

19个练习技术的在线网站，你知道几个？

原创

快乐星球没有乐  已于 2022-01-31 16:30:46 修改  198  收藏 3

分类专栏: [程序员](#) 文章标签: [安全](#) [python](#)

于 2022-01-29 13:03:56 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/m0_58477260/article/details/122741816

版权



[程序员](#) 专栏收录该内容

91 篇文章 6 订阅

订阅专栏

不管你是一名开发人员、安全工程师、代码审计师、渗透测试人员,

即便你是零基础, 通过不断的练习才能让你成为一个优秀安全研究人员。

如果是对Python感兴趣的, 这边也提供全套学习资料, 请看文章末尾

以下网站希望能给各位安全小伙伴带来帮助! (排名不分先后)

1.track黑客导航

[导航Track黑客导航](#)

黑客导航, 含有众多网安一行的官方网站及平台

无论你是老鸟还是萌新, 都可以找得到对应适合的文章!

2.封神台靶场

[封神台 - 掌控安全在线演练靶场](#), 是一个在线黑客攻防演练平台。

在线练习靶场, 是一个在线黑客攻防演练平台!

在成长的道路上怎么能缺少打怪刷级的乐趣呢!

但是市面上的网站漏洞组成复杂, 而且会有法律风险, 想要技术, 封神台靶场是一个好的选择

3.黑客社区

[Track 安全社区 - 掌控安全在线教育](#)

[Track_黑客_社区](#)-追随黑客大神的脚印

安全动态每天看 黑客新闻

4. bWAPP

这是一个集成了各种常见漏洞和最新漏洞的**免费和开源的web应用程序安全项目**。

目的是帮助网络安全爱好者、开发人员和学生发现并防止网络漏洞。

地址: www.itsecgames.com/

5、Damn Vulnerable iOS App (DVIA)

DVIA是一个iOS安全的应用0,

它的主要目标给移动安全爱好者学习iOS的渗透测试技巧提供一个合法的平台。

APP涵盖了所有常见的iOS安全漏洞, 它免费并开放源码,漏洞测试和解决方案覆盖到iOS 10版本。

地址: damnvulnerableiosapp.com/

6、Damn Vulnerable Web Application (DVWA)

基于 php 和 mysql 的虚拟 Web 应用, “内置”常见的 Web 漏洞, 如 SQL 注入、xss 之类, 可以搭建在自己的电脑上

地址: www.dvwa.co.uk/

7、Game of Hacks

基于游戏的方式来测试你的安全技术, 每个任务题目提供了大量的代码,其中可能有也可能没有安全漏洞!

地址: www.gameofhacks.com/

8、Google Gruyere

一个看起来很low的网址, 但充满了漏洞, 这个网站中存在大量的安全漏洞, 专为那些刚开始学习Web应用安全的新手而设计, 该网站的目标主要有以下三个:

- 学习黑客是如何找出安全漏洞的;
- 学习黑客如何利用网站漏洞来实施攻击;
- 学习如何防止黑客发现并利用安全漏洞;

地址: google-gruyere.appspot.com/

9、HackThis!!

可以让你了解黑客是如何进行非法入侵和数据窃取等行为的, 并且教会你如何保护自己的网站免受黑客的入侵。

提供了超过五十种难度级别, 而且还有一个活跃的在线交流社区,

所以也是一个学习黑客技术、了解安全新闻和技术文章的好地方。

地址: www.hackthis.co.uk/

10、Hack This Site

对于任何人来说, HackThisSite都是一个练习黑客技术的好地方。

该平台提供了黑客新闻、技术文章、黑客论坛和大量新手教程, 你可以通过完成网站中的各种挑战任务来学习并练习黑客技术。

地址: www.hackthissite.org/

11、Hellbound Hackers

光说不练假把式!

提供了各种各样的安全实践方法和挑战,目的是教你如何识别攻击和代码的补丁建议。

主题包含应用程序加密和破解,社工和rooting。

社区有接近10万的注册会员,也是最大的一个黑客社区之一。

地址: www.hellboundhackers.org/

12、McAfee HacMe Sites

迈克菲提供的各类黑客及安全测试工具, 这些网站可以帮助广大渗透测试人员和安全专家提升自己的技能。

该项目的每一个app都模拟出了真实世界的应用场景, 就连其中的安全漏洞也与我们真实生活中的非常相似。

地址: www.mcafee.com/us/download...

13、Mutillidae

mutillidaemutillidae是一个免费, 开源的Web应用程序, 提供专门被允许的安全测试和入侵的Web应用程序。

其中包含了丰富的渗透测试项目, 如SQL注入、跨站脚本、clickjacking、本地文件包含、远程代码执行等。

地址: sourceforge.net/projects/mu...

14、OverTheWire

基于游戏的让你学习安全技术和概念的黑客网站

地址: overthewire.org/wargames/

15、Peruggia

一个提供安全、合法攻击的黑客网站

地址: sourceforge.net/projects/pe...

16、Root Me

让你通过200多个黑客挑战任务以及50多种虚拟环境来提升自己的黑客技术和Web安全知识。

地址: www.root-me.org/

17、Try2Hack

最古老的黑客网站之一,提供多种安全挑战。

地址: www.try2hack.nl/

18、Vicnum

OWASP项目之一,简单框架,针对不同的需求,并基于游戏的方式来引导安全开发者学习安全技术。

地址: vicnum.ciphertechs.com/

19、WebGoat

最受欢迎的OWASP项目,

这个不安全的app可以让我们在真实环境中学习如何去处理各种复杂的安全问题。

地址: webappsecmovies.sourceforge.net/webgoat/

「 Python经验分享 」

学好 Python 不论是**就业**还是**做副业赚钱**都不错,但要学会 Python 还是要有一个学习规划。最后给大家**免费**分享一份**全套的 Python 学习资料**,给那些想学习 Python 的小伙伴们一点帮助!

