信息技术

本节内容

- 这是一门什么样的课?
- 这门课的成绩如何评定?
- 你将会在这门课学到什么?

课程介绍

信息技术是一门必修学分课程。

- 课程成绩会进入成绩单
- 部分同学需要参加明年6月举行的合格考

课程包括三个模块

- Python程序设计（50%）
- 计算机理论知识（30%）

  信息编码，计算机硬件，网络原理...
- 计算机应用实践（20%）

  图像处理，数据库，游戏设计，网页设计，人工智能...

成绩评定

成绩占比

- 课堂表现（10%），平时作业（60%），期末大作业（30%）

作业非常重要

- 每周作业会在周五晚上公布，下周三晚上10点前提交
- 作业可以反复提交，以最后提交的作业为准
- 错过截止时间后果很严重
  - 周四中午12点前提交：60%分数
  - 周四中午12点后提交：0分

参考教材

- 教科书：《信息技术》（华东师范大学出版社）
- 课程网站：https://cs2023.readthedocs.io/
- 课件和作业每周五在课程网站上发布

编程工具

- Replit: 可以在浏览器上完成复杂的编程
- 需要注册Replit账号并加入班级
- 编程作业在Replit上进行提交

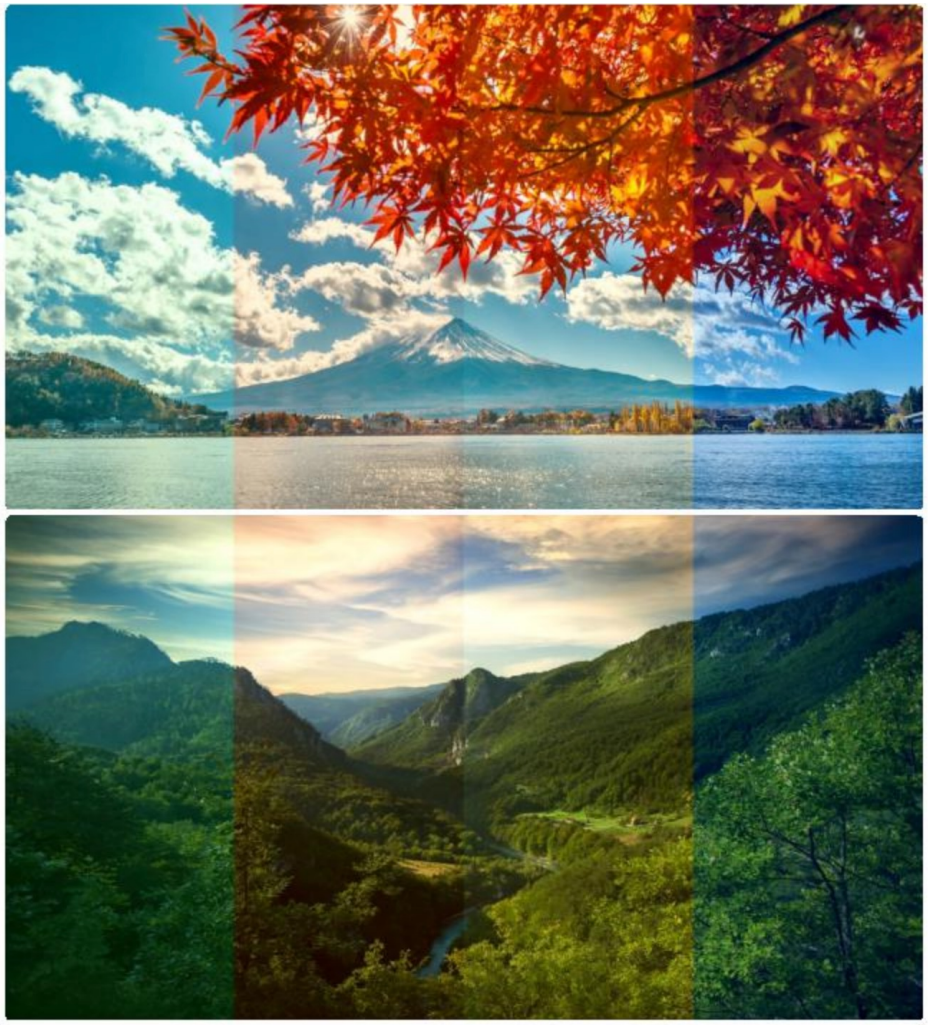
说在之前的话

- Knowledge vs. Skill
- 准备好进行大量的练习
- 多练简单的问题
- Google is your best friend.

