

从  SCRATCH 3.0

到  python

# 中小学信息技术师资培训班

厦门大学中美青年创客交流中心  
厦门大学海西通信工程技术中心

2021.11.11-2021.11.14



利用Python做些有趣的事



# 主要内容：

- **学习Python文件读取和简单数据处理。**
- 介绍Python目前常用的几个领域，主要在于介绍和展示，帮助大家了解学习Python未来的应用空间。

主要分为一下几个方面：

- **数据处理**：使用Python的NumPy和Pandas库快速实现数据处理。
- **数据可视化**：数据可视化可以将数据直观的展现在人们眼前，帮助人们进行数据分析。这部分主要展示一些数据可视化的案例。
- **结合硬件的Python**：如何利用Python控制硬件呢？有一种基于Python语法的工具叫做MicroPython，可以让我们使用Python语言控制硬件。这一部分主要展示使用Micropython控制树莓派Pico案例。

在介绍几个方面的同时，会给大家直观的展示如何简单的对他们进行应用。



# 目录

壹

文件读取 & 数据处理

Read & Write Files

貳

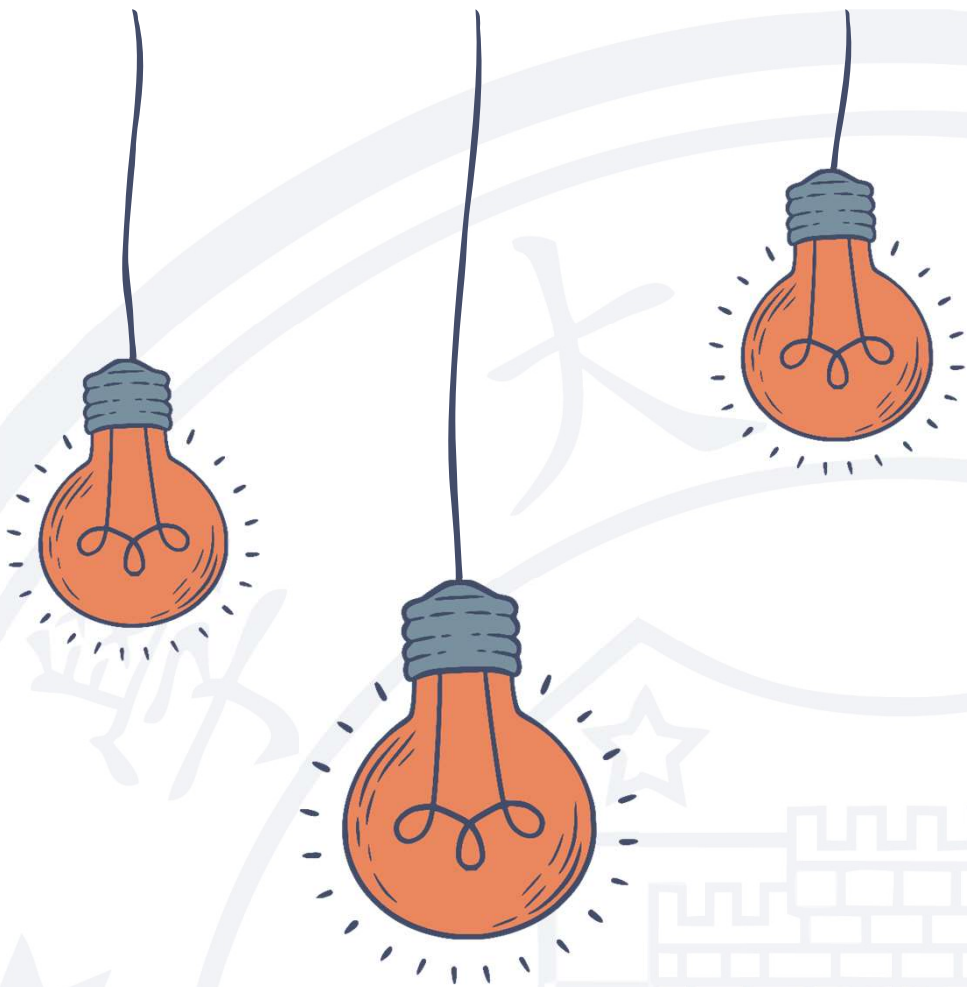
Python数据可视化演示介绍

Data Processing

叁

MicroPython和DIY

MicroPython and DIY



## Part 01

# Python 文件读取 & 简单数据处理

利用python读取文件，可以让我们处理表格信息，比如处理学生成绩信息。



# 学习目标

Chun Qiu Advertising Original Design



## 1. 什么是文件?

介绍文件和路径基础知识。了解计算机的奥秘。



## 2. 如何读取文件?

学习读取普通文件和表格文件并将数据转化成可以处理的模式。  
(pandas maybe)



## 3. 从读取文件到处理数据

利用numpy模块快速处理数据，包括取和，求平均值，排序等常见操作。



## 4. 给数据换个装

我们可以很好的处理数据，但是如何更好的展现数据，才能令人影响深刻呢?





# 第一部分 什么是文件?

what is file and what is the path of file?

0101000100001  
111100101...

计算机中的文件都是按照二进制的方式存放的。这些信息中包含文件的类型，大小。储存了什么样的信息。

**我们怎么判断一个文件的类型呢？是图片还是文本？**

一般来说，我们通过观察文件的后缀确定文件的类型。对于不同的文件，我们要使用不同的方法取读取它。操作系统会根据文件类型显示不同的图标帮助我们区分，同时会为我们选择合适的软件打开某一类文件。

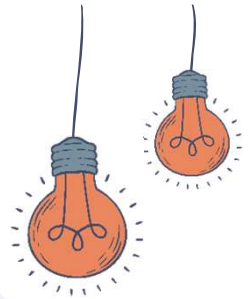
文件的真实样子



**我们可知，不同的文件拥有不同的结构，但他们的共同点就是都包含信息。那么，我们应该如何从文件中读取并修改信息呢？**

对于文本文件，PPT，表格文件，图片等，我们点击文件时，操作系统会自动为我们选择合适的程序打开，在这个过程中，软件会按照既定程序读取信息然后直观的呈现给我们。这是，我们就可以使用程序修改或者查看文件了，比如修图软件，文档编辑，表格处理等。

**学习目标：让我们可以批量处理大量结构相同的文件。比如处理大量文件，分析数据等**



# 第一部分 什么是文件路径?

what is file and what is the path of file?

