python基础教程_01_python基础

woniuppp

选择即改变

开发的路其实不好走

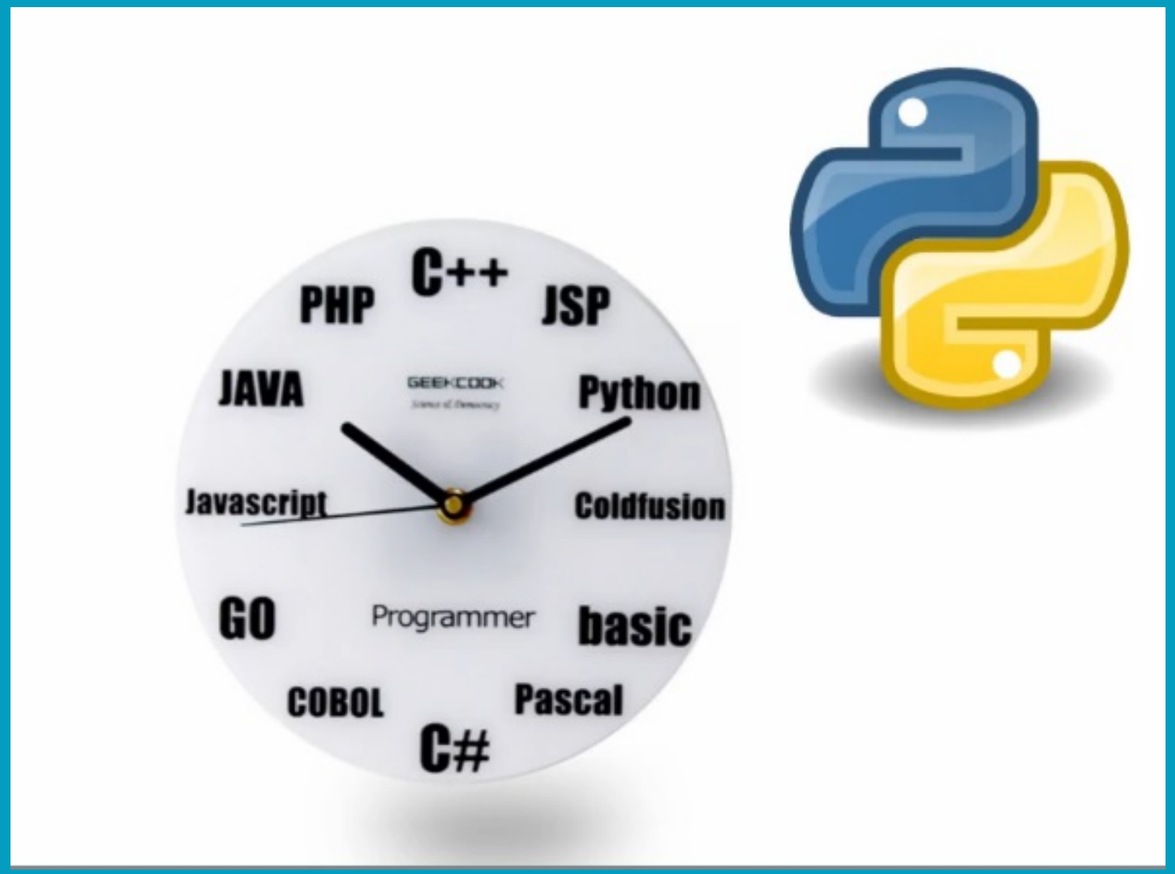
- 编程本身就很难
- 周六来听课很辛苦
- 来了reboot，一定有收获

为什么要学编程

传统运维(测试，技术支持)现状

- 一般运维做两三年，会有瓶颈
- 大公司运维的现状
- 云的发展，只运维，不会开发，饭碗被抢
- 容易被开发绑架
- 开源软件也需要定制性开发

为什么要学python



语言对比

语言	优点	缺点
C/C++	性能,游戏,底层	代码量大，上手难
java	明星级语言，比较成熟	不太好上手，代码量大，开发速度慢
.net/c#		过时了
shell	简单的系统维护	不能算是一门编程语言
php	易学，脚本	主要适合web
python	易学，脚本，功能强大	运行速度比java慢
javascript	易学，前端	前端脚本语言

为什么要学python

我们需要的语言是什么样的

- 上手简单
- 功能健全
- 语言生态系统完善，第三方库多
- 有大公司成功的案例

hello python

- 上手简单
- 脚本 游戏 图形化 web 爬虫 你能想到的方方面面
- github上 python项目足够多
- google FB dropbox 豆瓣 知乎 bat
- 运行速度不是瓶颈

python的历史--蛇叔的情怀

龟叔 Guido van Rossum 荷兰人

python的历史

- 有一个任务，知道用C怎么完成功能，但是代码量太大
- shell太简单，不能完成复杂的功能
- 为了打发圣诞节假期，Guido开始写自己的语言
- Guido是Monty Python（一个喜剧马戏团）的脑残粉，于是命名python（大蟒蛇）
- 个人电脑和internet开始发展，硬件不再是瓶颈，开发速度快成为优势