部分同学作业

用Python设计一个图像滤镜

朱镜轩

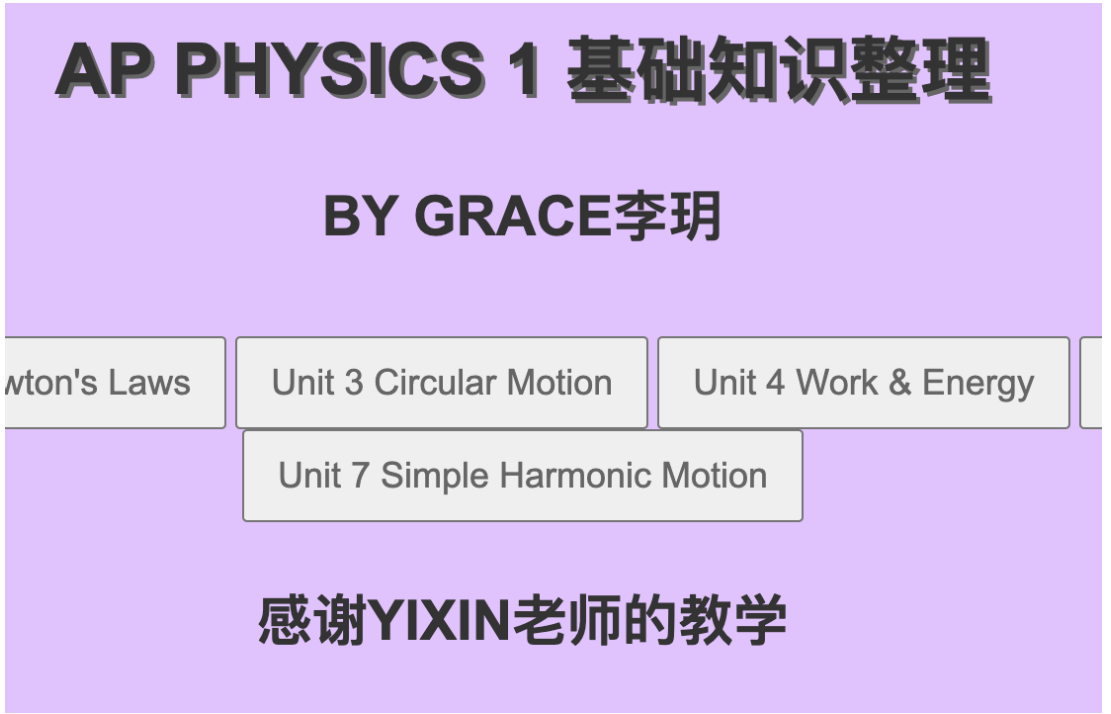


陈楷焜

部分同学作业

用html和css设计自己的网页

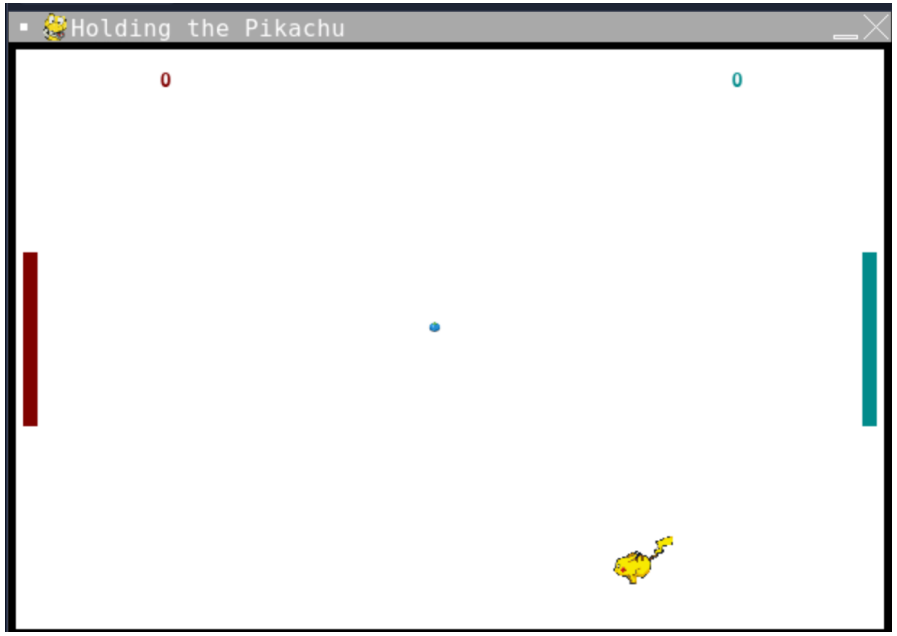
费沁沄



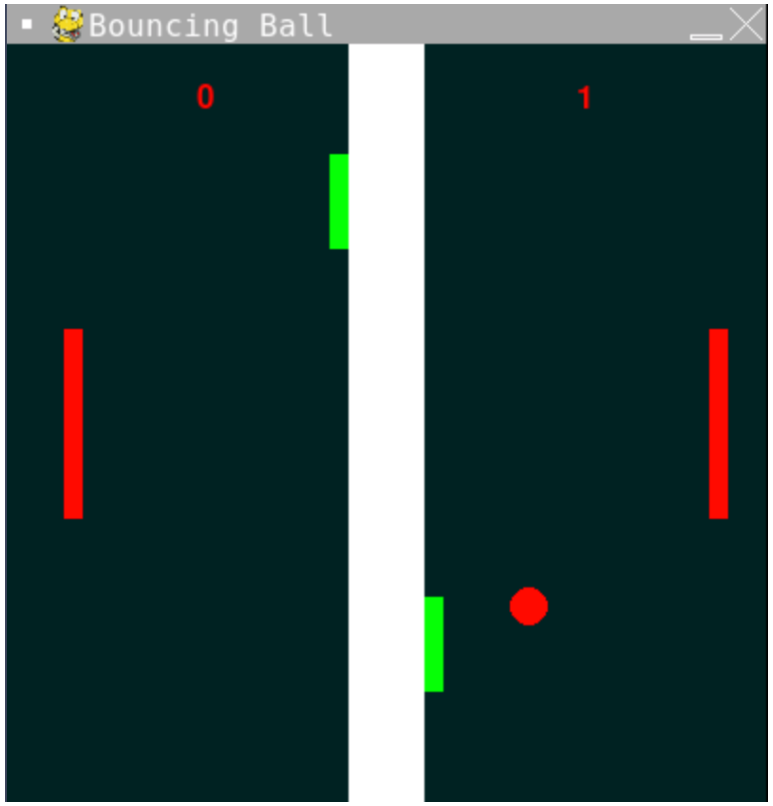
李玥

部分同学作业

设计一款乒乓游戏



袁子皓



倪嘉辰

来自学姐的的大作业总结

一学期来，我在python的学习上并不顺利是有目共睹的，但是我在期末作业中，居然成功写出了所有的题目，希望胡老师您也能为我感到欣慰，因为我自己非常为我骄傲。

在接触编程的初期，我时时被难倒。语法，据，函数，循环，充满严谨逻辑与严苛规则的编程语言，冲击着我的大脑。甚至，某段时间我最害怕的,不是数学作业，而是信息作业。看似我在锲而不舍地问和学，其实我内心依旧很害怕和抵触编程。尤其是周围的同学都觉得上信息课很轻松，而我还在被简单的问题困扰。

但是这次的期末大作业，是一个全新的动力。更多的任务和压力，让我不得不想尽办法和python合作，摩擦。在大量完成任务的工程中，我奇迹般地发现，很多问题我可以自己解决，很多思路我可以自己得出。从一开始花半个多小时研究老师的示例，到-次成功地设计出正确的函数，越来越快，越来越好。别人可能不相信，在看到run按钮停止加载，崭新的成品跳出在pythonProiect文件夹里，我的心可以因狂喜而激动地跳好久。

