

讲义 6: python 语言基础

从本课程开始,我们介绍服务器端的一些编程技术。服务器端编程使用了 python 与 Tornado。我们先介绍 python 语言(熟悉 python 的同学,可以飘过),然后讨论动态网页生成与客户端与服务器交互基本技术。

课程目标:

- 1) 掌握 python 基本语法与结构,使用 tornado 输出一个动态网页。
- 2) 掌握利用网络资源,解决 python 编程问题的基本方法。

使用工具:

Notepad++ or sublime 与插件;
chrome browser;
python2.7; pip 1.4; pylint; Tornado 3.1;

课程资源与要点:

- 1) 维基百科(中文):“python”,“tornado”
- 2) python 官方教程(需要 C 基础): <http://docs.python.org/2/tutorial/>
 - 2. Using the Python Interpreter: 源代码的脚本和字符集申明
 - 3. An Informal Introduction to Python: 数,字符串(处理函数,格式化 <http://docs.python.org/2/library/stdtypes.html#typeseq>), list 类型。
 - 4. More Control Flow Tools: if, for, range(), pass, 定义函数
- 3) tornado 官方文档(中文):
 - Overview: 运行 Hello world!程序,程序的基本结构;
 - Tornado 攻略一请求处理程序和请求参数:第一个案例,理解处理 URL;
 - Tornado 攻略一模板;模板结构;模板的基本使用;模板输出与转义(escape);
 - Tornado 攻略一调试模式和自动重载: debug 模式

任务 1: 了解 python 和 tornado

任务 1.1: python

谷歌“python”

选择 [Python - 维基百科,自由的百科全书]

问题 1: python 的图标是

问题 2: python 的发明者 python 是开放的语言,结合和 ... 和 ... 语言的习惯。

为什么选择 python? 和 Mat-lab 一样是主要科学计算工具,得到全球各大学,科研机构的广泛支持。积累了大量的程序包,从科学计算,人工智能,自然语言处理,到数据挖掘等等。是你大学期间必须掌握的工具!

问题 3: 如何获得帮助 python 网上编程社区丰富,几乎你能想到的所有的问题都可以找到答案,唯一的要求就是使用英文检索。

例如:你不会使用 for 语句。解决方案,谷歌“python for”。请优先选择 Stack Overflow 的答案。不会 sorting 就谷歌“python sorting”。一句话,你在和全球的学生一起使用 python,你的问题一定是他们曾经遇到的问题。

任务 1.2: tornado

谷歌 “tornado”

问题 4: tornado 的中文翻译是

选择 [Tornado Web 服务器]

问题 5: 这是一个 Facebook 开源的 web 框架。主要用于.....即时信息服务（如微薄、微信、陌陌、轻博客）等社交应用。

为什么选择 Tornado? 也许是一个错误的选择。一个相对比较原始的框架也许能够更好理解网络协议与联网信息处理。同时，它不友好的学习文档，给你带来无尽的痛苦（文档中有许多地方要求你看源代码）。

提示：快速 web 开发不是课程目标。课程目标是在 web 开发实践中，你为了解决问题，不得不去看一些源代码。学会阅读开源项目代码，并适当定制，这无疑将是你在大学读书最大受益之一。

谷歌 “python tornado source code” 或 百度 “python tornado 源代码” 都会帮你找到很多有用的资源。

如果你希望快速开发 web 项目，python 的官方列表给你更多更好的选择：

<https://wiki.python.org/moin/WebFrameworks>，例如 django，TurboGears，Zope2 等。

任务 2: 系统安装与程序开发环境

任务 2.1: 安装 python 2.7.x

谷歌 “python”

选择 [Python Programming Language – Official Website]

选择 Download 选择 2.7.x 版本，对应的操作系统安装包，下载并安装。

任务 2.2: 测试安装

选择 DOCUMENTATION -> Python2.x -> Tutorial -> 2. Using the Python Interpreter

阅读 2.1 段内容。Windows 环境操作如下；

打开控制台，运行：

```
set path=%path%;C:\python27
python
```

显示

```
C:\Documents and Settings\panml>python
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
```

注意：这个文档是面向专业人员的，要求你有 C++ 的基础。argv, try 等不知道，请从非专业人员学习入口进入。

任务 2.2: 安装 pip 1.4

谷歌 “python pip”