刚才我们知道了什么是文件，那我们在磁盘中找到特定文件呢？

联想问题：

双十一在网上买东西，快递过几天就会送到我们大家的手里，快递是如何正确的送到我们的位置的呢？

福建省 厦门市 思明区 思明南路 422号

上面是厦门大学思明校区的地址，根据这个地址，快递员就可以正确的把快递送到学校。由此我们可以联想，在操作系统中找到一个文件呢？就是地址，在操作系统中，这个地址就是**文件的路径**。

D:\Desktop\一个文本文件.txt

上面是一个文本文件的路径(Windows系统下)，我们在操作系统中可以正确的找到这个文件。在上面地址中，分隔不同部分的是**空格**。在Win系统下，这个分隔符为**反斜杠 (“\”)**，在Linux/Mac系统中，这个分隔符为**斜杠 (“/”)**。

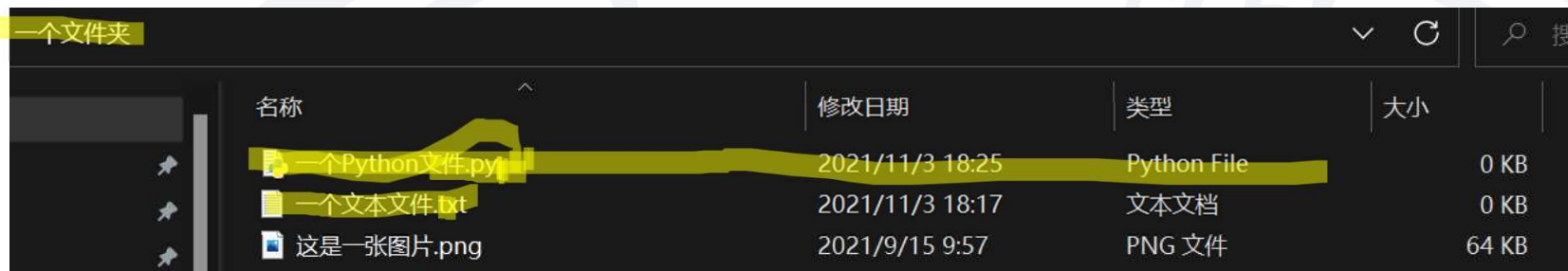
路径还分为**相对路径**和**绝对路径**。对于上面地址，我们还可以在前面加上中国，地球，太阳系……之类的限定，但是对于一个收件人和寄件人都在国内的包裹来说，那些信息显得十分多余。

对于一个程序来说也是如此，如果需要读取的文件就在身边，可以按照**相对路径**直接读取，如果文件在很远的地方，可以按照**绝对路径**读取。



# 第一部分 相对路径和绝对路径

Relative path and Absolute path



名称	修改日期	类型	大小
一个Python文件.py	2021/11/3 18:25	Python File	0 KB
一个文本文件.txt	2021/11/3 18:17	文本文档	0 KB
这是一张图片.png	2021/9/15 9:57	PNG 文件	64 KB

## 相对路径

这个文件夹中，我们知道，文本文件和python文件在同一个文件夹中，那我们文本文件相对于Python文件的路径为“**.\一个文本文件.txt**”或“**一个文本文件.txt**”。

在操作系统中，一个点(“.”)表示当前目录下，两个点(“..”)表示上一级目录。

## 绝对路径

**D:\Desktop\一个文本文件.txt**

上面个就是一个绝对路径，它需要包含文件从根目录到这个文件的具体路径。（在Win系统中，分为不同的磁盘，磁盘的根目录就是盘符加冒号，了解）

## 第二部分 如何读取一个文件?

How to read a file?

例子：打开文件

主要的几个模式：

r: 只读模式，指针放在文件头。

w: 只写模式，这个模式会将已存在的文件清空。

a: 追加模式，在一个已存在的文件中追加内容，指针放在文件末尾。

相对路径或者绝对路径

文件路径

打开方式

其他参数

可能用到的参数：指定编码方式

encoding = "utf-8"

with open(file\_path, mode, ...args) as f: →

f.read()

f.readline()

f.readlines()

直接理解：用mode模式打开一个文件路径在file\_path的文件，作为f。

直接读取文本内所有内容，不推荐这种方式，如果文本太大，读取会非常缓慢。返回字符串类型。

每一次读取一行，再次执行时再读取下一行，返回的时字符串类型。

按照行读取整个文件，每一行都成为列表里的一个元素，最后返回一个列表。

在vscode演示



## 第二部分 读一个表格?

Read a form?

student\_score.csv

```
1  学号,高数,英语,Python
2  1001,89,67,90
3  1002,67,98,91
4  1003,76,90,86
5  1004,97,91,89
6  1005,76,32,65
7  1006,44,89,54
8  1007,78,96,34
9  1008,67,78,76
```

一个简单的成绩表格

### 两种方式

1. 自己编写代码读取。
2. 使用现有的模块。

### CSV表格文件

这中表格结构简单，每一项用分隔符隔开，分隔符可以是逗号，空格等。

一般的xls文件可以转存为csv格式文件。如果手动读取表格，通常读取的就是这种文件。

如果使用现有模块，可以直接这种csv文件和读取xls文件。

### 学习重点：

手动读取只是为了复习前面学习的知识，所以在真是使用中很少手动处理数据。使用模块读取相对简单，在使用时也更加方便。

### 下面进行代码实操



## 第三部分 从读取文件到处理数据

Simple data processing

### 介绍

数据处理在python中使用最多的模块是Numpy和Pandas。在深度学习中，这个模块广泛用于数据清洗和数据归一化。在数据可视化中，也经常用到这两个模块。不过这两个模块的用法非常多，我这里只做一些简单的演示。

前面已经介绍了如何使用Pandas读取表格文件。下面介绍一些简单的数据处理操作。数据处理中最常用的集中操作：**求平均值**，**数据筛选和排序**。我主要介绍这几种操作的一些方法。

### 学习和查询

中文Pandas网站（中文友好）：<https://www.py pandas.cn/docs/>

官网（资料详细，最先更新）：<https://pandas.pydata.org/docs/>

简单的入门教程（内容不全）：<https://www.runoob.com/pandas/pandas-tutorial.html>

### 学习重点

计算机科学不同于其他学科，学习计算机要面对大量的信息量。如果单纯依靠记忆力记住知识点是非常难得。这就要求我们要用很强的总结能力。要记住规律而不是某一个符合规律的特例。

正如我们学习乘法一样，乘法的本质是加法，而我们计算乘法的方式却不是完全的按照加法去做。我们是先记住了一些基础知识（乘法表），然后掌握了乘法的规律（如何计算乘法）。这样我们就可以轻松了实现任何两个书的乘法。

**学习计算机就是给自己的思维升维的过程，掌握规律比掌握知识本身更重要。**



## 第四部分 给数据换个装?

Data visualization

数据处理流程

原始数据

数据筛选

数据处理

?