我慢慢开始思考，编程是否并不是洪水猛兽?其实新的世界早已向我敞开大门，而我却束手束脚，逃避走出舒适圈。

完成所有的期末任务，在以前看来，我是不可能做到的。但是除去自己吓自己的环节，克服一遍遍的错误带来的挫折感，抛弃无用的畏缩与自卑、我发现，耐心与细心常伴，高峰终将被征服。希望我可以带着这样的令人自豪的勇气，不断地进步。

还记得学期初，胡老师问我到底是编程难还是托福难，我毫不犹豫地回答: 编程难。现在看来，其实两者是非常相似的。不熟悉的代码和语法，在多看几遍，多敲几行之后也渐渐熟练了。这让我回想起了刚开始学写英语作文时，打字非常缓慢，单词常常打错。但是过了一周，我就能和打中文一样打出英文了。编程和英语，都是一门语言，只是看似存在于两个世界中，本质上都是人们思想的输出方式，只不过一个是向电脑说，一个是向人类说。如果从小我们说的，用的，写的，背的都是python语，我的答案绝会转变:“托福难!”故而，学习万物皆有过程，没必要因一时的失利而气馁。接下来的半个学期，我将调整心态，再接再厉!

什么是程序?

计算机也是机器!

- 必须设置好才能运行!
- “编程序” = 给计算机设定好运行的步骤

程序

- 人们用来告诉计算机应该做什么的东西

那么告诉计算机哪些东西，可以让它听懂呢?
以什么形式告诉它，它才能够明白?