选择 [pip 1.4.1 : Python Package Index]

按 Install howto 指示，首先下载 ez_setup.py，并运行它。假设 ez_setup.py 在 d:\downloads

d:

```
cd \downloads
python ez_setup.py
```

完成 setuptools 工具安装后，继续安装 pip

```
set path=%path%; C:\Python27\Scripts;
easy_install pip
pip list
```

显示已经安装的程序包。

pip 使用非常简单，**pip** 在 **pypi** 的 **Quickstart** 就足够了。

任务 2.3：安装编程环境与 IDE

简单程序；自带的 IDLE

小型程序；**sublime** + 插件 or **Notepad++** or emacs + 插件 or Vim + 插件

复杂程序；eclipse + pydev

注意：**eclipse** 可用于开发 **c++** or **java**。没玩过的同学，建议自己下载，编写一个小程序。在类 **Unix** 环境下，是最 **powerful** 的开发工具了。

任务 2.4：安装 **pylint**

```
pip install pylint
```

注意：无法通过 **pylint** 检查的代码，会被扣分

任务 2.5：安装 **Tornado**

```
pip install tornado
pip list
```

任务 2.6：设置环境变量

将 **python** 目录与 **python\scripts** 设置到 **path** 环境变量中。以保证以后所有程序都可以使用。

任务 3： Hello World for All

进入 **python** 官方网站，documentation -> Python2.x -> Tutorial

任务 3.1： Say Hello in Interpreter

进入文档 “2. Using the Python Interpreter”，
在控制台输入 **python**，出现提示符 “>>>”

```
>>> name = raw_input("Please enter your name: \n")
Please enter your name:
Su
>>> print "Hello %s" % name
Hello Su
>>> name
'Su'
>>> quit()
```

任务 3.2： Say Hello in Console

使用任意编辑器输入以下代码，存为 **hello.py**。

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  print u"%s 您好！" % sys.argv[1]
```

注意：前两行代码的含义与顺序，解释见文档 2.2.2.和 2.2.3.

控制台输入：python hello.py yourname

汉字处理是初学 python 的难题。

任务 3.3: Say Hello in Web

将 tornado 主页上程序复制到编辑器，保存为 webhello.py

Hello, world

以下是经典的 "Hello, world" 示例：

```
import tornado.ioloop
import tornado.web

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello, world")

application = tornado.web.Application([
    (r"/", MainHandler),
])

if __name__ == "__main__":
    application.listen(8888)
    tornado.ioloop.IOLoop.instance().start()
```

在控制台运行该程序。

python webhello.py

打开浏览器，输入 URL: <http://localhost:8888/>

该程序结构：库函数引用申明 -> HTTP 请求处理器 -> 应用程序实例(处理 URL)
-> main 程序

其中，r"/" 表示原始的字符串，u"" 表示 unicode 字符串。

大致的解释是：

main 程序启动等待客户端 HTTP 请求的循环；

application 对象接收 HTTP 请求，将 "/" 文档请求的 HTTP 命令，如 get, post 等交给了 MainHandler 类的实例。

MainHandler 实现了 GET 命令，将 Plain Text 输出给请求者。

(注意：如果你看不懂这段解释，维基百科“HTTP”)

任务 4: python 程序编程入门

任务 4.1: 基本数据类型

进入文档“3. An Informal Introduction to Python”

本部分你需要在 Python Interpreter 练习。

● 3.1.1. Number

完成前三个基础练习；

● 3.1.2 String

完成前两个练习；

完成“cat 两个字符串”操作练习；

练习“Strings can be subscripted (indexed)”以后的案例，你需要练习：

