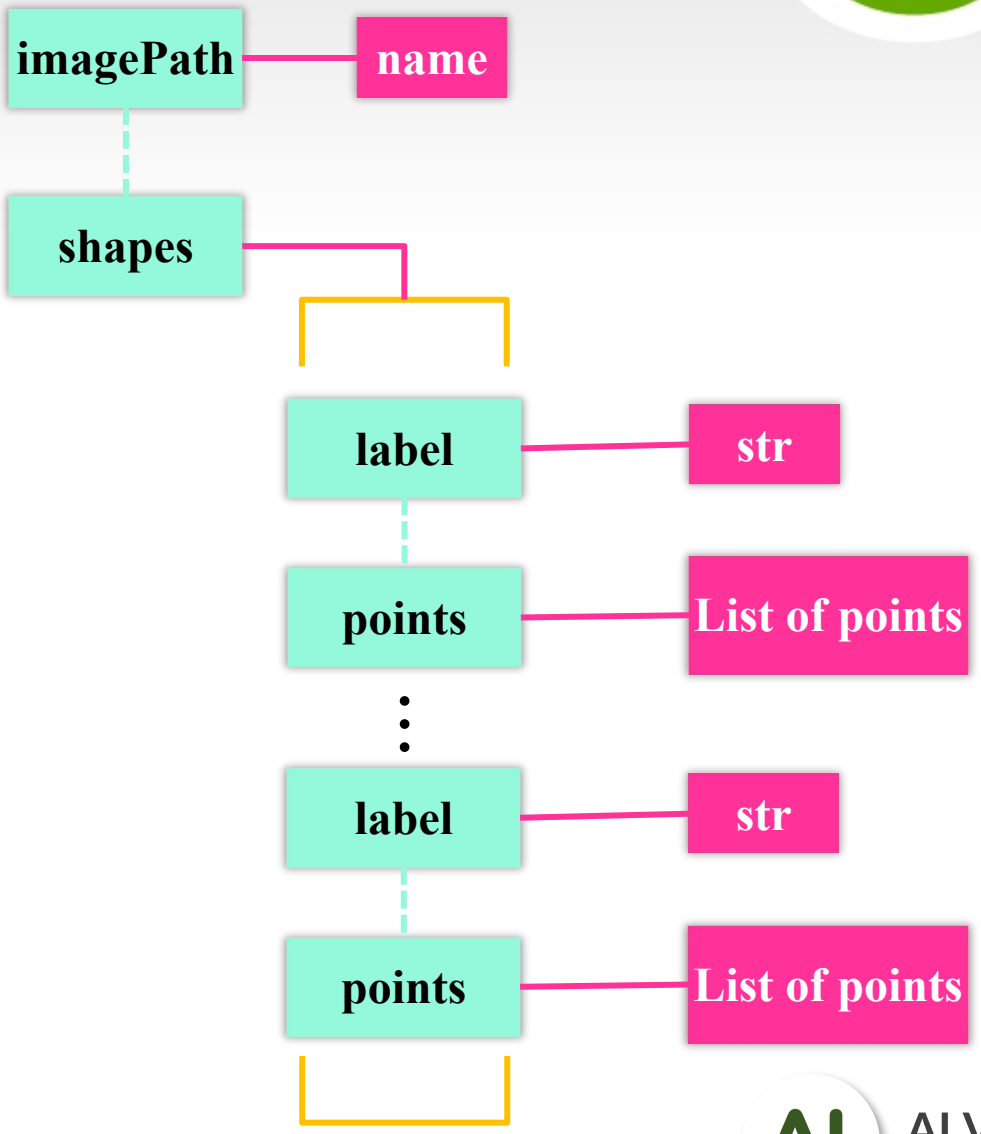
Exercise1: PASCAL VOC and CreateML Annotation Format

```
# Examples 1b
res2 = readjson(path1b)
it2 = iter(res2.items())

img_path, data = next(it2)
draw_one_img(img_path=img_path,
             data=data,
             is_from_xml=False)

...
# Only 5 times
img_path, data = next(it2)
draw_one_img(img_path=img_path,
             data=data,
             is_from_xml=False)
```