问题一

告诉计算机哪些东西，计算机才能够运行呢？

给你一个数列：

78, 56, 69, 31, 36, 67, 31, 47, 69, 34, 45, 74, 61, 82, 43, 41, 76, 79, 81,
66, 54, 50, 76, 51, 53, 28, 74, 39, 45, 61, 52, 41, 43, 75, 78, 84, 72, 51,
43, 64, 75, 81, 69, 55, 74

哪个数字最大？

你会怎么思考这个问题？

你是怎么做的?

哪个数字最大?

- 把某一个数字取出来，当作一个临时的“特别数字”记住，并假设这个数字最大;
- 拿这个临时的“特别数字”与其他数字相比较;
- 如果有其他数字比临时的“特别数字”更大，就把“特别的数字”换成这个更大的数字;
- 重复上述过程直到把所有的数字都比较完毕;
- 那么大脑中这个“特别数字”就记录了最大的数字;

我的大脑这样运行

把自己的大脑当作计算机

- 在大脑中开辟一片“存储空间”存放输入的数字；
- 使用了另一片“存储空间”存放“特别数字”；
- 按照某种规律“反复”选定“输入存储空间中的数字”与“特别数字”比较；
- 每次比较时，判断“选定的数字”是否大于“特别数字”；
- 如果大于，重新“刷新” “特别数字”；
- 如果“特别数字”与其他数字都进行了比较，说出“特别数字”；

把这个过程稍作整理

更清楚的描述方式

- 在你的大脑里开辟一片存储空间存放输入的数字；
- 开辟另一个存储空间存放“特别数字”；
- 从存储空间中的第一个数字开始，直到最后一个数，重复以下操作：
  - 比较“存储空间中的数字”与“特别数字”；
  - 如果“存储空间中的数字”大于“特别数字”；
  - 那么，将“特别数字”换成“存储空间中的数字”；
- 说出“特别数字”；

Python代码

```
array = [78, 56, 69, 31, 36, 67, 31, 47, 69, 34, 45, 74, 61, 82, 43, 41,
76, 79, 81, 66, 54, 50, 76, 51, 53, 28, 74, 39, 45, 61, 52, 41, 43, 75, 78,
84, 72, 51, 43, 64, 75, 81, 69, 55, 74]

max = array[0]

for i in range(1, len(array)):
    if array[i] > max:
        max = array[i]

print("The maximum number in the array is: " + str(max))
```

提个要求

如果你要创造一门程序设计语言，你将如何回答以下的问题?

- 问题1：是不是无论我们在程序里写什么“单词”，计算机都可以听明白呢?
- 问题2：是不是无论我们在程序里写什么“数字”和“计算符号”，计算机都能听明白呢?
- 问题3：世界上可以用程序表达的逻辑纷繁复杂，程序设计语言里需要有多少种“句式”才够用呢?