数据可视化

数据可视化是指将数据以视觉形式来呈现，如图表或地图，以帮助人们了解这些数据的意义。如果将一堆数字放在人们面前，人们可能需要仔细分析才能看懂数据反映的情况。但如果使用图表的形式，那么数据的特征就会一览无余。

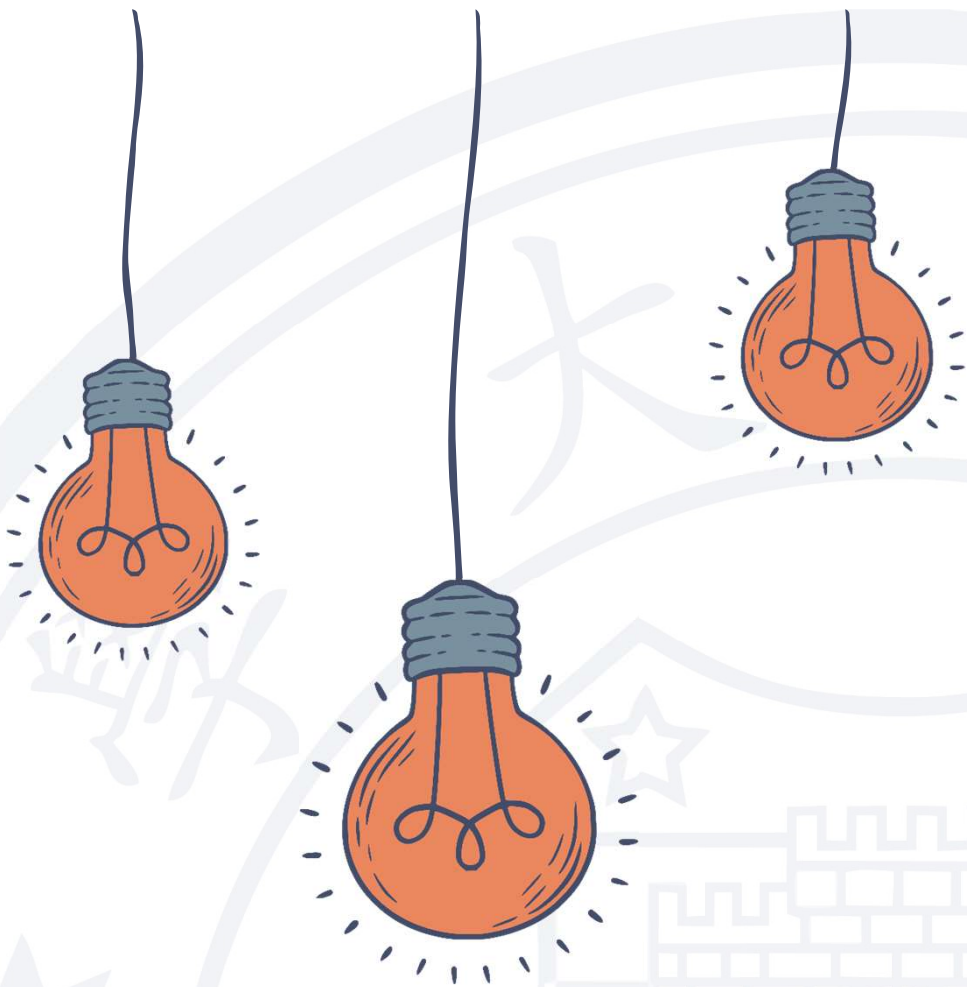
Python中提供数据可视化的对应模块，我们可以使用这些模块轻松的做出很多有趣的图表。

Python中的数据可视化

在Python中，matplotlib是一个非常常用的绘图模块，我们目前只会用到它的子模块pyplot，所以我们在引用时，只需要引入子模块即可。经典的引用方法，如下

```
import matplotlib.pyplot as plt
```





## Part 02

### Python数据可视化 演示和介绍

Python data vision



# Python模块介绍

Learning skills

## 学习方法

**首先我们思考：为什么会产生这些模块？**

（这些模块的诞生就是计算思维的体现，他们都是一中固定的模式，在我们数据处理过程中经常需要使用到。就比如计算平均值，我们不可能每一次都写一遍循环去计算平均值，直接封装成mean函数，可以快速的得出一组数据的平均值）

**进一步思考：如何学习这些模块？**

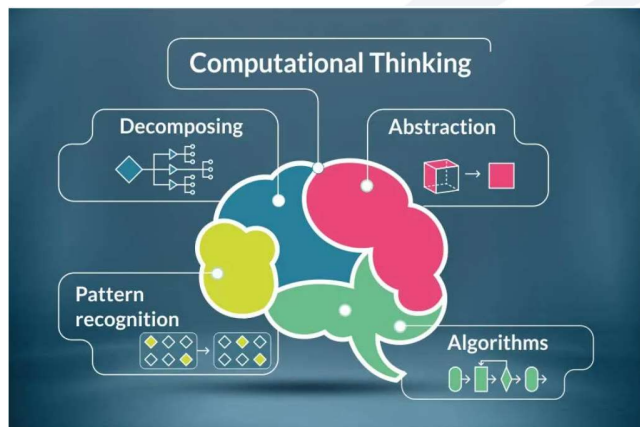
（授之以鱼不如授之以渔，与其去掌握每一个知识点，不如学会自己查找Python库文档的方法。当掌握自行查询Python库的方法（固定模式）后，即使我们不能同时掌握所有知识点，也可以在需要用到某些功能的时候，通过查看文档的方式掌握需要用到的知识点）

**Python模块的思想是计算思维的体现。**



# 学习这个部分的分析方法

Learning skills



## 计算思维可以划分为四个主要组成部分：

- **解构或分解**，即把问题进行拆分，同时厘清各个部分的属性，明晰如何拆解一个任务；
- **模式识别**，即找出拆分后问题各部分之间的异同，为后续的预测供依据；
- **模式归纳**，或“抽象化”，即探寻形成这些模式背后的一般规律；
- **算法开发**，即针对相似的问题提供逐步的解决办法。