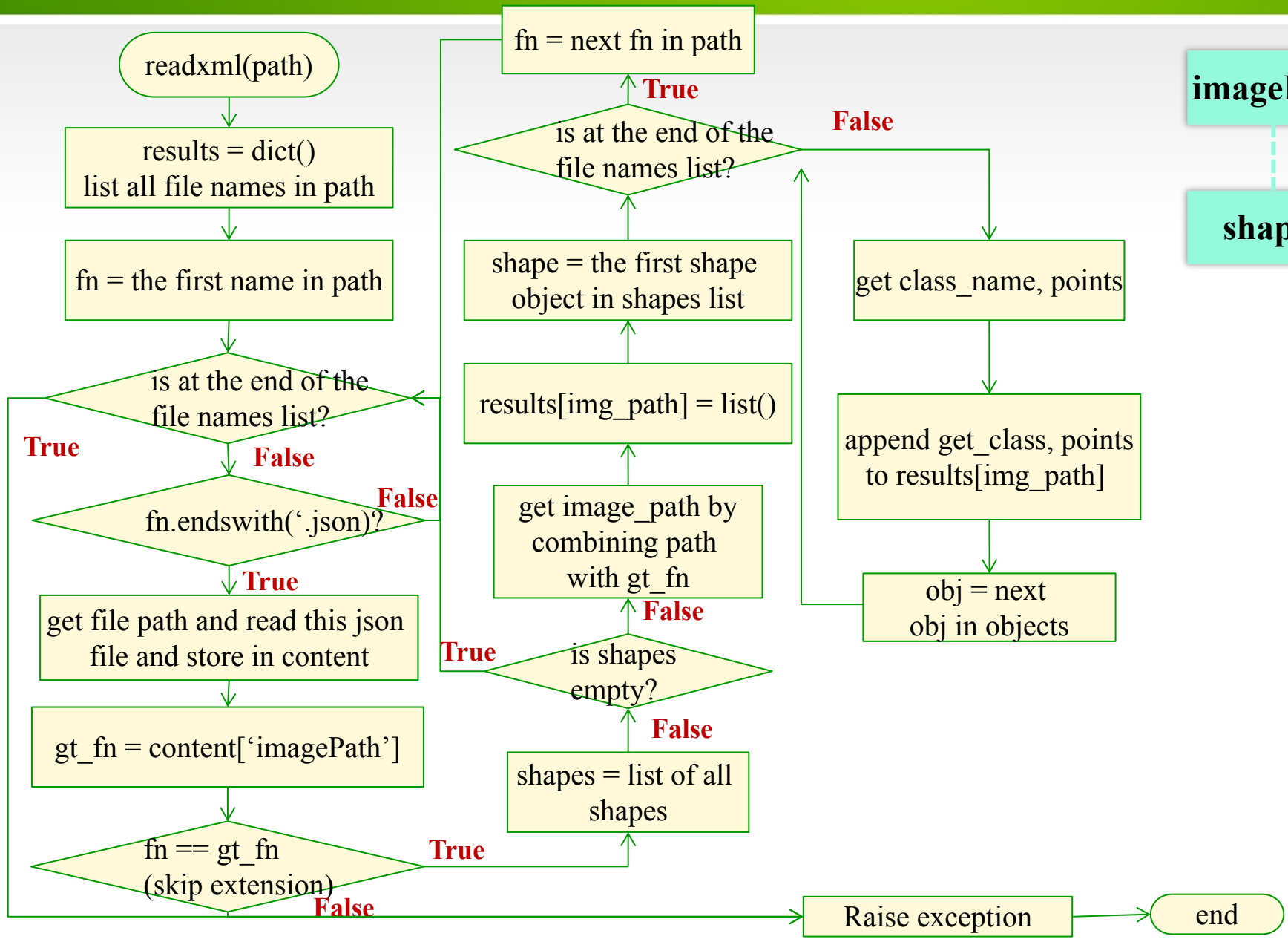
Hình 5: Kết quả 1b

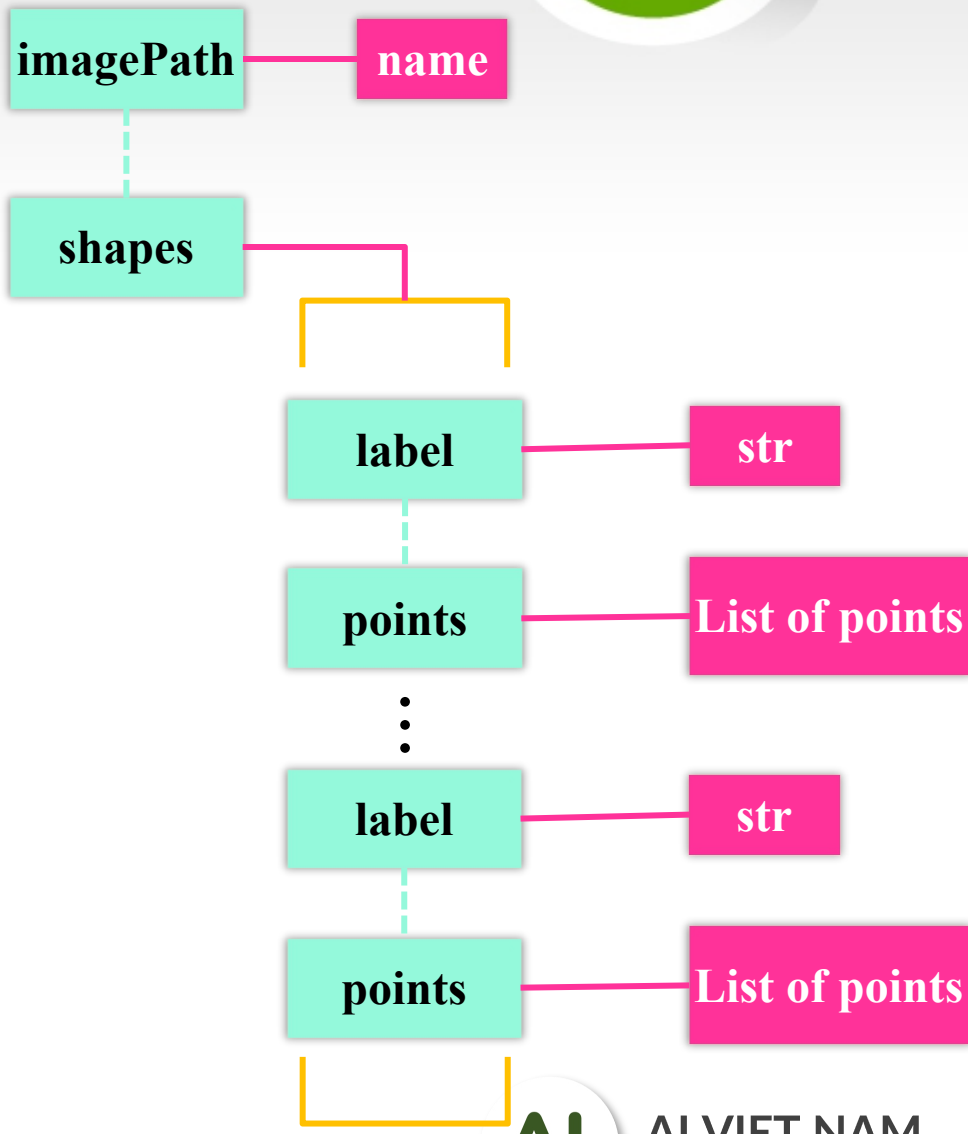
Exercise1: PASCAL VOC and CreateML Annotation Format