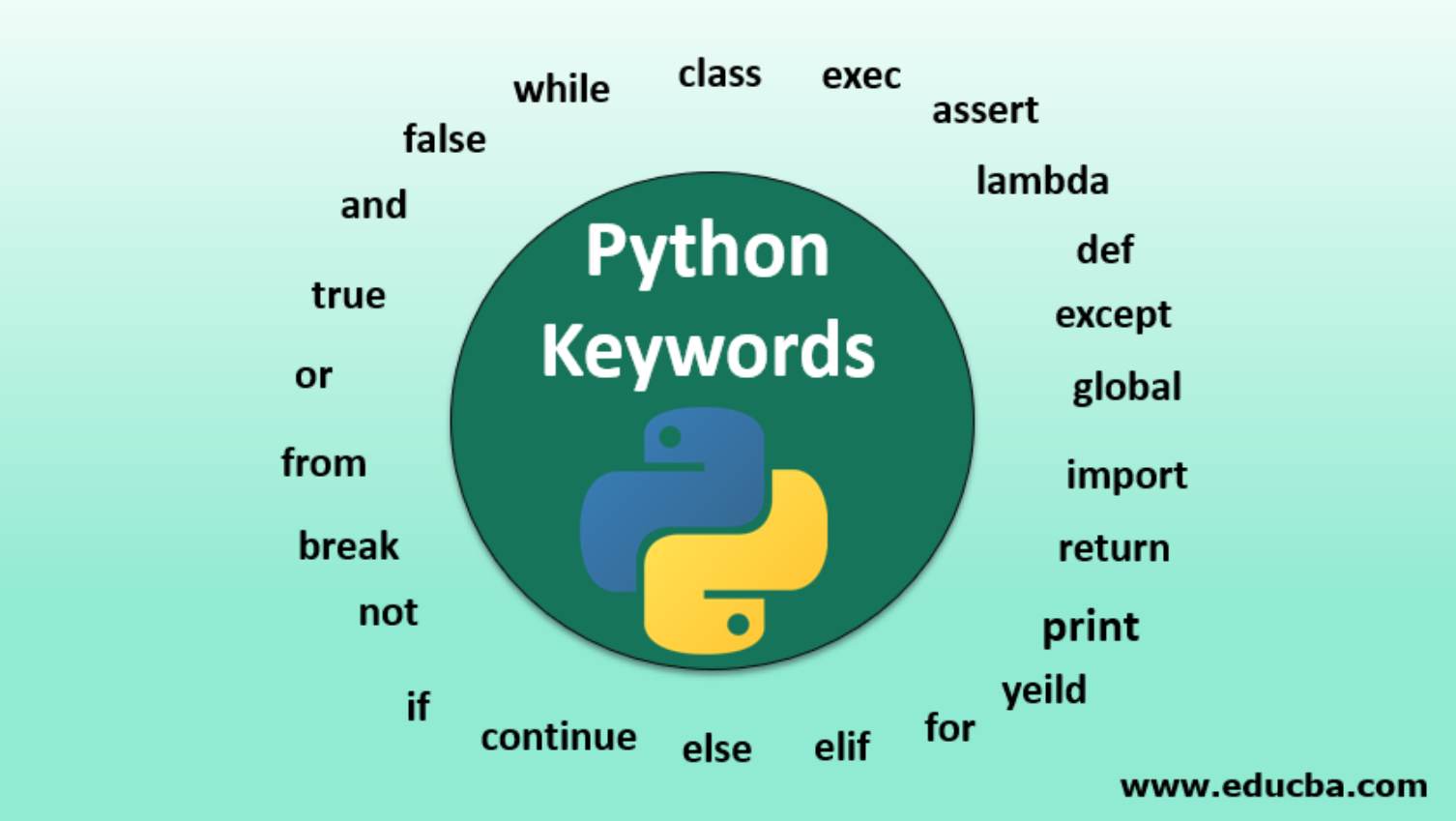
问题1

是不是无论我们在程序里写什么“单词”，计算机都可以听明白呢?

- No!
- 编程语言中定义了一些特定含义的“关键词”，计算机只能“明白”这些“单词”的含义。

计算机能够认识的单词

这些单词称为：关键词(keyword)



不需要记下来

关键词在编译器中会高亮显示

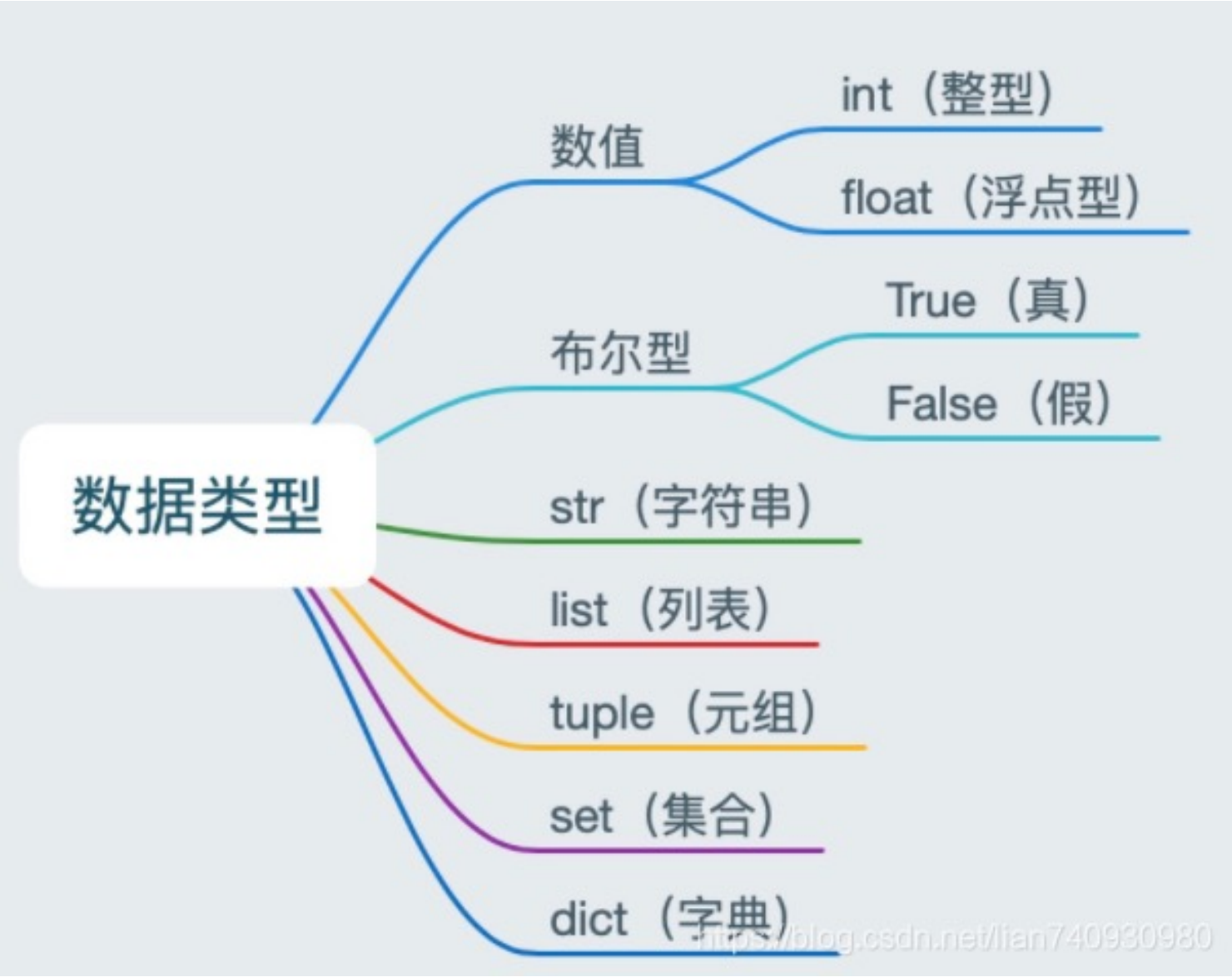
```
array = [78, 56, 69, 31, 36, 67, 31, 47, 69, 34, 45, 74, 61, 82, 43, 41, 76, 79, 81, 66,
  54, 50, 76, 51, 53, 28, 74, 39, 45, 61, 52, 41, 43, 75, 78, 84, 72, 51, 43, 64, 75, 81,
  69, 55, 74]

max = array[0]

for i in range(1, len(array)):
    if array[i] > max:
        max = array[i]

print("The maximum number in the array is: " + str(max))
```