## 模式归纳：

**提出问题：** 如何进行处理大量特定结构的信息？比如学生成绩信息，公司财务信息等。

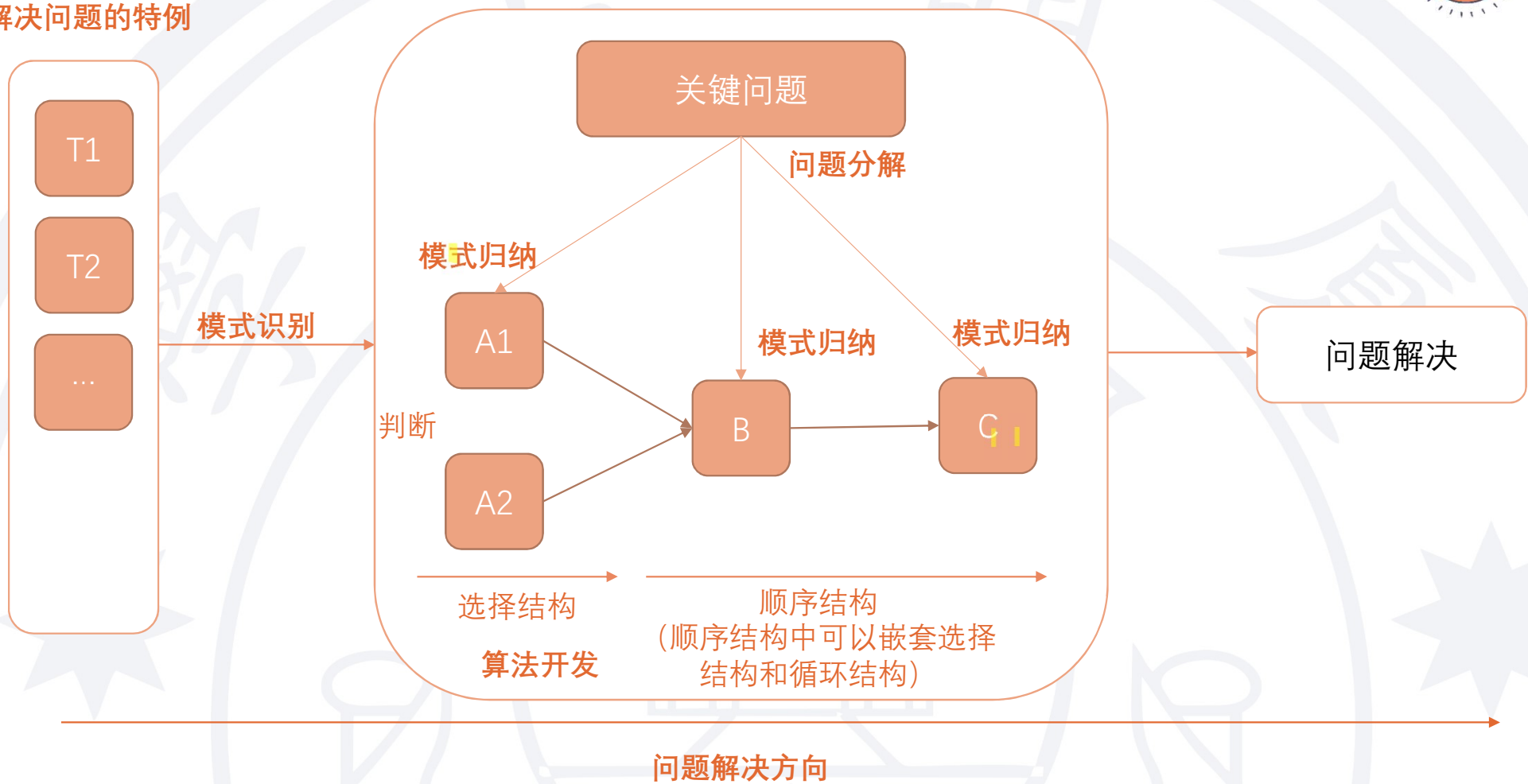
**特例解决：** 第一次，我们可以按照特定步骤对数据进行处理（处理过程包括：过滤，求和，平均，排序等），我们将这次处理记作T1。第二次，我们仍然可以按照类似的方法处理同样结构的数据，记作T2。那有没有办法更加高效的解决这个问题呢？

**模式归纳：** 假设我们可以将上述问题拆分为三个部分（A，B，C），其中B，C两个部分的规律具有一致性，再假设A部分可以分成两种情况（A1，A2），那我们就可以将上述问题的解决分为（A1，A2，B，C）四大块，分别实现。按照如下图设计算法。

# 学习总结和学习思想

Learning skills

解决问题的特例





# 数据可视化前奏：Pandas库介绍

## Introduction to Pandas

### Pandas是什么？

Pandas是一个强大的分析结构化数据的工具集；它的使用基础是Numpy（提供高性能的矩阵运算）；用于数据挖掘和数据分析，同时也提供数据清洗功能。

[IO工具（文本，CSV，HDF5，...）](#) | [Pandas 中文 \(pypandas.cn\)](#)

### Pandas中重要的两个部分

Series

它是一种类似于一维数组的对象，是由一组数据(各种NumPy数据类型)以及一组与之相关的数据标签(即索引)组成。仅由一组数据也可产生简单的Series对象。

DataFrame

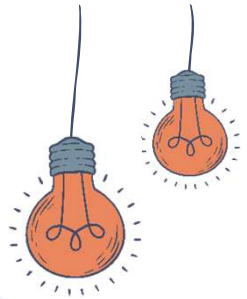
DataFrame是Pandas中的一个表格型的数据结构，包含有一组有序的列，每列可以是不同的值类型(数值、字符串、布尔型等)，DataFrame即有行索引也有列索引，可以被看做是由Series组成的字典。





# 数据可视化前奏：NumPy库

Python Module NumPy



## 何时NumPy?

NumPy是使用Python进行科学计算的基础软件包。除其他外，它包括：

- 功能强大的N维数组对象。
- 精密广播功能函数。
- 集成 C/C+和Fortran 代码的工具。
- 强大的线性代数、傅立叶变换和随机数功能。

[NumPy 用户指南 | NumPy 中文](#)

Numpy和Python自带的列表有很多相同的地方，但是Numpy的数组功能（Array）比列表更加强大好用，我们可以轻松的对数组进行维度变换，切片，等数据处理操作。



# 数据可视化核心：将数据可视化

Data visualization



Python中有一个将数据可视化的库叫做：Matplotlib

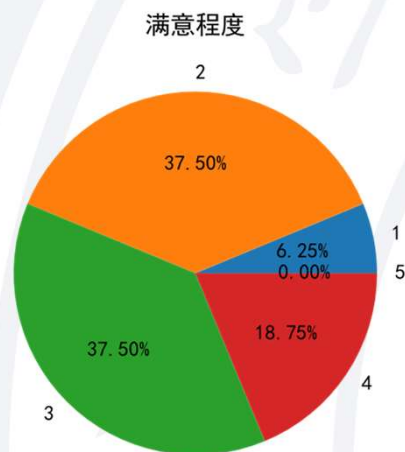
## 什么是Matplotlib?