Exercise1: PASCAL VOC and CreateML Annotation Format





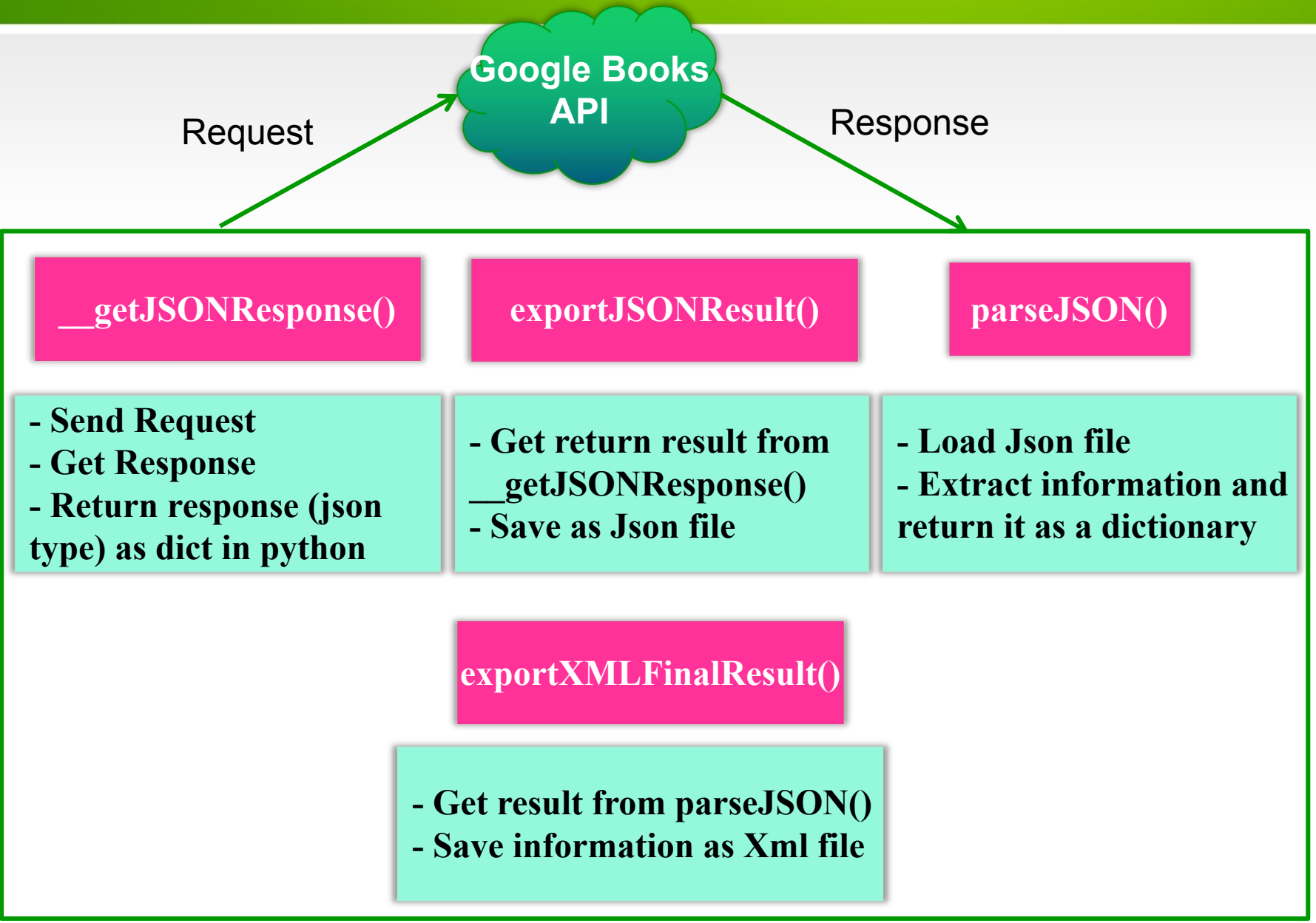
Content

- **Exercise1: PASCAL VOC and CreateML Annotation Format**
  - Extract information from PASCAL VOC and CreatML annotation files for the Object Detection task
- **Exercise2: Google Books API**
  - Search, extract, and save necessary book information with the Google Books API
- **Exercise3: Compensation for Different Focal Length Images**
  - Compensate the short focal length image to achieve approximately the same result as the long focal length image