实战课程目标

- python的基础，掌握基本的编程思想，具体两个任务
- access_log日式处理
  - url，ip，访问状态维度，统计访问次数
  - 排序，打印出访问次数最多的前10
- 简单的cmdb
  - 基于flask
  - 数据库mysql
  - 前端jquery+bootstrap
  - 简单粗暴的完成最简单的增删改查

学习编程的方法

学编程分为两个部分

- 编程的思想
  - 怎么用编程的思路考虑问题
- 具体的编程语言去实现
  - 我们选的python

入门学习编程的方法

- 慢一点 多理解
- 多练习，记笔记
- 把编程当作工具，去解决问题，编程是手段，不是目的
- 碰到问题google baidu
- 申请github账号，加入开源社区

选择版本-安装

2.7和3.0

- 2.7和3.0不兼容，选择2.7
- linux都是自带2.7的
- windows下载安装包安装即可

第一个python程序

```
python
>>>print 'hello world'
hello world
```

尝试输入其他东西

```
>>>2+3
5

>>>print 'hello'+'python'
hellopython

>>>1+2*4
9

>>>(+2)*4
12
>>>exit()
```

更进一步

写的文件想保存下载

- 交互式适合学习，不能保存
- 新建一个文件,以 hello.py结尾
- 选择一个编辑器，在hello.py文件里输入下面内容并保存

```
print 'hello word'
```

- 在文件所在路径，执行 python hello.py

关于编辑器

不要纠结，学习的时候，哪个顺手就用那个

- vim 强大学习成本高
- emac 强大学习成本高
- sublime 轻量级，界面漂亮
- nodepad++
- eclipse pycharm等等
- 就像学武术一样，入门不要纠结于木剑还是倚天剑

vim存活教程

- vim hellp.py
- 一开始不能编辑，输入i 可以编辑
- 点击esc，回到最初的不能编辑状态
- 不能编辑的状态输入 :wq 保存并且退出文件
- 把精力放在目的上，而不是工具
- 入门阶段，不要在意那些细节

内容简介

- 变量 语句
- python数据类型
- 四则运算
- 获取用户的输入
- 流程控制
- 循环

编程的思路

一个小栗子--做饭

- 把盐，放到盐罐子里
- 然后拿一些番茄和鸡蛋，放盐，放到锅里炒
- 如果有特殊的重口，放点辣椒
- 最后到炒熟为止---每30秒看一下

变量

- 定义一个变量，相当于在程序里挖坑，可以放数据
- 定义完变量，供以后使用
- 相当于起外号

```
x ='salt'
print x
```

- 上面这个语句，相当于拿了一个罐子X，把'salt'放里面，等我们需要用盐的时候，直接把x这个罐头拿来用就行

语句

语句，就是做了一件事，一个行为

- 刚才的x=’salt’ 就是一个语句
- 做了一件事，把salt赋值给x
- print也是一个语句

获取用户输入的小栗子

```
x = raw_input'(please input you name)'
print x
y = input'(please input )
print y
```

- input必须输入完整的python数据 字符串要带引号
- raw_input适合学习

注释

- 代码不仅是可执行的，还要可读
- 代码一行最前面加一个 # 此行内容不会被执行
- 方便调试和加上代码解释

```
#coding=utf-8
#获取输入的用户名
 x = raw_input'(please input you name)'
# print it
print x
```

数组结构之数字

she11里感受四则运算

```
>>> 2+2
4
>>> 1+2*4
9
>>> 4-2
2
>>> 1/2
0
>>> 1.0/2.0
0.5
>>> 10%3
1
```

字符串--用单引号或者双引号包起来

```
>>>print 'hello world'
hello world
>>>print "hello world"
hello world
>>>print " I'am woniuppp"
I\am woniuppp
>>>print 'hello'+' world'
hello world
>>> x = raw_input('please enter you name')
please enter you name
>>>print 'welcome ' + x
```

字符串格式化

%s占位符，代表一个字符 语法见代码

```
x ='woniuppp'
y ='man'
print 'my name is'+ x +' and I am a '+ y
print 'my name is %s and I am a %s'%  (x,y)
```