Matplotlib是一个Python的2D绘图库，它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形。

[Matplotlib 中文](#)

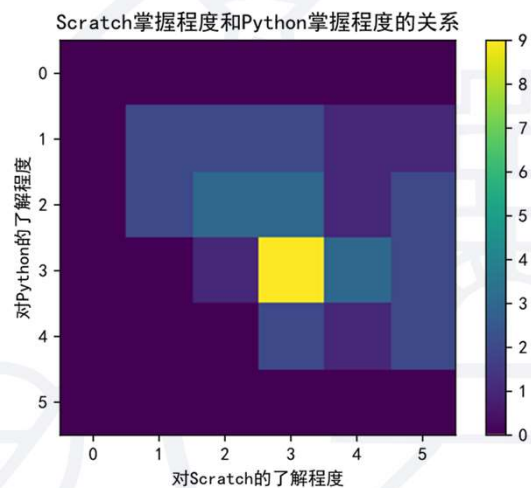
### 绘制圆饼图

`plt.pie()`



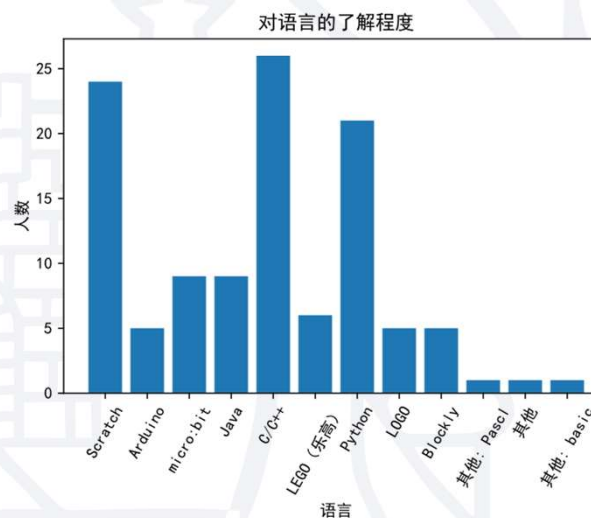
### 绘制热图

`plt.colorbar()`



### 绘制柱状图

`plt.bar()`





## Matplotlib续

More about matplotlib

Matplotlib主要能够绘制哪些图像？

散点图

直方图

柱状图

树地图

热力图

饼图

折线图

气泡图

雷达图

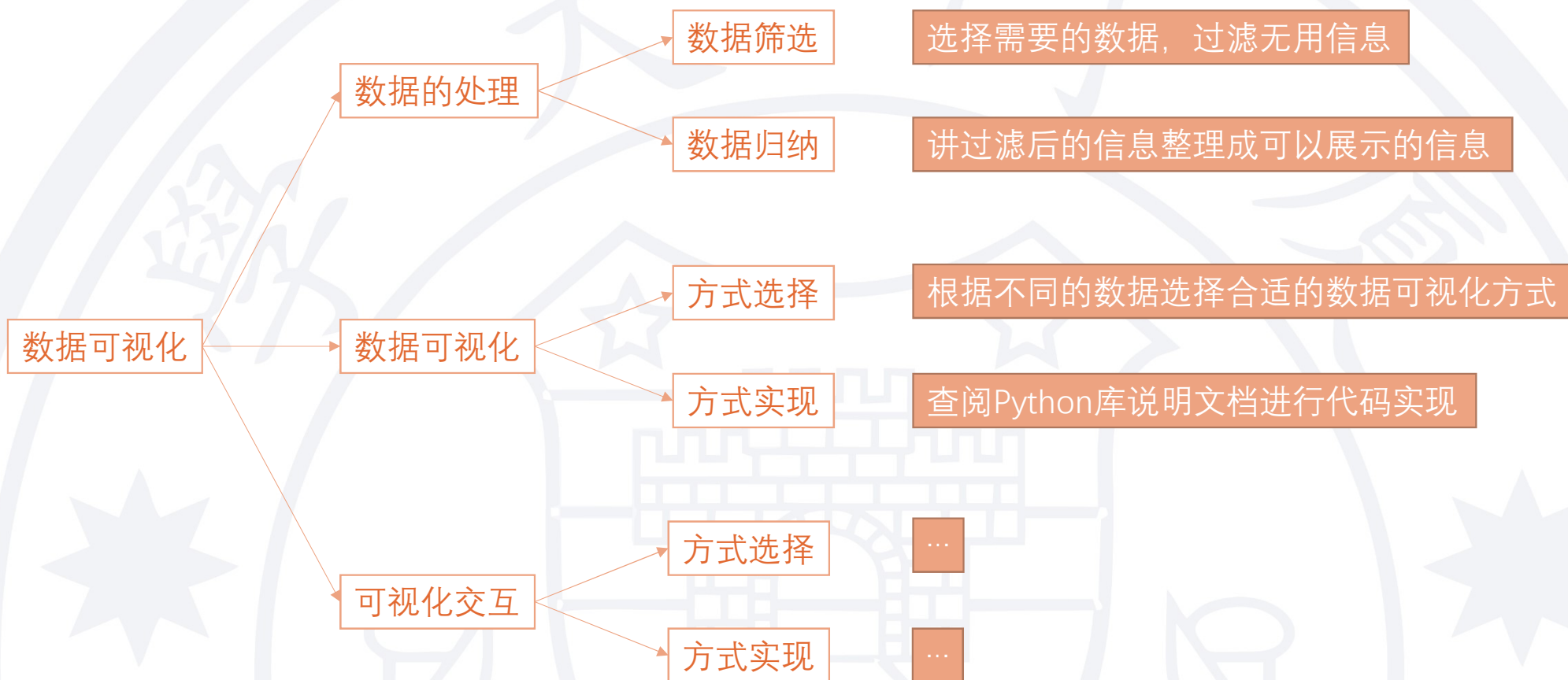
垂直堆叠条形图

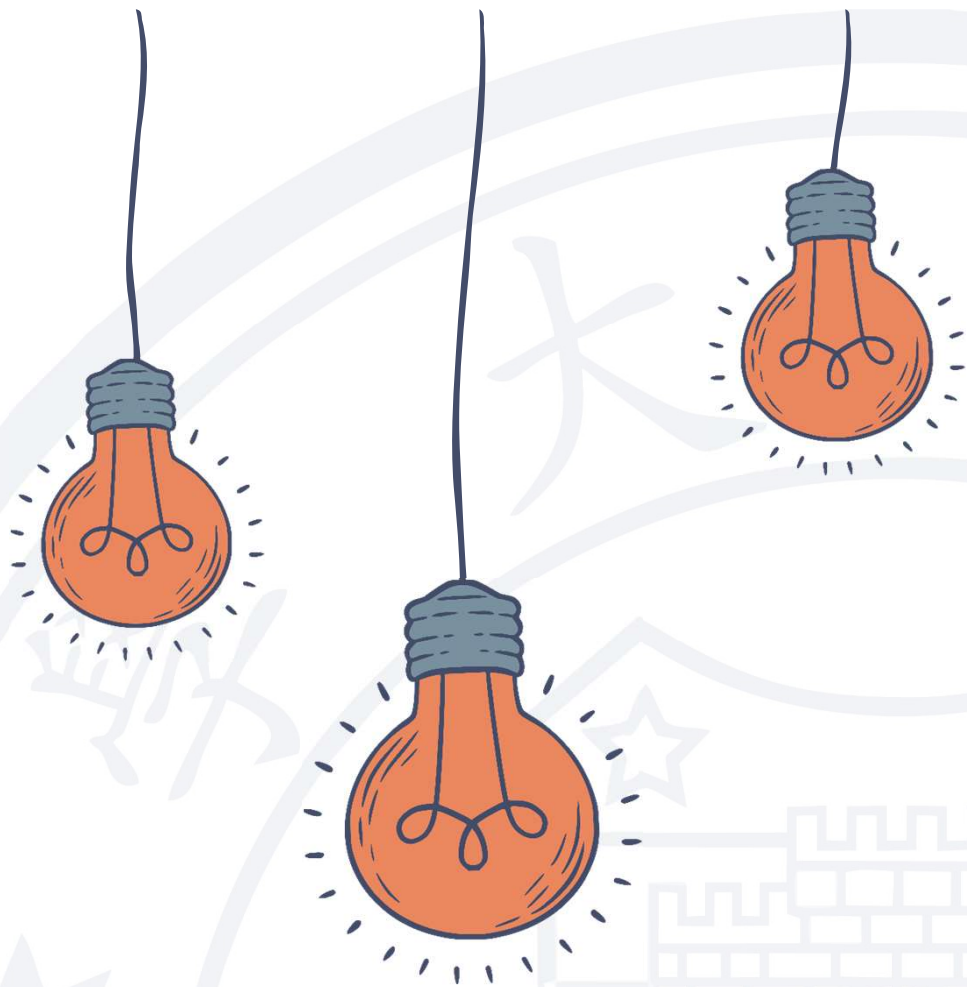
[\(28条消息\) Matplotlib画各种图\\_学海无涯-CSDN博客\\_matplotlib 图](#)



# 思想应用

The Application of skills





## Part 03

### MicroPython和DIY

MicroPython and DIY





# 什么是MicroPython?

What is MicroPython