Exercise2: Google Books API

- Yêu Cầu Của Đề Bài:
  - Hoàn thiện **SearchingGoogleBook** class để thực hiện việc tìm kiếm sách với google và lưu thông tin kết quả dưới dạng file xml
  - Luồng hoạt động
    - Request tên sách lên googleapi để thực hiện kiếm sách
    - Lưu response (kết quả tìm được) dạng file json (**exportJSONResult**)
    - Load file json kết quả, trích xuất các thông tin cần lấy và trả về kết quả dưới dạng dictionary (**parseJSON**)
    - Dùng dictionary tạo file xml để lưu kết quả thông tin (**exportXMLFinalResult**)

Exercise2: Google Books API



SearchingGoogleBook

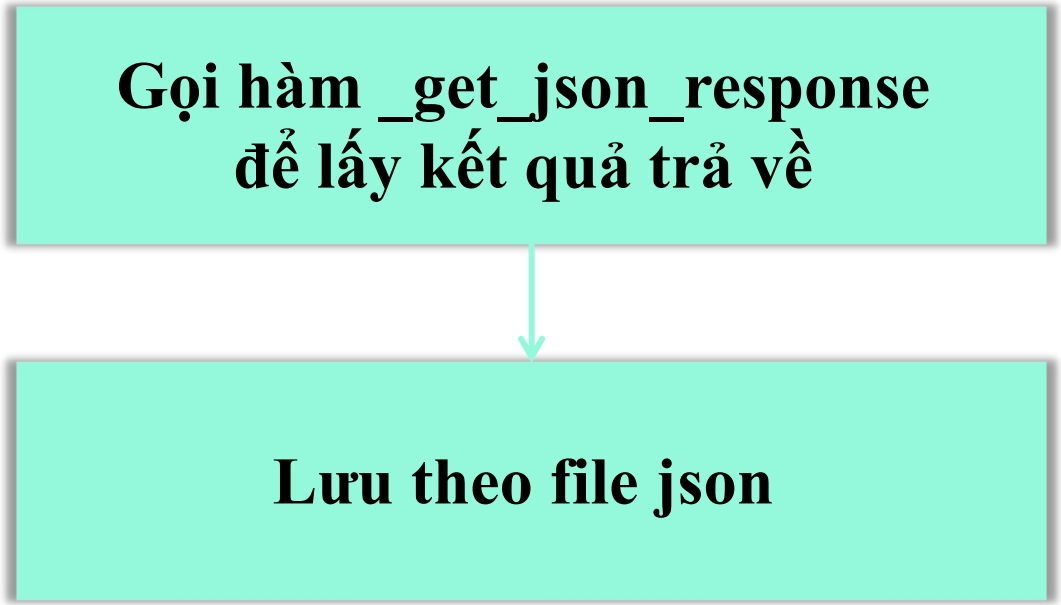
- endpoint: string
- maxRes: int
- query: str

+ __init__(maxRes)
+ __getJSONResponse(): void
+ exportJSONResult(): void
+ parseJSON(): dict()
+ exportXMLFinalResult(): void

Exercise2: Google Books API

(a) Hoành thiện method **exportJSONResult** để có thể xuất ra được file json kết quả sau khi search.

- **Input**: query=tên sách, json_path=đường dẫn lưu file json kết quả.
- **Ouput**: file json kết quả tìm kiếm cuốn sách được lưu theo đường dẫn.



Exercise2: Google Books API

- Python yêu cầu json nên dùng **utf-8**, utf-16 hoặc utf-32
- ensure_ascii=False, thể hiện chữ cái thay vì dùng \u escape sequence
- indent=4 để làm kết quả hiển thị đẹp hơn


```
"kind": "books#volume",
"id": "RaQwtwAACAAJ",
"etag": "gBXJxYe2770",
"selfLink": "https://www.googleapis.com/books/v1/volumes/RaQwtwAACAAJ",
"volumeInfo": {
    "title": "T\u00f4i t\u00e0i gi\u1ecfi, b\u1ea1n c\u0169ng th\u1ebf",
    "authors": [
        "Adam Khoo",
        "Tony Buzan",
```

```
"kind": "books#volume",
"id": "RaQwtwAACAAJ",
"etag": "6e0oQYVCjgE",
"selfLink": "https://www.googleapis.com/books/v1/volumes/RaQwtwAACAAJ",
"volumeInfo": {
    "title": "Tôi tài giỏi, bạn cũng thế",
    "authors": [
        "Adam Khoo",
        "Tony Buzan",
```


T\u00f4i

Decoded

Unicode Escape	Tôi

https://pynative.com/python-json-encode-unicode-and-non-ascii-characters-as-is/

Exercise2: Google Books API

(b) Quan sát file json đã được save, để viết method **parseJSON** lấy các thông tin như sau:

- totalItems: số lượng search được từ query
- Trong một list các sách trả về, chỉ cần lấy 3 cuốn sách đầu tiên gồm các thông tin sau (GÍA TRỊ NÀO KHÔNG KIẾM ĐƯỢC THÌ TRẢ VỀ STRING 'Unknow'):
  - title
  - authors
  - publisher
  - publishedDate
  - description
  - buyLink
  - textToSpeechPermission
  - pdf
- Thêm 1 cặp key-value, với key là "query" và value là tên cuốn sách ban tìm kiếm vào dictionary output
- **Input**: là đường dẫn fie json được ghi xuống từ function của câu 2a ở trên
- **Ouput**: là dictionary với các thông tin muốn lấy
- Tham khảo file TTGBCT_2.json và aaa_2.json

Exercise2: Google Books API

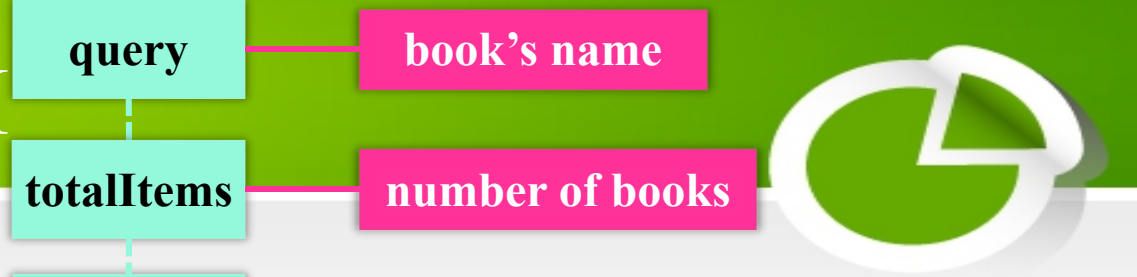
- **parseJSON**(json_path):
  - **Input**: đường dẫn lưu file
  - **Output**: dictionary chứa các thông tin cần thiết
  - Đọc đề bài kết quả trả về cần 3 thứ:
    - query: tên cuốn sách cần google search
    - totalItems: số lượng serach được từ tên sách (query)
    - booksInfo: list gồm 3 cuốn sách đầu tiên và mỗi cuốn sẽ có 1 số thông tin như yêu cầu đề bài
  - Quan sát file json từ hàm **exportJSONResult** để tìm cách lấy thông tin

```
{
    "query": "Tôi tài giỏi bạn cũng thế",
    "totalItems": 466,
    "booksInfo": [
        {
            "title": "I Am Gifted, So Are You",
            "authors": [
                "Adam Khoo"
            ],
            "publisher": "Marshall Cavendish International Asia Pte Ltd",
            "publishedDate": "2014-07-15",
            "description": "Every student can achieve and excel if given the
            "buyLink": "https://play.google.com/store/books/details?id=WIZ1B
            "textToSpeechPermission": "ALLOWED",
            "pdf": true
        },
        {
            "title": "Tôi tài giỏi, bạn cũng thế!",
            "authors": [
                "Adam Khoo"
            ],
            "publisher": "Unknow",
            "publishedDate": "2018",
            "description": "Unknow",
            "buyLink": "Unknow",
            "textToSpeechPermission": "ALLOWED",
            "pdf": false
        },
        {
            "title": "I Am Gifted, So are You!",
            "authors": [
                "Adam Khoo",
                "Tony Buzan",
                "Ernest Wong"
            ],
            "publisher": "Unknow",
            "publishedDate": "2002",
            "description": "Unknow",
            "buyLink": "Unknow",
            "textToSpeechPermission": "ALLOWED",
            "pdf": false
        }
    ]
}
```

