

Python如何进行控制



大纲

- 自动驾驶车?
- 操作系统 ? 中断、异常、系统调用
- Python调用C..
- 作弊 ?

数据科学

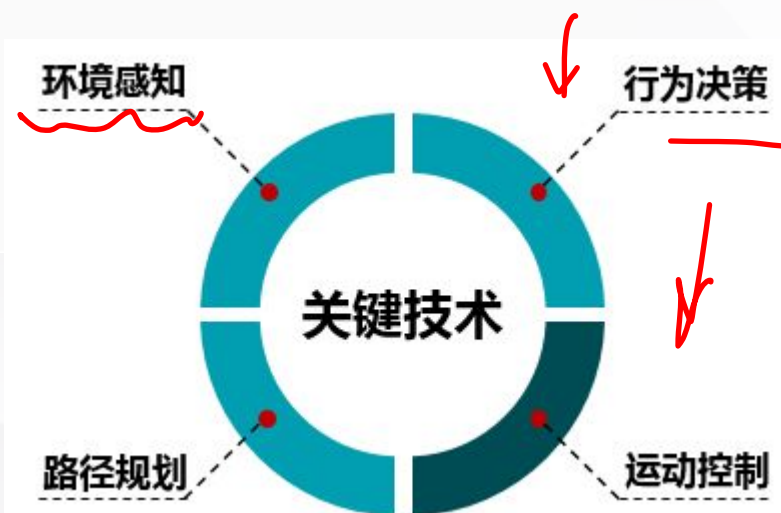
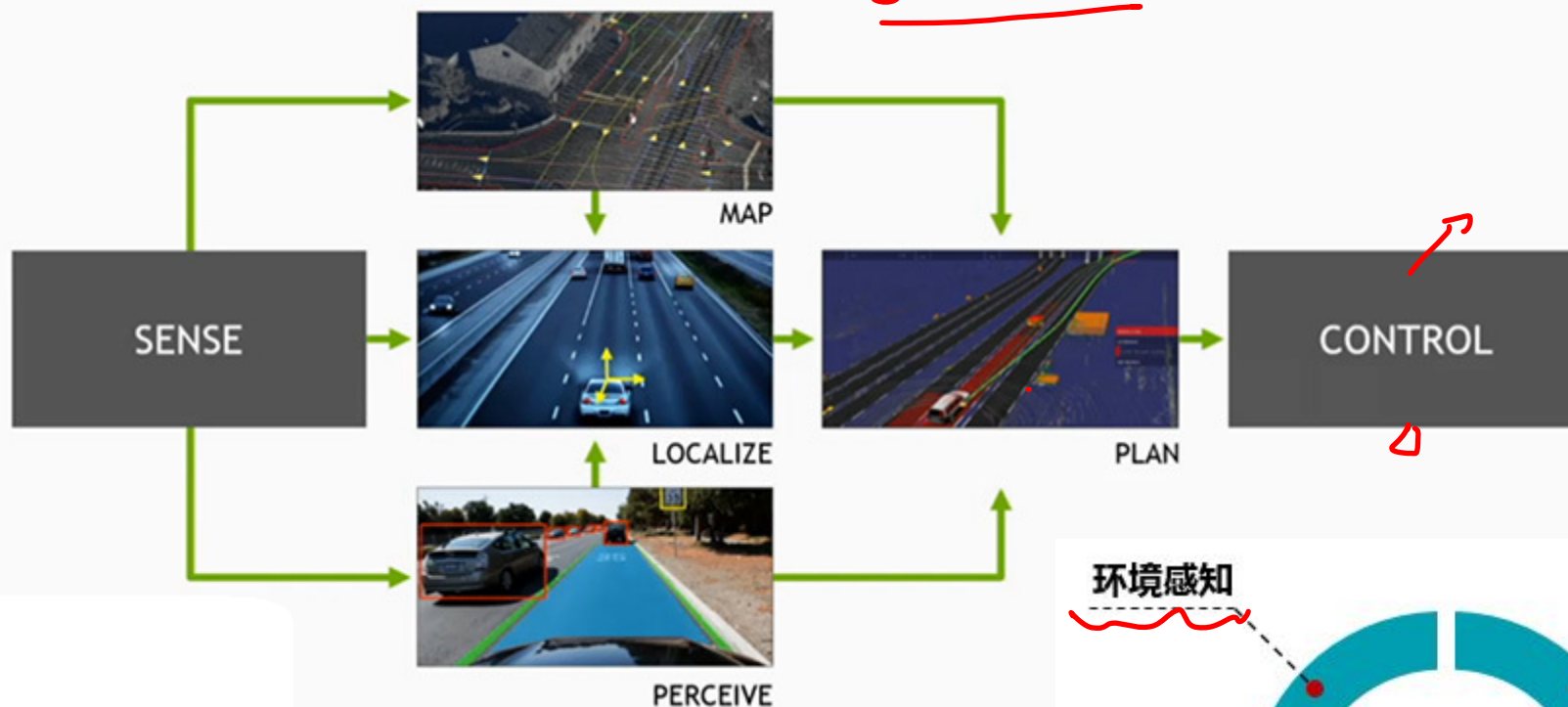