## 介绍

**官方介绍:** MicroPython是Python 3编程语言的一种精简而高效的实现，它包含了Python标准库的一个小子集，并针对在微控制器和受限环境中运行进行了优化。

**简单的说:** MicroPython就是一些人基于Python3语法开发的专门用在单片机上面的语言。我们在单片机上最常用到的语言是**C语言**。例如Arduino。

那我们**为什么要用Python进行开发单片机呢?**

因为相对于C语言，Python要简单的多，它屏蔽了大部分硬件原理的部分，只给使用者提供丰富的功能操作，可以实现快速的单片机开发。（对于单片机的理解，可以认为是类似micro:bit和Arduino一类的东西）

[MicroPython 中文资源站-MicroPython Chinese Hub](#)

[使用MicroPython - 廖雪峰的官方网站 \(liaoxuefeng.com\)](#)

[MicroPython - Python for microcontrollers](#)



# 树莓派Pico?

What is Raspberry Pico



**树莓派 Pico**

灵活、易用的微控制器开发板

[树莓派 Pico 中文站](#)

[Raspberry Pi Pico - Waveshare Wiki](#)

## 介绍

Raspberry Pi Pico 是一款树莓派官方设计的低成本，高性能的微控制器开发板，具有灵活数字接口。硬件上，采用 Raspberry Pi 官方自主研发的 RP2040 微控制器芯片，搭载了ARM Cortex M0 + 双核处理器，高达 133MHz 的运行频率，内置了 264KB 的 SRAM 和 2MB 的内存，还板载有多达 26 个多功能的 GPIO 引脚。软件上，可选择树莓派提供的 C/C++ SDK，或者使用 MicroPython 进行开发，且配套有完善的开发资料教程，可方便快速入门开发，并嵌入应用到产品中。