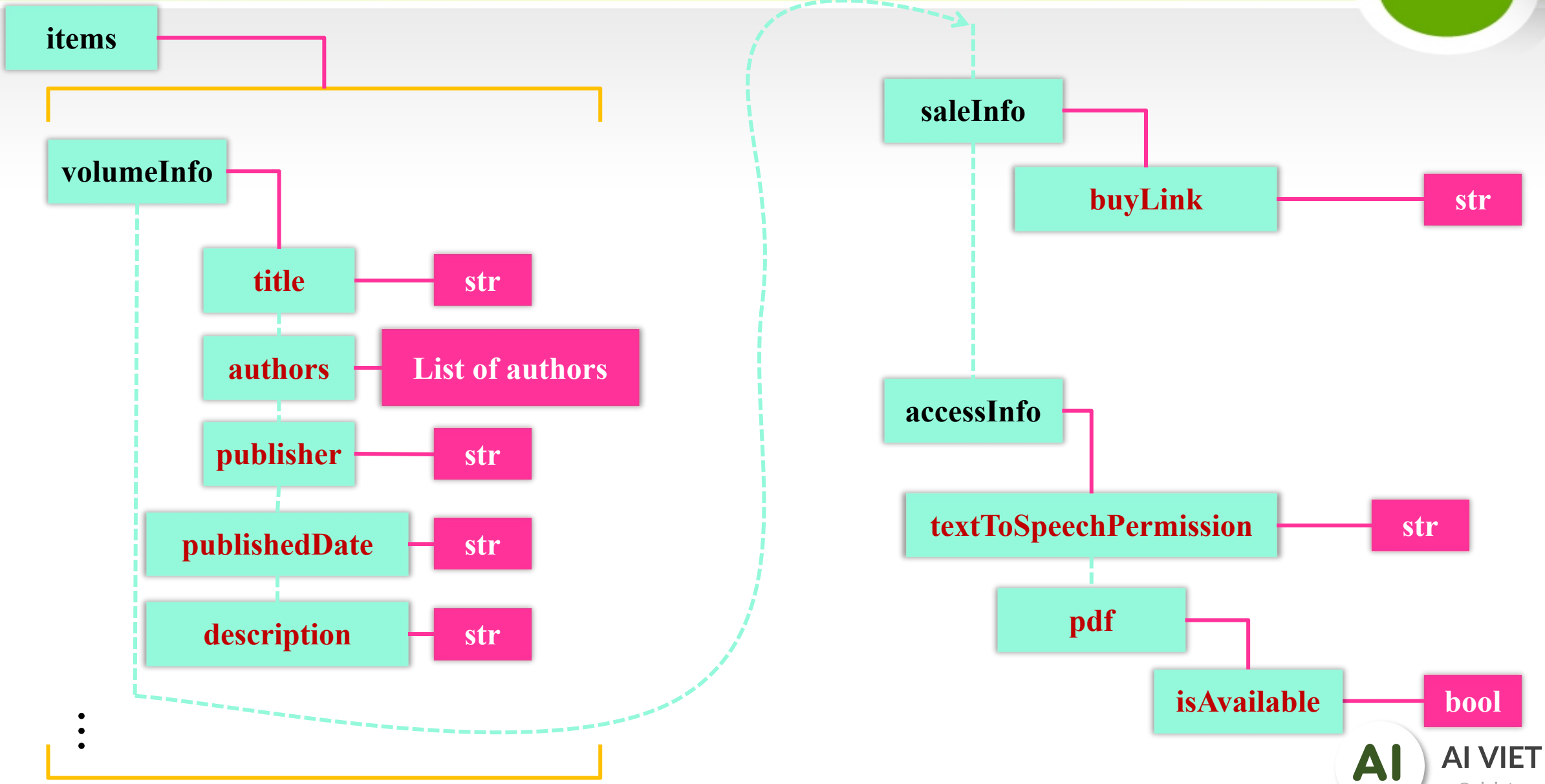


```
{
    "query": "aaaaaaaaaaasssssssssss
    "totalItems": 0,
    "booksInfo": []
}
```


ks API



Exercise2: Google Books API



Exercise2: Google Books API

(c) Viết method **exportXMLFinalResult** và Dùng kết quả từ **parseJSON** tạo thành 1 file xml tương ứng.

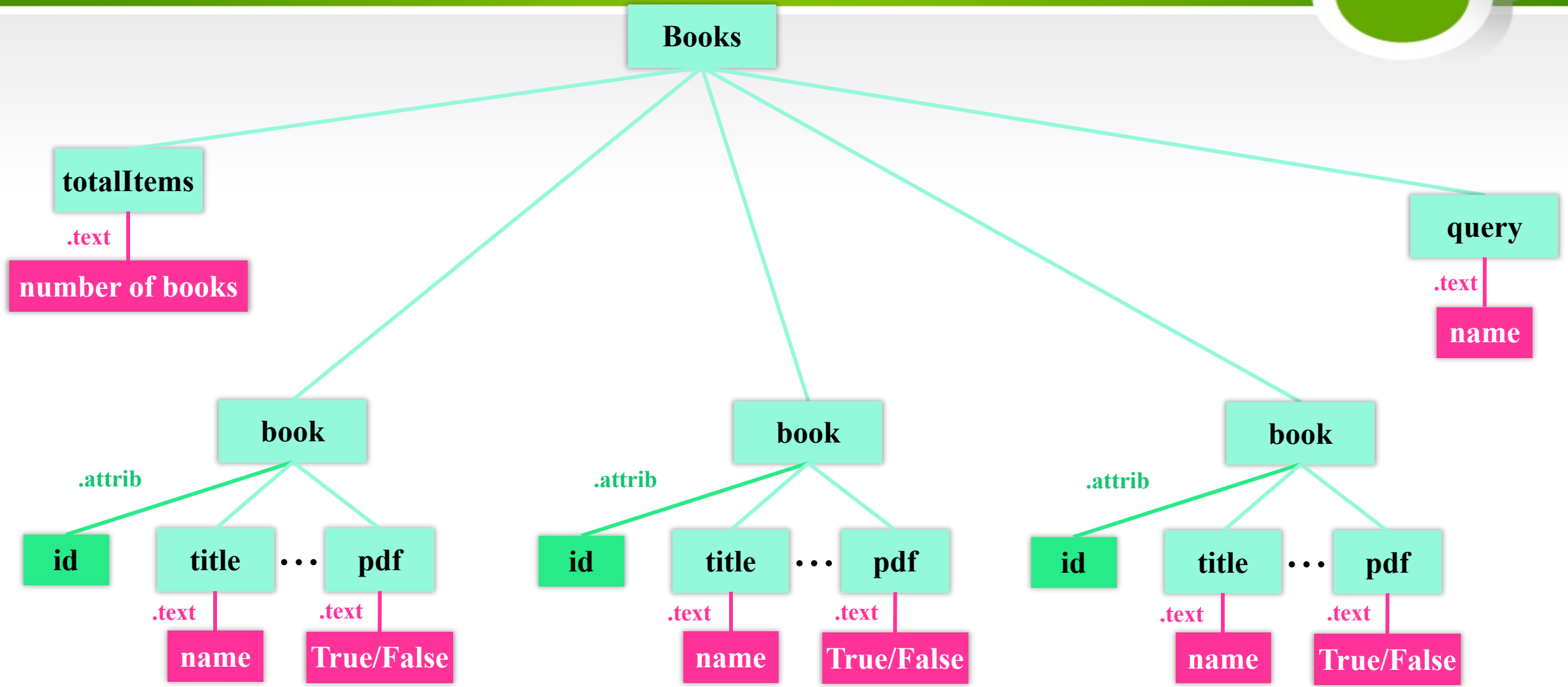
- Nếu kết quả trả về có key 'booksInfo' là list rỗng (trường hợp không tìm thấy tên sách) thì không cần tạo element này
- **Input**: json_path là đường dẫn fie json được ghi xuống từ method của câu 2a ở trên, và xml_path là đường dẫn ghi file xml
- **Output**: là file xml có format như TTGBCT_3.xml và aaa_3.xml
- **Lưu ý**: đối với 'booksInfo' là list chứa thông tin từng cuốn sách lấy được, thì mỗi element sẽ tạo ra element <book> </book> trong xml với attribute id= 0, 1, 2 theo thứ tự trong list của dicitonary. Nếu 'booksInfo' là list rỗng thì bỏ qua không cần làm gì hết.
- **Lưu ý**: authors là list nên trong xml sẽ join lại tên các tác giả và cách nhau bằng dấu ','