The basic self-driving loop

camera

sensor

Feature



<https://www.nvidia.com/en-us/self-driving-cars/drive-px/>

GPU

Powerful Hardware

150 MACBOOK PROS IN YOUR TRUNK



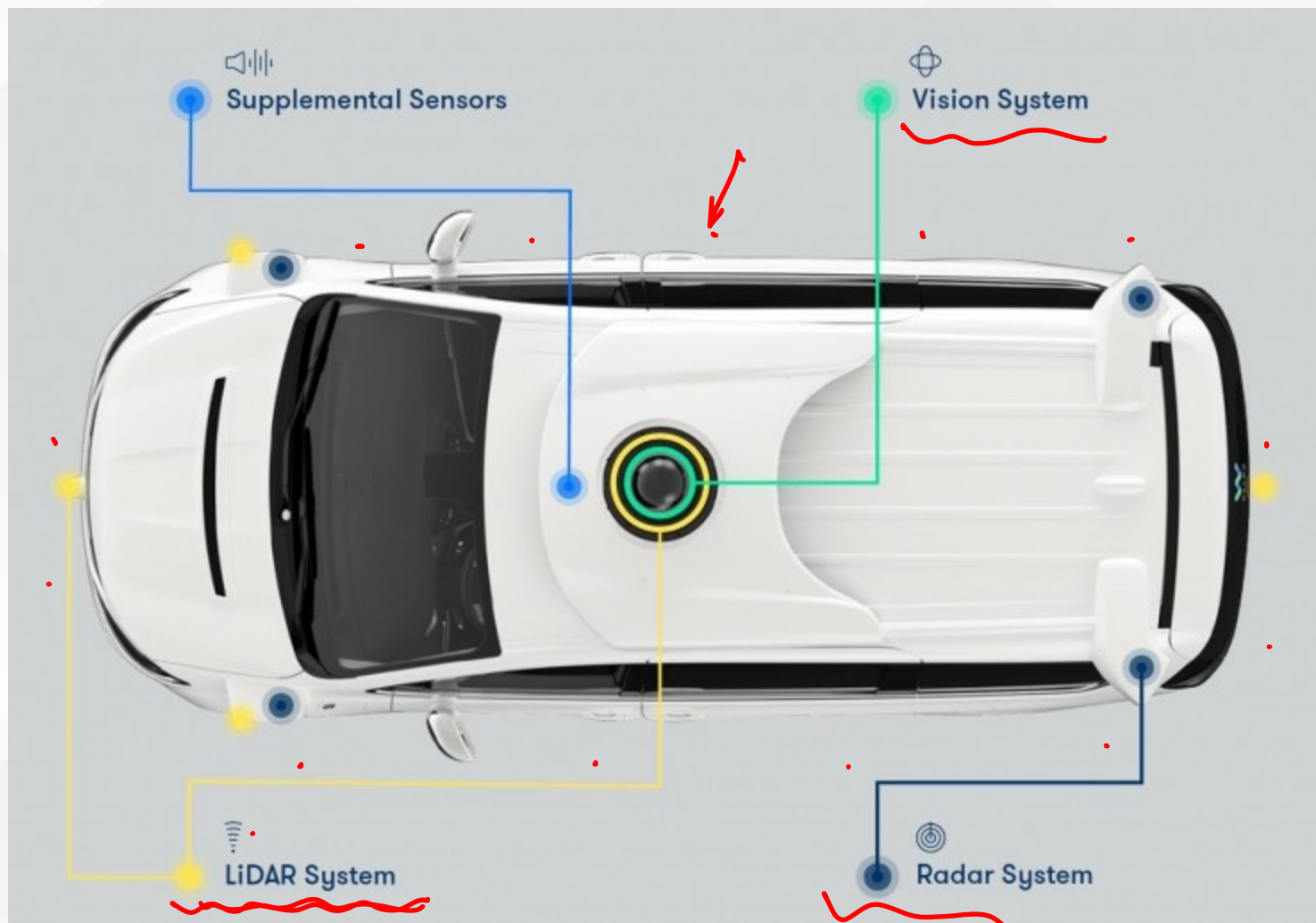
NVIDIA DRIVE PX

6 TITAN X = 42 TFLOPS, Core i7 = 280 GFLOPS, $42 / 0.28 = 150$ MacBook Pros.

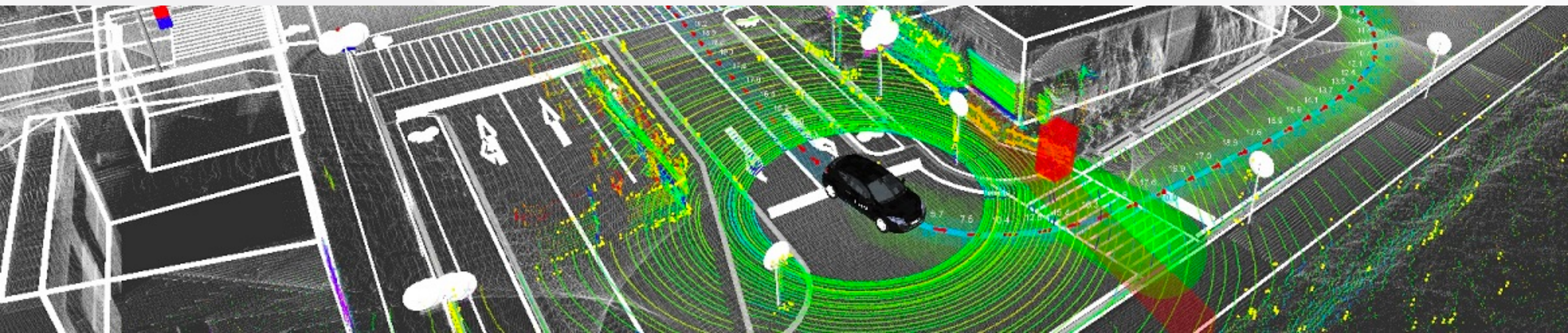
传感器

- GPS (定位)
- 里程计
- 陀螺仪
- 视觉系统
- 数传电台