```
>>> word[-1]
'A'
>>> word[-2]
'p'
>>> word[-2:]
'pA'
>>> word[:-2]
'Hel'
```

index char; substring; 能解释 这些语句的含义。

练习求字符串长度 len(s);

问题 1: 字符数组下标从 0 or 1 开始的?

- 3.1.4. Lists (重要)

完成前四个基本练习;

问题 2: “Unlike strings, which are *immutable*,”。在 python 中 mutable 类型与 immutable 类型数据的区别是什么? 举例说明。

List 数据的 remove, insert, replace, clear, len 操作

- 3.2. First Steps Towards Programming

练习, 打印 Fibonacci 数列

任务 4.2: 控制流结构

进入文档 “4. More Control Flow Tools”

本部分所有需要在文本编辑器中录入, 在 Console 中运行。

建议: 将程序放置在类似 c:\workspace 这样的目录下, 避免汉字路径名, 长目录名带来的各种麻烦。

- 4.1. if Statements

练习第一个案例

- 4.2. for Statements

练习第一个案例

- 4.3. The range() Function

阅读前几个例子, 使用 for 语句计算[3..10]的和

- 4.6. Defining Functions

练习第一个案例

任务 5: python 能力测试

自我练习: 编一个小程序, 将 Fibonacci 数的前 10 个数存入 List, 并按倒序打印。

任务 6: tornado 入门

进入 tornado 中文主页 (<http://www.tornadoweb.cn>), 点击 “Tornado 文档”

任务 6.1: tornado 处理 URL 及参数的方法

选择 “请求处理程序和请求参数”

- 从 URL 中获取参数

修改 webHello.py 程序, 使用第一段案例代码替换原来的请求处理器和 application 实例部分。另存为 webInput.py。运行该程序。分别输入:

输入 URL: <http://localhost:8888/>

输入 URL: <http://localhost:8888/story/71>

请自己解释得到输出的过程。注: `[0-9]+` 表示可以匹配长度 1 以上由 0..9 组成的字符串。正则表达式将在后面介绍。

任务 6.2：使用模板输出

选择“模板”

按该部分要求操作，并运行程序。

输入 URL: <http://localhost:8888/>

请解释得到输出的过程。并此技术的应用的操作与实践中去。

小技巧：当你决定了用 `tornado` 开发网站，并且开始写新的 `handler` 的时候，你会发现需要频繁的，关掉，再重新启动服务器，这是一件非常让人抓狂的事情，我们需要在程序修改后，让服务能够自动 `reload` 新的代码，和 `php` 一样。

加一个 `debug=True` 到 `WebApplication` 的构造函数就可以实现了。

```
8 application = tornado.web.Application([
9     (r"/", MainHandler),
10 ], debug=True)
```

它是如何实现的，如果你从源代码中找到答案，请 email:panml@mail.sysu.edu.cn

作业与练习：

1、简答以下问题

- 1) 简述 python list 类型，请给出从头部插入元素和从尾部添加元素的语句。
- 2) 请谷歌“python list copy”或“python list clone”。请给出最简单的 shallow copy 方法。
- 3) 阅读文档 “The Python Standard Library -> 5.6. Sequence Types”。给出以下程序的 Output

```
s = "Eagle"
print s[ :-2]
```

```
#!/usr/bin/python
s = "ZetCode"
for i in s:
    print i,
```

```
#!/usr/bin/python
nums = "1,5,6,8,2,3,1,9"
k = nums.split(",")
print k
m = ':'.join(k)
print m
```

4) 举例解释 mutable 类型与 immutable 类型数据的区别。

二、操作与实践

1、Lab 3: Ex1 （比较简单，课堂内完成）

提示：

- 1) 下载“MP3 Music Library”中的“music.html”，改造成 tornado 的模板
- 2) 读取文件目录，请谷歌“python list files”或“python 文件遍历”。必要参考手册：os.path 部分。最简单方法 os.listdir(path)
- 3) 获得文件名，请谷歌“python extract filename from path”
- 4) 获得文件大小，请谷歌“python get file size”

三、Q&A

Q: 王博士的 pdf 上面 lecture3 的第四页上面看到一句红色的英文：Never use Table for layout! 但是，在 wiki 的源代码里面有大量使用 table 标签来布局，我试了一下，也觉得挺方便的。（李效良）

A: 使用<div>布局灵活方便，方便使用浮动和定位属性。配合 id 和 class 属性，展示的区域语义清晰，便于管理。<table>是复杂的 HTML 元素，<table>中元素浮动或定位比较复杂，容易导致具体浏览器实现的兼容问题。很多情况下，如果元素是矩阵排列的，不需要浮动或定位。因此，局部采用<table>布局，应该是合理的选择。