Exercise2: Google Books API

```json
{
    "query": "Tôi tài giỏi bạn cũng thê",
    "totalItems": 466,
    "booksInfo": [
        {
            "title": "I Am Gifted, So Are You",
            "authors": [
                "Adam Khoo"
            ],
            "publisher": "Marshall Cavendish International Asia Pte Ltd",
            "publishedDate": "2014-07-15",
            "description": "Every student can achieve and excel if given the
            "buyLink": "https://play.google.com/store/books/details?id=WIZ1B
            "textToSpeechPermission": "ALLOWED",
            "pdf": true
        },
        {
            "title": "Tôi tài giỏi, bạn cũng thê!",
            "authors": [
                "Adam Khoo"
            ],
            "publisher": "Unknow",
            "publishedDate": "2018",
            "description": "Unknow",
            "buyLink": "Unknow",
            "textToSpeechPermission": "ALLOWED",
            "pdf": false
        },
        {
            "title": "I Am Gifted, So are You!",
            "authors": [
                "Adam Khoo",
                "Tony Buzan",
                "Ernest Wong"
            ],
            "publisher": "Unknow",
            "publishedDate": "2002",
            "description": "Unknow",
            "buyLink": "Unknow",
            "textToSpeechPermission": "ALLOWED",
            "pdf": false
        }
    ]
}
```

```xml
<books>
    <query>Tôi tài giỏi bạn cũng thê</query>
    <totalItems>466</totalItems>
    <book id="0">
        <title>I Am Gifted, So Are You</title>
        <authors>Adam Khoo</authors>
        <publisher>Marshall Cavendish International Asia Pte Ltd</publisher>
        <publishedDate>2014-07-15</publishedDate>
        <description>Every student can achieve and excel if given the opportuni
        <buyLink>https://play.google.com/store/books/details?id=WIZ1BAAAQBAJ&am
        <textToSpeechPermission>ALLOWED</textToSpeechPermission>
        <pdf>True</pdf>
    </book>
    <book id="1">
        <title>Tôi tài giỏi, bạn cũng thê!</title>
        <authors>Adam Khoo</authors>
        <publisher>Unknow</publisher>
        <publishedDate>2018</publishedDate>
        <description>Unknow</description>
        <buyLink>Unknow</buyLink>
        <textToSpeechPermission>ALLOWED</textToSpeechPermission>
        <pdf>False</pdf>
    </book>
    <book id="2">
        <title>I Am Gifted, So are You!</title>
        <authors>Adam Khoo, Tony Buzan, Ernest Wong</authors>
        <publisher>Unknow</publisher>
        <publishedDate>2002</publishedDate>
        <description>Unknow</description>
        <buyLink>Unknow</buyLink>
        <textToSpeechPermission>ALLOWED</textToSpeechPermission>
        <pdf>False</pdf>
    </book>
</books>
```

Exercise2: Google Books API



Content

- **Exercise1: PASCAL VOC and CreateML Annotation Format**
  - Extract information from PASCAL VOC and CreatML annotation files for the Object Detection task
- **Exercise2: Google Books API**
  - Search, extract, and save necessary book information with the Google Books API
- **Exercise3: Compensation for Different Focal Length Images**
  - Compensate the short focal length image to achieve approximately the same result as the long focal length image