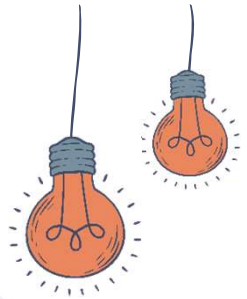
## 它能干什么？

桌面时钟，寻迹小车，游戏机，智能家居……开动你的脑洞，潜力无限。

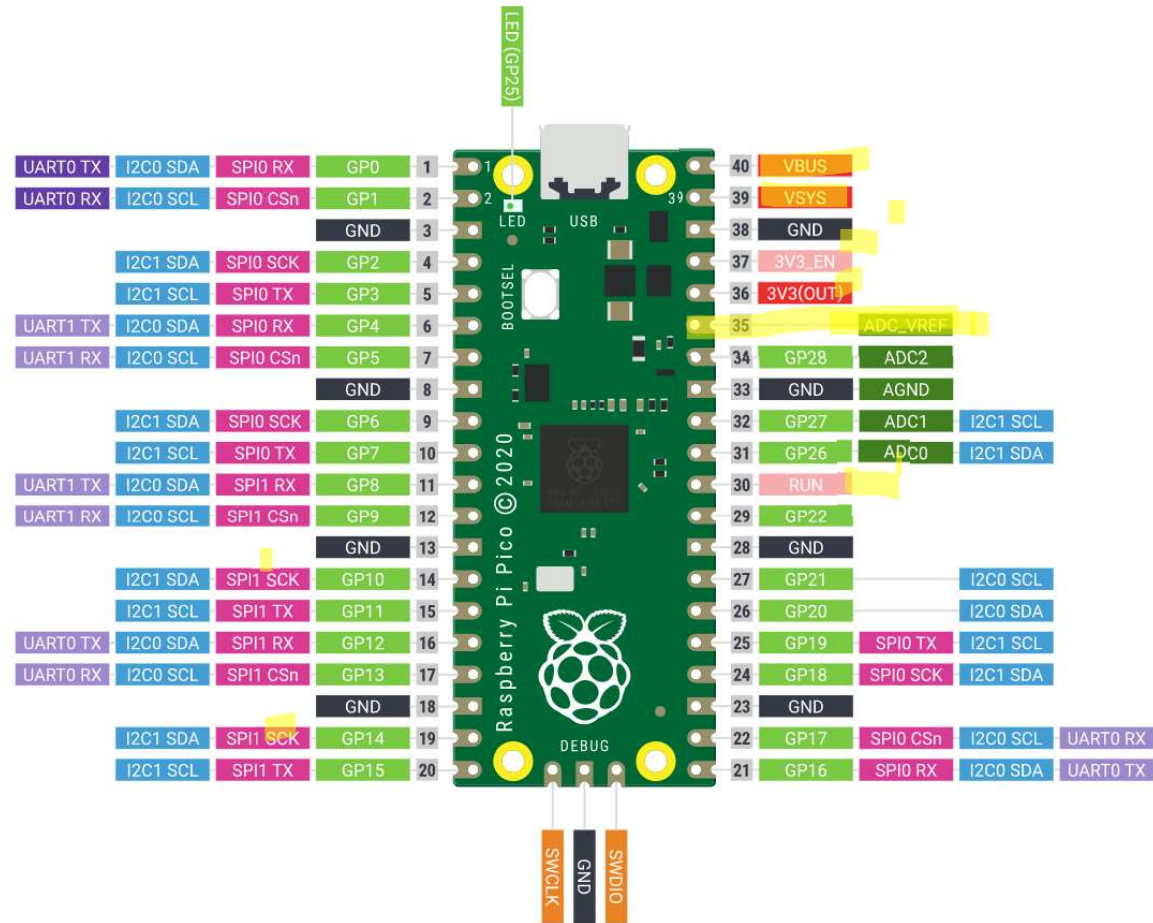


# 树莓派Pico?

What is Raspberry Pico



## Pico引脚图



Power	Ground	UART / UART (default)	GPIO, PIO, and PWM	ADC	SPI	I2C	System Control	Debugging
-------	--------	-----------------------	--------------------	-----	-----	-----	----------------	-----------



# 树莓派Pico!

Raspberry Pico yyds!

## 展示内容

案例一：利用Pico实现闪烁灯。  
--> 进一步实现呼吸灯。

Pin控制 PWM

案例二：利用Pico控制舵机。

舵机控制原理

案例三：超声波测距

超声波原理

**说明：**因为大家现在没有硬件，所以这部分更加侧重于原理的讲解，涉及到硬件原理的讲解，即使大家没有基础没有关系（原理相对简单）。可以结合这个部分融入教学中，增加课程的趣味性。希望在下午的分享中能够看到大家的分享。





# 案例一 点灯!

Case one: bright light!

## 什么是GPIO?

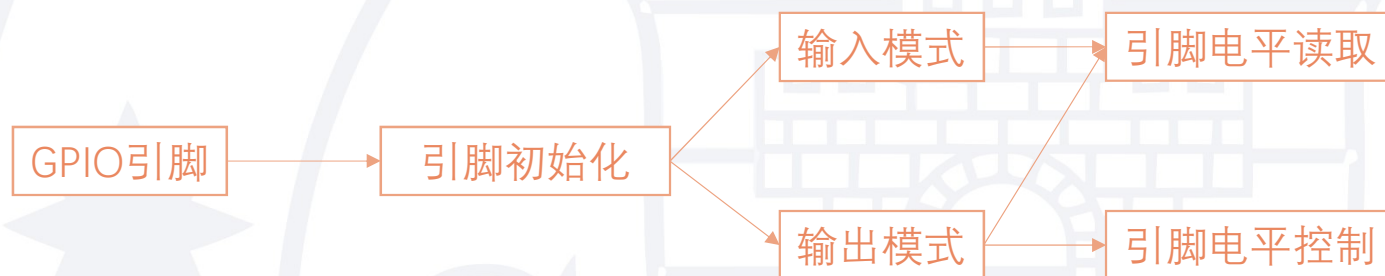
GPIO指的是通用型输入输出接口 (General Purpose Input/Output)。简单的理解, 我们可以通过代码控制这个接口的电压 (输出), 也可以通过代码读取这个接口的电压 (输入)。我们通常把这个接口称之为引脚。

在上面的Pico原理图中, GPIO口有很多, 我们把它简写GPx(x表示数字, 这个数字标记了我们在代码中要控制哪个引脚)

[Quick reference for the RP2 — MicroPython 1.17 documentation](#)

## MicroPython中GPIO控制

在MicroPython中, 我们使用machine这个内置模块来控制GPIO。具体用法如下:

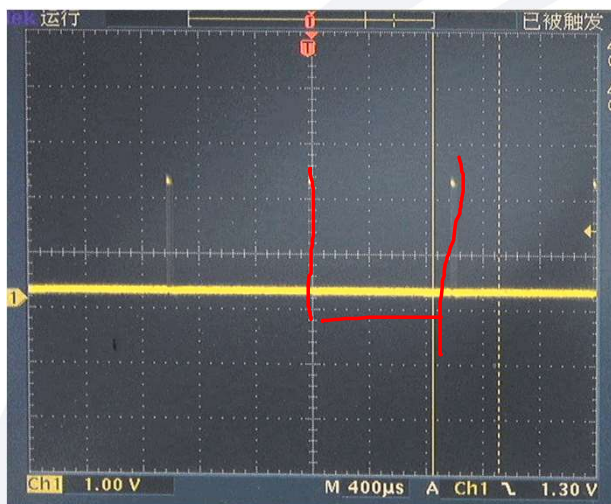






## 案例一 呼吸灯?

Case one: light breathe!



产生如图所示的信号

### 什么是PWM?

PWM是脉冲宽度调制 (Pulse Width Modulation) 的缩写。如左图所见，是一个方波信号，脉冲宽度调制的大致意思是我们可以将数字信号转化位一个特定占空比 (duty) 的方波信号（占空比是指在一个脉冲循环内，通电时间相对于总时间所占的比例。通电即高电平，不通电即低电平，可理解为高电平占整个周期的比例）。

### 如何利用PWM实现呼吸灯呢?

我们知道，一个灯的亮度取决于输入灯功率的大小。输入的功率越高，灯就会越亮，反之就小。我们还知道，在一个周期内，通电时间越长（占空比，duty），该周期内的平均功率就越大，反之就小。

那么，通过控制一个周期内灯的通电时间，不就可以控制灯的明亮程度了嘛?

