
欢迎！

（下载幻灯片和 .py 文件并跟随！）

6.0001 讲座 1

今天

课程信息

什么是计算

python基础

数学运算

python 变量和类型

注意:每节课前幻灯片和代码文件

- o 强烈建议您在讲课前下载它们
做笔记并运行代码文件
- o 带上电脑来回答课堂练习题!

课程信息

分级

- 大约20% 测验
- 大约最终 ◦ 约40%

30% 问题集 ◦ 约。 10% MITx
手指练习

课程政策

协作

- 可以与任何人合作 ◦需要独立编写代码并

写下

提交的所有合作者

- 我们将在所有 pset 上运行代码相似性程序

扩展

- 无扩展

◦**迟到**, 请参阅课程网站了解详细信息 ◦期末考试成绩
中最多两个 pset 的**跌落**重量 ◦只能在紧急情况下使用

朗诵

非强制性

两种口味

1) 讲座回顾:如果你错过了讲座,如果你需要对相同的概

- 念有不同的看法,请回顾讲座材料

-

2) 问题解决:教你如何解决编程问题

- 如果您不知道如何从 pset 单词设置伪代码,这很有用◦ 我们展示了几个更难的问题

- 引导您了解如何解决问题

- 与朗诵讲师一起头脑风暴代码解决方案

- 之后会发布解决方案

快节奏课程

定位自己以取得成功！

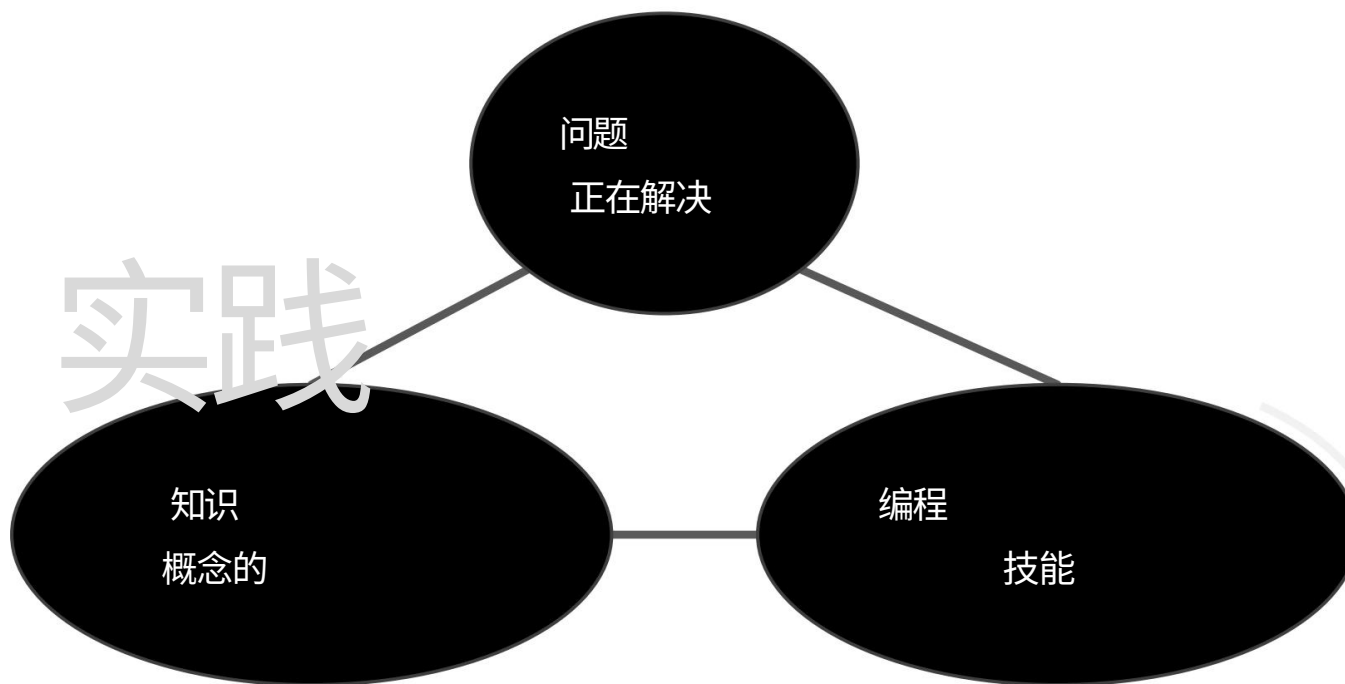
- 在 pset 出来时阅读它们,稍后再返回
- 在紧急情况下使用迟到

编程新手? **实践。实践?实践!**
能

- 讲课前下载代码并跟随
- 做 MITx 手指练习 ◦不要害怕尝

试 Python 命令!