Exercise3: Compensation for Different Focal Length Images

Thực hiện xây dựng một hàm có chức năng compensate focal length cho một cặp image với focal length khác nhau

• **Input**: Hai ảnh có cùng kích thước nhưng khác nhau về focal length như hình 9

• **Output**: Hai ảnh, một ảnh được chụp với long focal length camera (ảnh trái) hình 14 và một ảnh được chụp với short focal length camera đã được compensate như hình 14 (ảnh phải)

Long Focal Length Short Focal Length

Hình 9: Input ảnh trái và phải khác nhau về focal length

Exercise3: Compensation for Different Focal Length Images

Long Focal Length

Rectified Short Focal Length

Hình 14: Ảnh thể hiện kết quả sau khi xử lý vấn đề focal length khác nhau giữa hai ảnh. So sánh ảnh long focal length (ảnh trái) và output short focal length sau khi được compensate (ảnh phải)

Exercise3: Compensation for Different Focal Length Images

Các bước thực hiện

Exercise3: Compensation for Different Focal Length Images

```
[ ]  1 !pip install --upgrade pip
     2 !pip install opencv-python==4.5.5.64
```

```
[ ]  1 !gdown --id 1A_HmSetIc9J2FvhE-BNAOxtMNxEqE2nT
[ ]  1 !gdown --id 1tYKv0SZQjETVRM-lf6dx9emKpk9aDWTv
```

```
[ ]  1 import numpy as np
     2 import cv2 as cv
     3 import matplotlib.pyplot as plt
     4 from google.colab.patches import cv2_imshow
     5
     6 left_img_path = "/content/detect1_0.bmp"
     7 right_img_path = "/content/detect1_1.bmp"
     8
```

```
[ ]  1 left_img, right_img, left_points, right_points = get_pair_keypoints(left_img_path, right_img_path)
     2 draw_keypoints(left_points, left_img)
     3 draw_keypoints(right_points, right_img)
```

```
[ ]  1 left_img, right_img, left_points, right_points = get_pair_keypoints(left_img_path, right_img_path)
     2 draw_keypoints(left_points, left_img)
     3 draw_keypoints(right_points, right_img)
```

Hình 10: Các keypoints của hai ảnh input(hàng trên) và map keypoints tương ứng giữa hai ảnh (hàng dưới)

Exercise3: Compensation for Different Focal Length Images

Long Focal Length

Short Focal Length

Euclidean distance: $d(p1, p2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Inner-distance: tìm Euclidean distance từ một point đến tất cả các point trong cùng một ảnh. Lặp hết tất cả các point

inner-distance_list 1

inner-distance_list 2

inner-distance_list 1 ÷ inner-distance_list 2 = ratio_inner-distance_list

scale = **mean**(ratio_inner-distance_list)

Exercise3: Compensation for Different Focal Length Images

Short Focal Length

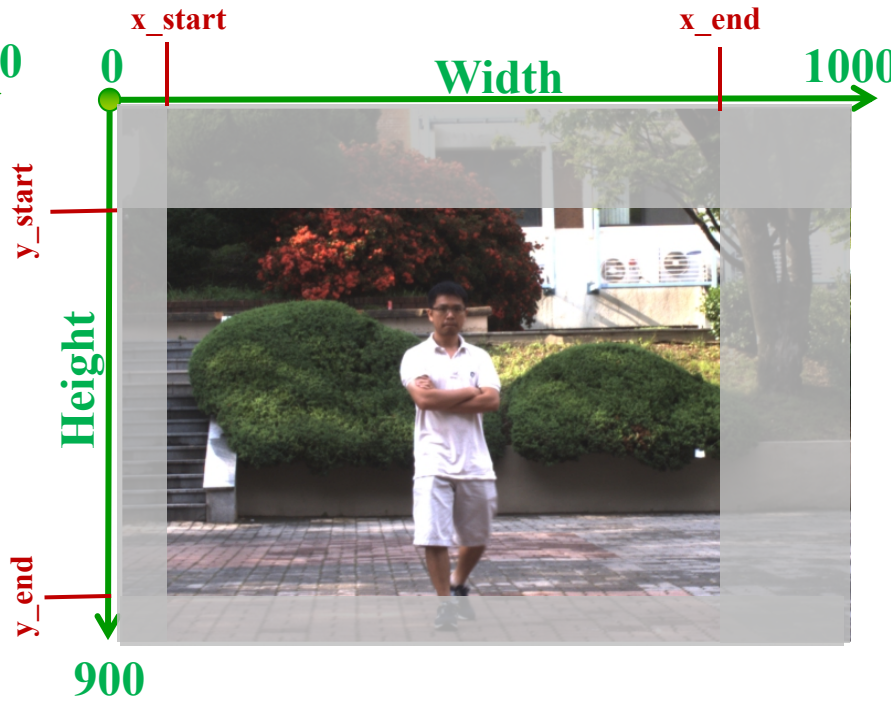
resize width*scale
height*scale

Resized Short Focal Length by scale

Exercise3: Compensation for Different Focal Length Images

```
1  crop_img = img[y_start:y_end, x_start:x_end,:]
```

x_start = 100
x_end = 300
y_start = 220
y_end = 100



Exercise3: Compensation for Different Focal Length Images