根据以上思路，我们设计代码的思路：



## 案例二 舵机控制!

Case two: control of servo

### 舵机控制原理

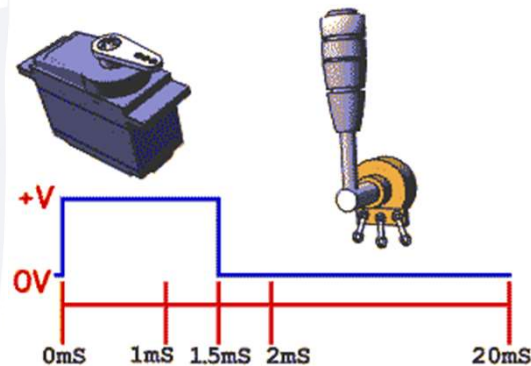
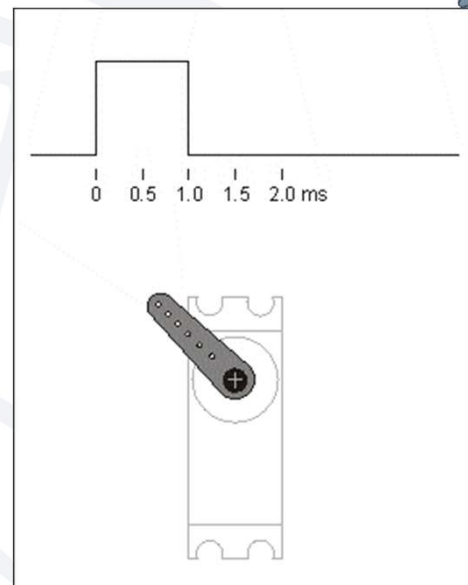
一般通过PWM控制舵机转动的角度。舵机一共有三个线，分别是电源线（VCC），地线（GND），信号线（PWM控制输入）。

控制信号PWM频率固定设置为50Hz，可知方波的周期为1/50s(20ms)。舵机转动角度的控制是通过控制PWM的占空比实现的。它们的对应关系如下（此处多级型号为SG90，可转动角度为180°）：

单周期内高电平时间

0.5ms	——	0度
1.0ms	——	45度
1.5ms	——	90度
2.0ms	——	135度
2.5ms	——	180度

舵机角度



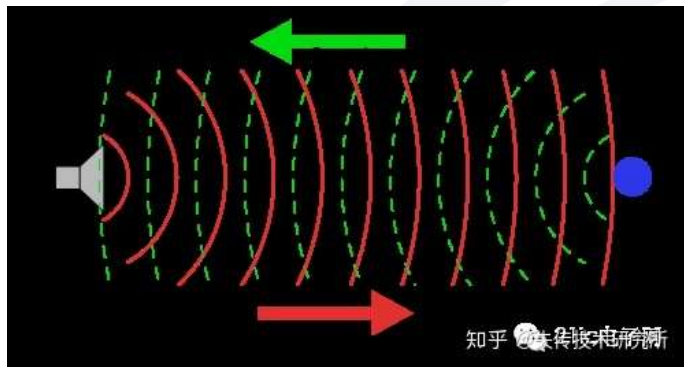
[\(29条消息\) 舵机控制原理和结构\\_zeruns-CSDN博客\\_舵机控制原理](#)

## 案例三 超声波测距

Case three: ultrasonic ranging!

### 超声波测距原理

如右图，就是声波测距原理。但是我们需要控制超声波模块进行测距，Pico怎么利用超声波模块进行测距呢？我们可以将该过程分为三步，**发送测距信号**，**接收测量信息**，**利用信息计算距离**。超声波模块一共有四个引脚，分别是**VCC(电源)**，**echo**，**trig**，**GND(地线)**。具体如何控制如下：



#### 发送测距信号

如何告诉超声波模块，现在需要测量距离了呢？  
在trig(触发引脚)产生一个10us的高电平即可。

#### 接收测量信息

超声波模块如何告诉Pico它的测量结果呢？  
超声波测量完成后，会在echo(回应)产生一个高电平，高电平的时长等于从发送声波道接收反射波的时间。

#### 利用信息计算距离

如何根据超声波提供的信息计算出物体的距离？  
我们知道声音在室温下的一般速率为 $v=340\text{m/s}$ ，超声波模块为我们提供了声波的往返时间 $t$ 。你能写出距离 $d$ 的计算公式吗？

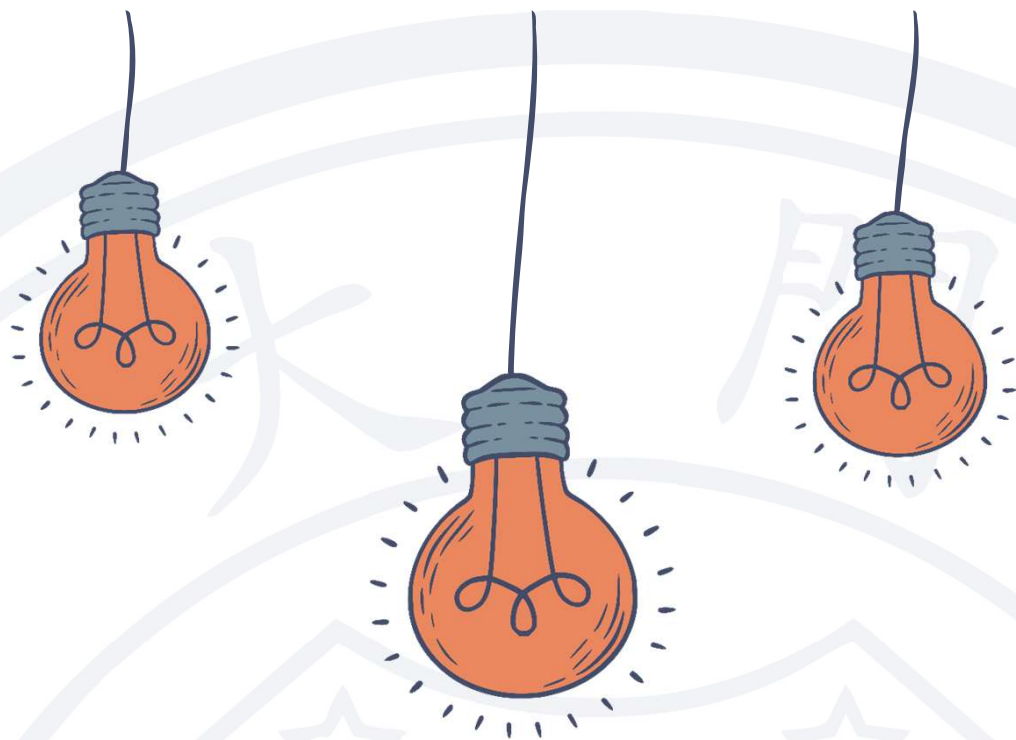
顺序结构

## 问卷内容



请大家及时填写





感谢您的支持与关注



## 补充内容：Python安装模块

### 第一步 添加Python的环境变量

什么时环境变量？

我们可以理解为系统运行时，经常需要运行的程序的路径。我们专门用一张表去记录那些程序的路径，这样我们就可以很容易的找到他们。我们需要将python添加到这个列表里，才能在终端快捷的使用Python程序。

如何添加Python环境变量？

[安装更正](#)

### 第二步 安装python模块

安装工具：pip

pip是python官方的包管理器，我们可以通过它轻松的使用别人写好的模块，站在巨人的肩膀上。

使用方法：

```
pip install [包名]
```

示例：安装numpy模块

```
pip install numpy
```