sensor fusion



LIDAR(激光雷达) 即Light Detection And Ranging



LiDAR（光检测和测距）的日夜工作，可在每秒内输出数百万个360°的激光脉冲，测量反射到表面并返回车辆所需的时间

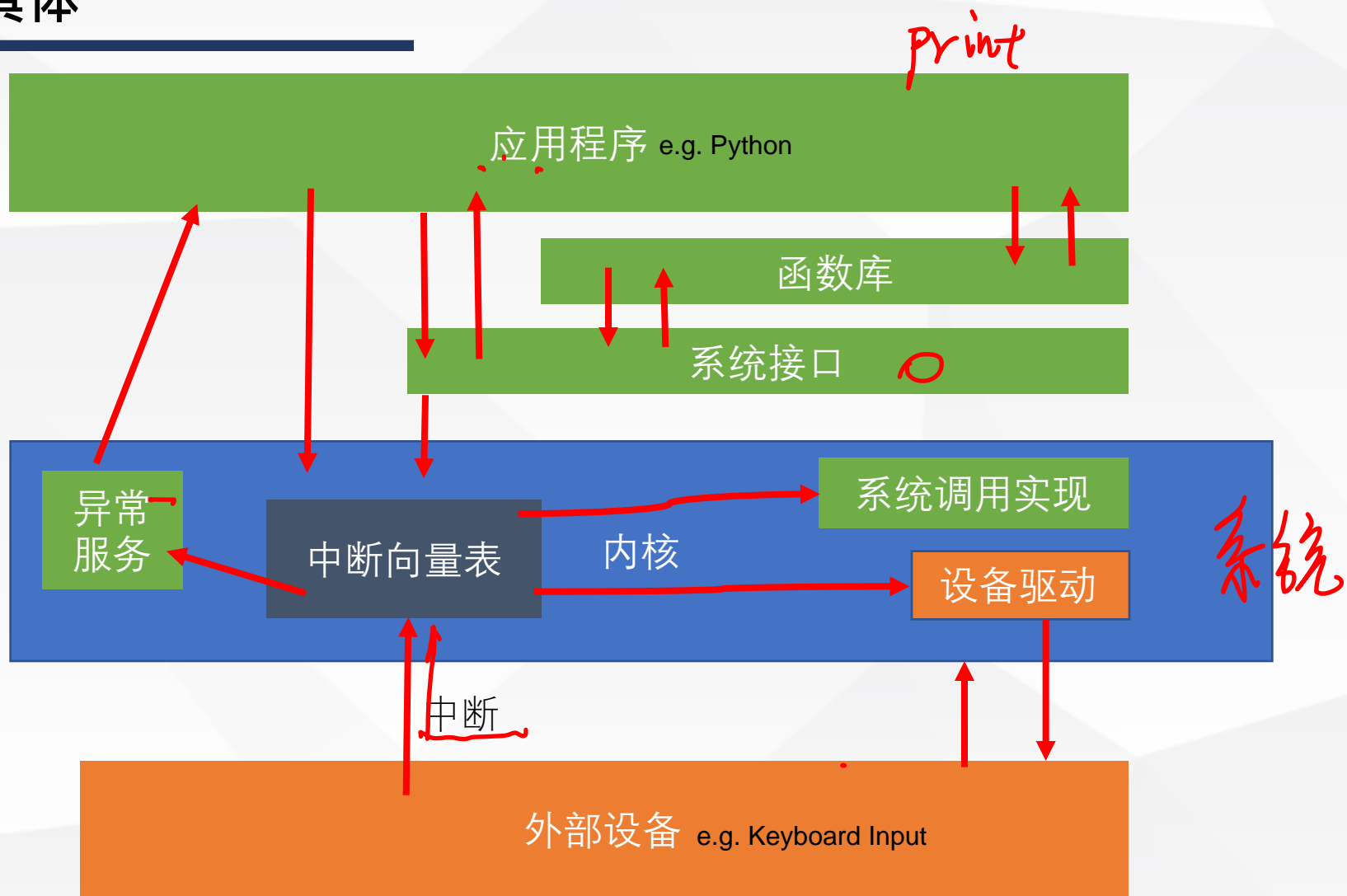
<https://www.autoware.ai/>

中断、异常和系统调用

Interaction between App (e.g. Python) and Operating System and Outside Device (e.g. Keyboard input)

- 计算机运行中，内核被信任
- 只有内核可以执行特权指令
- 方便应用程序交互
- 中断和异常，
- 当外设与系统交互时，如何处理
- 程序处理意想不到的行为时，会出现什么现象
- 系统调用
- 用户应用程序如何得到系统服务
- 系统调用 vs 功能调用

具体



比较

- 系统调用，应用程序，异步或同步
- 应用程序主动向操作系统发出的服务请求
- 异常，意想不到的行为，同步
- 非法指令或其他原因导致的指令执行失败
- 中断，外设，异步
- 来自硬件设备的处理请求

printf

```
#include <stdio.h>
int main()
{
    .
    printf("greetings")
    .
}
```

用户态

内核态

标准C库

Write()
系统调用实现

→ 文件
descriptor

文件系统和文件

- 文件系统是操作系统中管理持久性数据的子系统，提供数据存储和访问功能
 - 组织、检索、读写访问数据
 - 大多数的计算机系统都有文件系统
- 文件具有符号名，由字节序列构成的数据项集合