字符串格式化

```
 x ='woniuppp'
 y =180
#报错 数字不能和字符串直接想加
print 'my name is'+ x +' and I am '+ y +' cm tall'
# %s 换成%d 占位符 代表一个数字
print 'my name is %s and I am %d cm tall'%  (x,y)
```

小练习1

用户输入两个数，求平均值

- 提示 int(str)可以把字符串转成数字类型

```
print '2'+'4'
print 2+'4'
print 2+int('4')
```

流程控制

真和假 True False(首字母大写)

```
>>> 2==3
False
>>> 2<3
True
```

流程控制

and or

- A and B，‘且’，A和B都是真的时候，才为真，否则是假
- A or B，‘或’，A和B有一个真的时候，就是真，否则是假

```
>>> True and False
False
>>> True or False
True
>>> 3>2 and 4>3
>>> 3<4 or 2>4
```

流程控制

if else 语句 实现逻辑控制

```
if(判断真假):
    如果是真 执行 （缩进）
else:
    如果是假 执行
这里的语句，和if无关 都会执行
```

逻辑控制

python是用缩进来判断代码块的，建议直接四个空格

```
x = raw_input'(please enter you name)'
if x=='WD':
    print 'you are a nice boy'
else:
    print 'nice to meet you'
```

逻辑控制

多层if判断

```
 x = raw_input('please enter you name')
if x=='WD':
    print 'you are a nice boy'
elif x=='PC':
    print 'you are nice too'
else:
    print 'nice to meet you'
```

逻辑控制

if可以嵌套多层（多层缩进）

```
x = raw_input('please enter you name')
y = raw_input('please enter your age')
if x == 'WD':
    if y=='20':
        print 'nani'
    elif y == '10':
        print 'interesting'
    else:
        print 'I don"n know'
else:
    print 'not an age'
```

循环--while

一直循环执行语句,注意缩进

```
#注意缩进
while 判断条件:
    #如果判断条件是真，循环体的语句就会一直执行
      语句1
      语句2
      修改判断条件中的变量，使得循环是可以结束的
 这里的语句，和wilie无关（缩进）
```

循环--while

```
 i=0
while i<20:
    print i
     i = i +1
```

循环--while

```
 name =''
while not name:
     name = raw_input('please enter you name')

print 'hello'+name
```

小练习2

- 让用户一直输入数字
- 如果输入的是’pc’，终止程序
- 打印所有数字之和

小练习3

- 让用户一直输入数字（只输入数字）
- 如果没输入任何值，终止程序
- 打印所有输入数字的平均值
- 小提示：

```
>>> 3/2
>>> 3/2.0
>>> 1+1.0
```

小练习4

- 存10000块钱，年利率是3.25%
- 求多少年之后，存款能翻番

for循环

可以用来遍历一个序列

- 可以先把['woniuppp','WD','PC']理解为一个存好几个数据的序列

```
for name in ['woniuppp','WD','PC']:
    print name
```

```
for num in range(0,29):
    print num
```

小练习5

- 遍历一个序列 ['C','js','python','js','css','js','html','node']
- 统计这个序列中，js出现的次数

小练习6

- 一个序列

  [1, 2, 3, 2, 12, 3, 1, 3, 21, 2, 2, 3, 4111, 22, 3333, 444, 111, 4, 5, 777, 65555, 45, 33, 45]

- 求这个list的最大值

跳出循环

break语句，可以跳出循环（for，while）

- 跳出哪个循环，由代码缩进决定

```
for num in range(0,10):
    if num == 7:
        #num大于7的时候，结束整个循环
        break
    print num
```

break语句，可以当前这一次循环，从下次开始继续循环

（for，while）

```
for num in range(0,10):
    if num == 7:
        #num等于7的时候，结束当前循环，继续下一次循环，所以8.9还会打印出来
        continue
    print num
```

小练习7

用户输入数字 判断是不是闰年

- 如果是100的倍数，要被400整除
- 被4整除
- 比如1900不是闰年，2000，2004是闰年
- 如果输入不是闰年，提示信息，并且继续输入

dict简介

dict 就是key value值，索引有意义

- 和list的区别

```
# 定义dict
d = {
'name':'wd'
}
# 获取dict值
print d['name']
# 增加新值
d['age'] = 12
print d['age']
# 修改值
d['name'] = 'pc'
print d['name']
```

小练习8

- ['C','js','python','js','css','js','html','node','js','python','js','css','js','h
- 求出这个list中，每个字符出现的次数

作业

- 一个序列

  [1, 2, 3, 2, 12, 3, 1, 3, 21, 2, 2, 3, 4111, 22, 3333, 444, 111, 4, 5, 777, 65555, 45, 33, 45]
- 求这个list的最大的两个值

微信公众号

Q & A

〈Thank You!〉

Powered By nodePPT v0.9.8-3