问题2

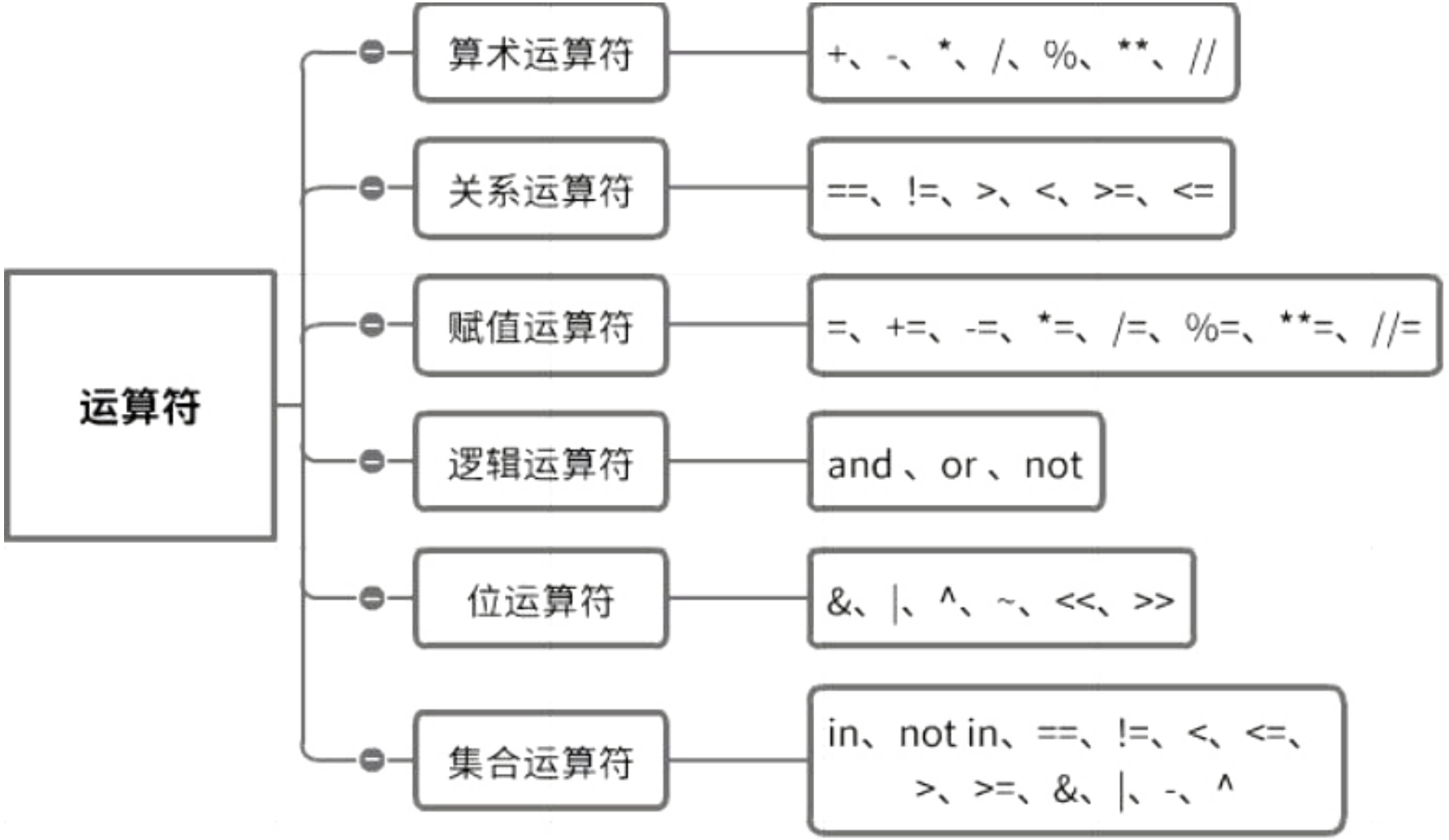
是不是无论我们在程序里写什么“数字”和“计算符号”，计算机都能听明白呢?