Python vs C++

Python

```
x=5  
print(x)
```

C++

```
#include <iostream>
```

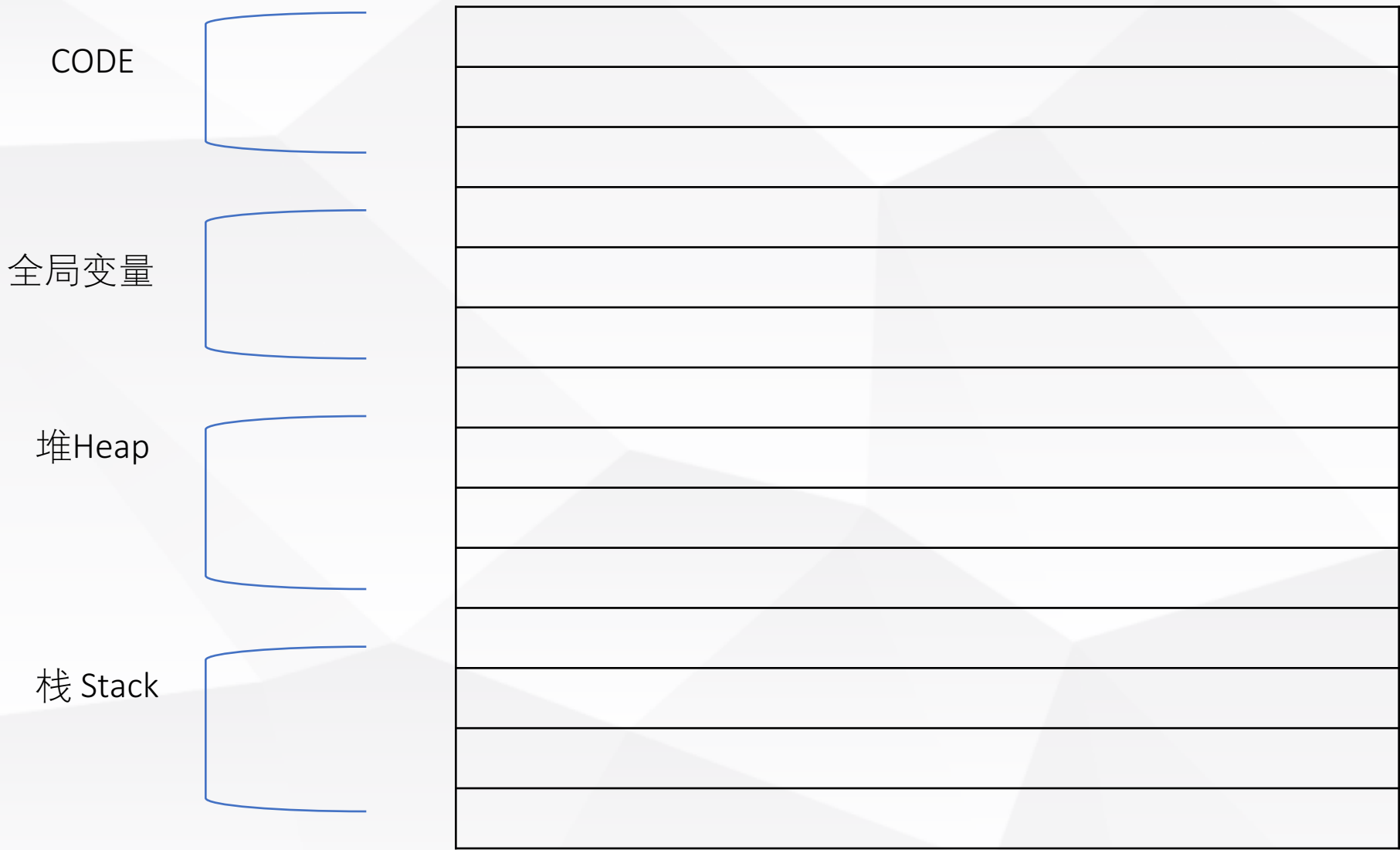
```
int main(){  
    int x;  
    x= 5  
    std::cout <<x;  
    return 0;  
}
```

Static vs dynamic Typing

All Variable are Binary

- A char is 1 byte
- A 16-bit integer would be 2 bytes
- A 32-bit integer would be 4 bytes
- A 32-bit float would be 4 bytes

How C++ Use RAM



performance

- Time
- Memory Usage
- Power Consumption

谢谢

beijing@dataapplab.com