◦不能被动地将编程当成一种技能



话题

用数据结构表示知识

迭代和递归作为计算隐喻

过程和数据类型的抽象

使用对象类和方法组织和模块化系统

不同类别的算法、搜索和排序

算法的复杂性

计算机做什么

从根本上说： ◦每
秒执行十亿次**计算!** ◦**记**
住结果

100GB 的存储空间!

什么样的计算? ◦语言**内置** ◦**您定**
义为程序员的语言

电脑只知道你告诉他们什么

知识类型

陈述性知识是对事实的陈述。

- 有人会赢得 Google 课程结束前的纸板

必要的知识是一种食谱或“方法”。

- 1)学生报名抽奖
- 2)安娜打开她的 IDE
- 3)安娜在第 1 个和第 n 个响应者之间选择一个随机数
- 4) Ana 在响应者表中找到数字。优胜者！

一个数值例子

推导数 x (16) 的平方根的配方

- 1) 从猜测开始, g
- 2) 如果 $g \cdot g$ 与 x 足够接近, 停止并说 g 是回答
- 3) 否则, 通过平均 g 和 x/g 进行新的猜测
- 4) 使用新的猜测, 重复过程直到足够接近

G	克*克	x/g	$(g+x/g)/2$
3	9	$16/3$	4.17
4.17	17.36	3.837	4.0035
4.0035	16.0277	3.997	4.000002

什么是食谱

- 1) 简单步骤的顺序
 - 2) 指定每个步骤何时执行的控制流程
 - 3) 确定何时停止的方法
- 1+2+3 = 一个算法!