- No!
- 计算机只能看懂某些类型的数字，这些“数据类型”和相应的“计算符号”也是定义好的。

计算机能够看懂的数据类型



计算机能够看懂的运算符号

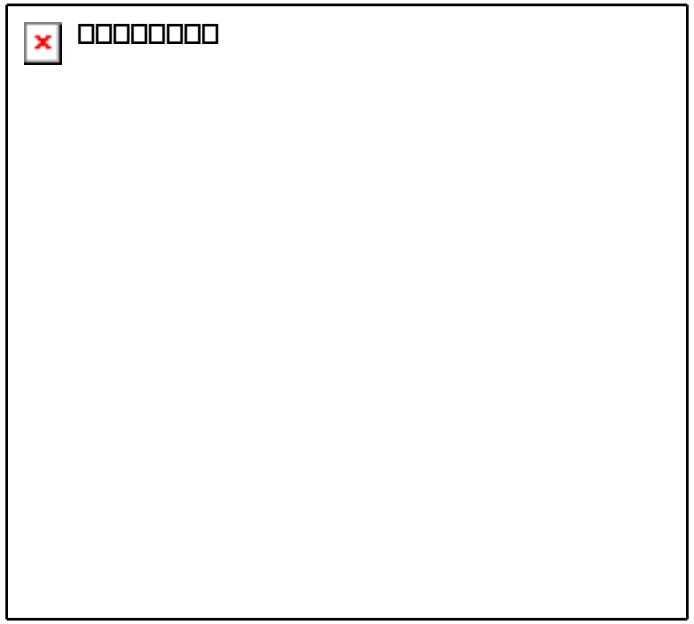


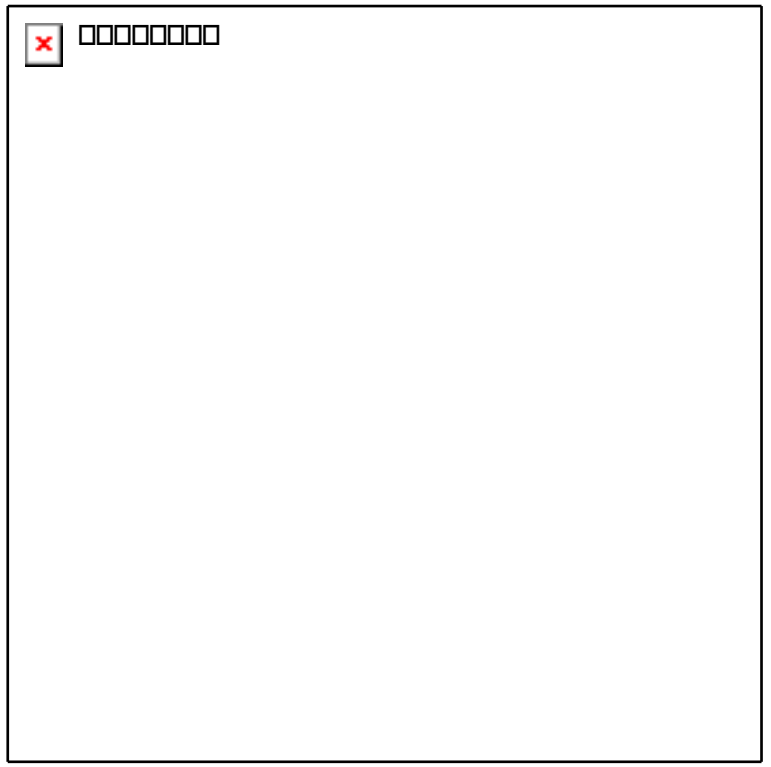
问题3

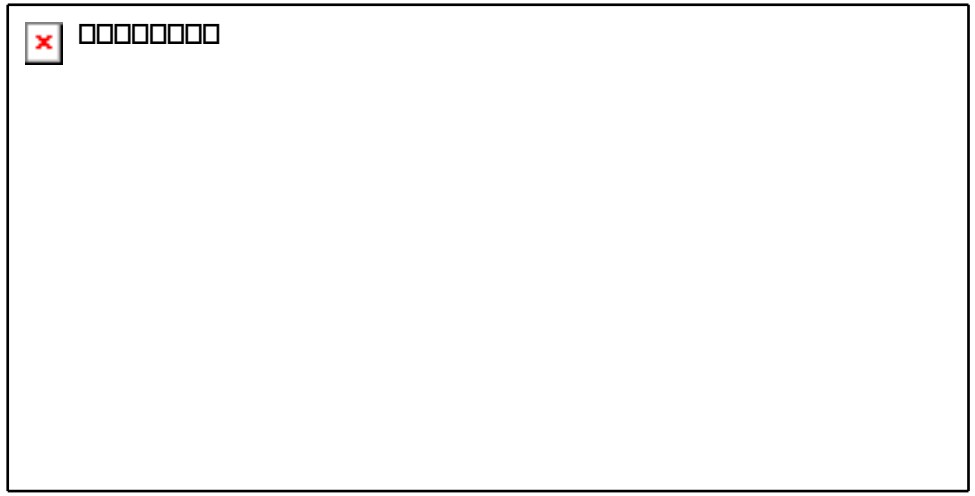
世界上可以用程序表达的逻辑纷繁复杂，程序设计语言里需要有多少种“句式”才够用呢?

- 不多！三种而已!
- C. Bohm& G. Jacopini, "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules," Communications of the ACM, vol9(5) May 1966, pp 366-371.

如何表达纷繁复杂的逻辑







分支语句

我们要学的编程语言Python

我们只需要学习：

- 20几个关键字
- 6种数据类型 + 10几个运算符号
- 三种基本逻辑语句

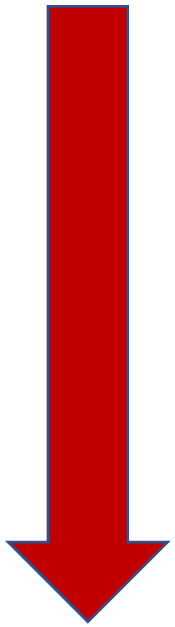
第一个Python程序

```
a = 1


b = 2


a = (a+1)*(b+2)


print(a)
```

你有哪些观察?

运行规则：由上往下依次运行

```
a = 1  #将1赋给a变量


b = 2


a = (a+1)*(b+2) #将运算表达式的结果赋给a变量


print(a)  #将变量a的值打印在屏幕上
```



你有哪些观察？

变量：用来存放数据

```
a = 1  #将1赋给a变量

b = 2


a = (a+1)*(b+2) #将运算表达式的结果赋给a变量


print(a)  #将变量a的值打印在屏幕上
```

完整解释：声明一个变量a，将1赋值给该变量

= 是赋值符，赋值顺序是将右边的值赋给左边的变量

执行完之后a和b的变量值分别为多少?

变量：用来存放数据

```
a = 1  #将1赋给a变量

b = 2


a = (a+1)*(b+2) #将运算表达式的结果赋给a变量


print(a)  #将变量a的值打印在屏幕上
```

完整解释：声明一个变量a，将1赋值给该变量

= 是赋值符，赋值顺序是将右边的值赋给左边的变量

执行完之后a和b的变量值分别为多少？

a的值变为8， b的值不变

变量使用之前要先声明并赋初值

```
b = 2

a = (a+1)*(b+2) #将运算表达式的结果赋给a变量

print(a)  #将变量a的值打印在屏幕上
```

删掉a = 1这一行

出现报错!

print()：打印括号的值

```
a = 1  #将1赋给a变量


b = 2


a = (a+1)*(b+2) #将运算表达式的结果赋给a变量


print(a)  #将变量a的值打印在屏幕上
```

print()是调试程序的一个重要部分

当程序运行结果错误的时候，要多用print()进行检查!

编程工具

我们使用两个工具进行编程

- Replit：浏览器编程，无需安装软件
- PyCharm：本地计算机编程，需要安装软件（暂时不需要）

90%的作业会在Replit上进行提交

Replit注册

请按照下面步骤操作：

- 打开replit.com，点击邀请链接
- 注册账号
  - continue with emails，用学校邮箱
  - FirstName和LastName，请填写真实的拼音姓名

注册完之后你可以看到第一个课堂作业！