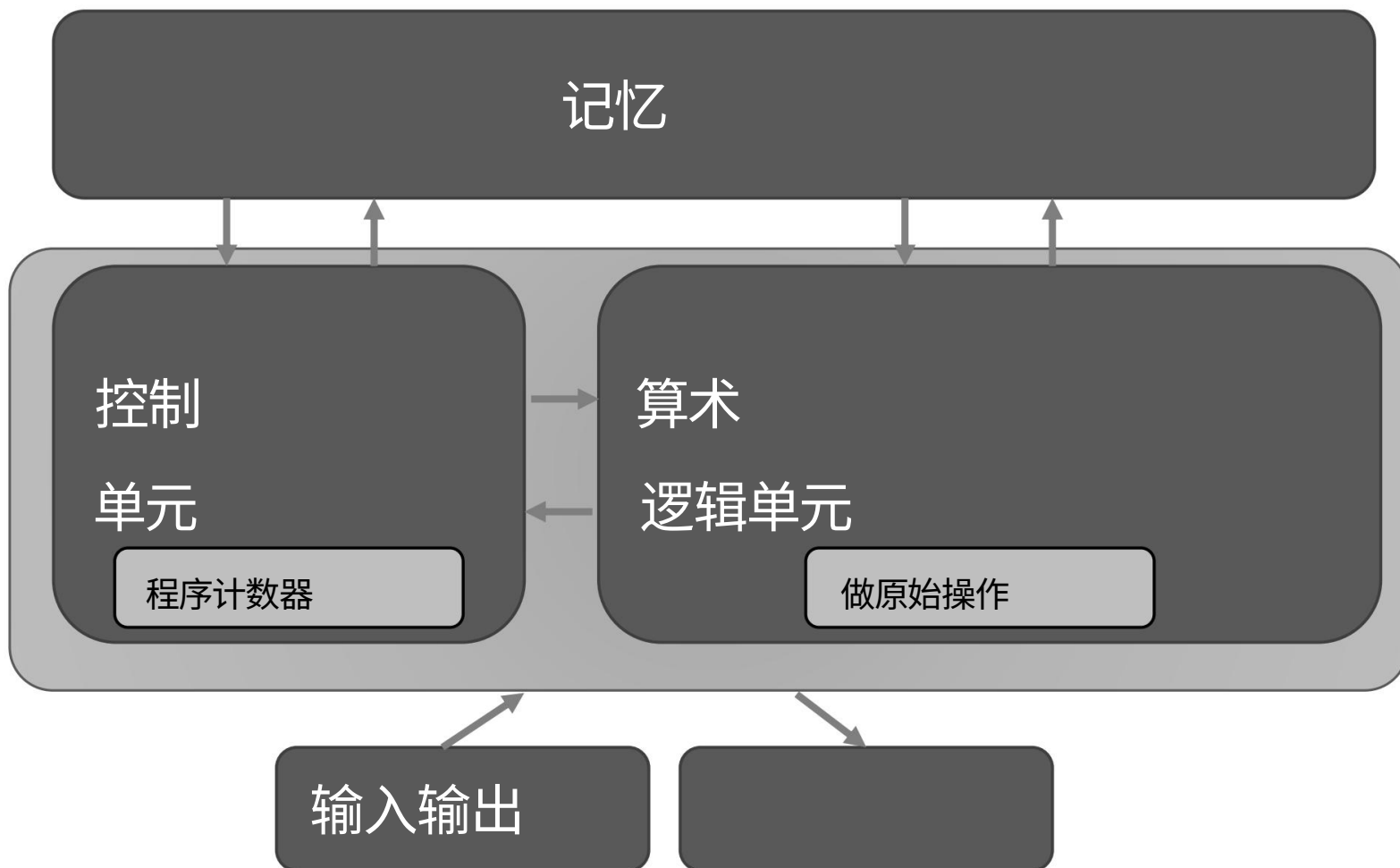
计算机就是机器

如何在机械过程中捕捉配方

固定程序计算机 ◦ 计算器

存储程序计算机 ◦ 机器存储和
执行指令

基本机器架构



存储程序计算机

存储在计算机内部的指令序列

- 从预定义的原始指令集构建

- 1) 算术和逻辑

- 2) 简单测试

- 3) 移动数据

特殊程序（解释器）按顺序执行每条指令

- 使用测试通过序列改变控制流

- 完成后停止

基本原语

图灵表明你可以使用 6 个原语**计算任何东西**

现代编程语言有更方便的原语集

可以抽象方法来**创建新的原语**

任何可以用一种语言计算的东西都可以用任何其他编程语言
计算

创建食谱

一种编程语言提供一组原始**操作**

表达式是编程语言中原语的复杂但合法的组合

表达式和计算在编程语言中具有**值**和含义

语言方面

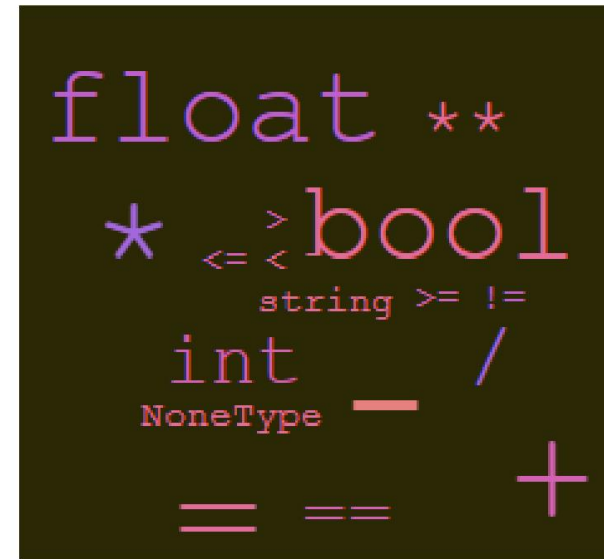
原始结构 · 英语:单词 · 编程

语言:数字、字符串、简单

运营商



Word Cloud 版权所有Michael Twardos,保留所有权利。此内容不包含在我们的知识共享许可中。有关更多信息,请参阅<https://ocw.mit.edu/help/faq-fair-use/>。



词云版权未知,保留所有权利。

此内容不包含在我们的广告素材中
公共许可。有关更多信息,请参阅<https://ocw.mit.edu/help/faq-fair-use/>。

语言方面

句法 ○英

语：“cat dog boy”

效

句法无效 “cat hugs boy”

句法有

○编程语言：“hi”5

语法无效

3.2*5

语法有效

语言方面

静态语义是语法上有效的字符串有意义

◦英语：“我饿了” 语法有效

但静态语义错误

◦编程语言 :3.2*5 语法有效

3+ hi 静态语义错误

语言方面

语义是与语法正确的符号串相关联的含义,没有静态语义错误

- 英语:可以有多种含义 “飞行飞机可能很危险”
- 编程语言:只有一种含义,但可能不是程序员想要的

哪里出了问题

句法错误 ◦ 常见且
容易被发现

静态语义错误

◦ 某些语言在运行程序之前会检查这些 ◦ 可能导致不可预知的行为

没有语义错误,但含义与程序员的意图不同 ◦ 程序崩溃,停止运行
◦ 程序永远运行 ◦ 程序给出了答案,但与预期不同

蟒蛇程序

程序是一系列定义和命令

- 评估的定义
- 由 Python 解释器在 shell 中执行的命令

命令 (语句)指示解释器做某事

可以直接在shell中键入或存储在读入 shell 并评估的文件中

- 问题集 0 将向您介绍 Anaconda 中的这些

对象

程序操作数据对象

对象有一个类型,它定义了程序可以对它们做的事情的种类

- 安娜是人类,所以她会走路、说英语等。
- Chewbacca 是个 wookie,所以他会走路,“mwaaarhrhh”等。

对象是

- 标量 (不能细分)
- 非标量 (具有可访问的内部结构)

标量对象

int – 表示整数,例如。 5

浮点数 代表实数,例如。 3.27

bool – 表示布尔值 True 和 False

NoneType – 特殊且只有一个值,无

可以使用 type() 来查看对象的类型

```
>>> type(5) 整数
```

```
>>> type(3.0) 浮点数
```

*what you write into
the Python shell*

*what shows after
hitting enter*

类型转换（铸造）

可以将一种类型的对象转换为另一种类型

`float(3)` 将整数 3 转换为浮点 3.0

`int(3.9)` 将浮点数 3.9 截断为整数 3

打印到控制台

向用户显示代码的输出,使用打印命令

在[11]中:3+2

出[11]:5

在 [12] 中:打印 (3+2) 5

*"Out" tells you it's an
interaction within the
shell only*

*No "Out" means it is
actually shown to a user,
apparent when you
edit/run files*

表达式

将对象和运算符组合成表达式

一个表达式有一个值,它有一个类型

简单表达式的语法

<对象> <操作员> <对象>

整数和浮点数的运算符

$i+j$ 总和

$i-j$ 差异

$i*j$ 乘积 i/j 除法

如果两个都是整数,结果是整数 如
如果一个或两个都是浮点数,结果是浮点数

结果是浮动

$i\%j$ i 除以 j 的余数 $i**j$ i 的 j 次方

简单的操作

括号用来告诉 Python 先做这些操作

无括号的运算符优先级

- $**$
- $*$
- $/$
- $+$ 和 $-$ 从左到右执行,如表达式中所示

绑定变量和 价值观

等号是给变量赋值

姓名

variable *value*

圆周率 = 3.14159

pi_approx = 22/7

值存储在计算机内存中

赋值将名称绑定到值

通过调用名称检索与名称或变量关联的值,输入 pi

抽象表达式

为什么给表达式的值命名？

重用名称而不是值

以后更容易更改代码

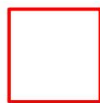
pi = 3.14159 半径 =
2.2 面积 =

$\text{pi} * (\text{radius} ** 2)$

编程与数学

在编程中,你不会“求解 x”

```
pi = 3.14159  
radius  
= 2.2 # 圆的面积 area  
= pi*(radius**2)  
radius =  
radius+1
```



an assignment
* expression on the right, evaluated to a value
* variable name on the left
* equivalent expression to `radius = radius + 1`
is `radius += 1`

更改绑定

可以使用新的赋值语句**重新绑定**变量名

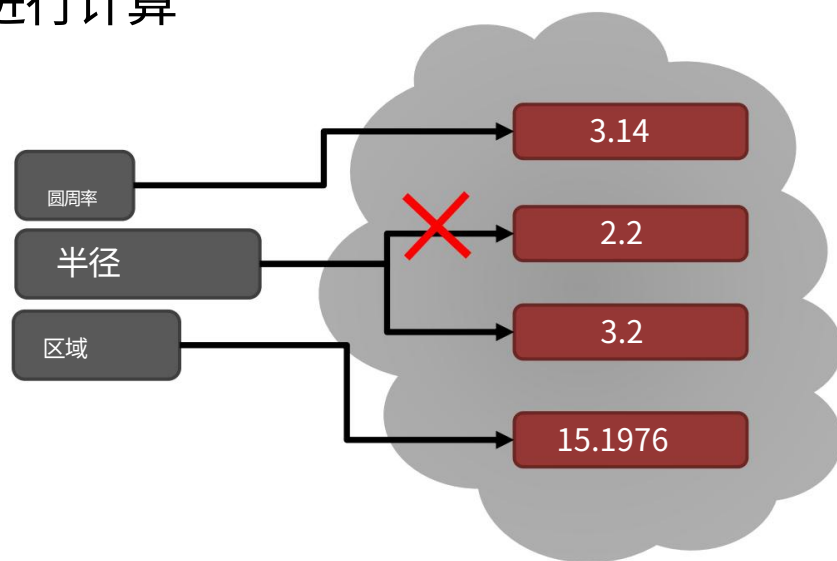
以前的值可能仍存储在内存中,但丢失了它的句柄

面积值不会改变,直到您告诉计算机再次进行计算

$\pi = 3.14$ 半径 =

2.2 面积 =

$\pi * (\text{radius} ** 2)$ 半径 = 半径+1



麻省理工学院开放课件[https://
ocw.mit.edu](https://ocw.mit.edu)

6.0001 计算机科学和 Python 编程简介
2016 年秋季

有关引用这些材料或我们的使用条款的信息,请访问: <https://ocw.mit.edu/terms>。