一、Python所有方向的学习路线

Python所有方向路线就是把Python常用的技术点做整理，形成各个领域的知识点汇总，它的用处就在于，你可以按照上面的知识点去找对应的学习资源，保证自己学得较为全面。



1. Python基础入门



2. Python高级编程



3. 爬虫



4. 数据分析



5. 前端开发



6. 后端开发



7. 机器学习





8. 自动化测试


二、学习软件


工欲善其事必先利其器。学习Python常用的开发软件都在这里了，给大家节省了很多时间。

 软件安装包和教程下这里

 7- Python第一本书


 6-python官方文档

 5-python编码规范

 4-python学习手册

 3-深入python

 2-python练习集

 1-python工具

三、入门学习视频

我们在看视频学习的时候，不能光动眼动脑不动手，比较科学的学习方法是在理解之后运用它们，这时候动手项目就很适合了。

p01-python前篇-程序与程序语言	p24-Python的比较运算符	p46-列表的创建	2021/9/18 13:40	文件夹
p02-python中篇-互联网软件流程思想解...	p25-Python的布尔运算符	p47-列表的特点	2021/9/18 13:40	文件夹
p03-python后篇-计算机基本组成	p26-Python的位运算符	p48-列表获取指定元素的索引	2021/9/18 13:40	文件夹
p04-python简介	p27-运算符优先级	p49-获取列表中单个元素	2021/9/18 13:40	文件夹
p05-python解释器安装	p28-程序的组织结构-顺序结构	p50-获取列表中多个元素	2021/9/18 13:40	文件夹
p06-PyCharm开发工具安装	p29-对象的布尔值	p51-列表元素的查询操作	2021/9/18 13:40	文件夹
p07-第一个函数print	p30-程序的组织结构-选择结构-单分支	p52-添加列表元素	2021/9/18 13:40	文件夹
p08-转义字符的使用	p31-程序的组织结构-双分支结构	p53-删除列表元素	2021/9/18 13:40	文件夹
p09-标识符与保留字	p32-程序的组织结构-多分支结构	p54-列表的排序	2021/9/18 13:40	文件夹
p10-python中的变量	p33-程序的组织结构-嵌套if使用	p55-列表生成式	2021/9/18 13:41	文件夹
p11-变量的多次赋值	p34-条件表达式	p56-字典的定义	2021/9/18 13:41	文件夹
p12-python的常用数据类型	p35-Python的pass语句	p57-字典的元素获取	2021/9/18 13:41	文件夹
p13-python的进制运算	p36-Python的range()函数	p58-字典的KEY判断	2021/9/18 13:41	文件夹
p14-python的float浮点类型	p37-Python的while循环结构	p59-字典的视图	2021/9/18 13:41	文件夹
p15-python的bool布尔类型	p38-Python的while循环练习	p60-字典元素遍历	2021/9/18 13:41	文件夹
p16-python的str字符串类型	p39-Python的for-in循环结构	p61-字典的特点	2021/9/18 13:41	文件夹
p17-数据类型如何转型	p40-break流程控制语句	p62-字典生成式	2021/9/18 13:41	文件夹
p18-其它类型转float类型	p41-continue流程控制语句	p63-什么是元组	2021/9/18 13:41	文件夹
p19-Python的注解	p42-else的使用	p64-元组的创建	2021/9/18 13:41	文件夹
p20-input函数的简单使用	p43-嵌套循环的使用	p65_元组为不可变序列	2021/9/18 13:41	文件夹
p21-input函数高级使用	p44-二重循环的break与continue使用	p66_元组的遍历	2021/9/18 13:41	文件夹
p22-Python的算术运算符	p45-列表的应用理解	p67_集合的定义	2021/9/18 13:41	文件夹
p23-Python的赋值运算符	p46-列表的创建	p68 集合的相关操作	2021/9/18 13:41	文件夹

四、实战案例

光学理论是没用的，要学会跟着一起敲，要动手实操，才能将自己的所学运用到实际当中去，这时候可以搞点实战案例来学习。

<input type="checkbox"/>	实例43_批量发送不同内容的邮件给不同的收件人	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例45_用Python分析文本数据的词频	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例46_Python文本数据可视化之“词云”图	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例47_Python替换不了word中的文字？	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例48_批量修改word文件中的段落格式	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例50_Python一键提取PDF中的表格到Excel	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例51_5行代码，Python给孩子出数学练习题	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例52_Pandas提取指定数据并保存在原Excel工作簿中	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例53_Python从原Excel表中抽出数据存入同一文件的新...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例54_Python指挥打印机批量打印文件	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例55_看你怎么作弊抄答案？Python出题，每个学生的...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例56_Python从多路径多Excel表中获取数据并存入新表	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例57_Python爬虫爬取会计师事务所网站的指定文章	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例58_Python爬虫~已爬取目标网站所有文章，后续如...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例59_Python检查word文件中的特殊标记词是否与文...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例60_Python汇总各单位Excel档领料记录并加总每日领...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例61_Python制作图形用户界面(GUI)让操作可视化	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例62_将Python程序打包成安装文件分享给小伙伴	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例63_Tkinter制作Python程序的图形用户界面(GUI)，...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例64_Python分块拆分txt文件中的数据	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例65_Python识别加密的word文件并移动到单独文件夹	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例66_Python一键更新Excel档“生产订单周报”的图表	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例67_Python爬取博客的所有文章并存储为带目录的wor...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例68_Python批量新建文件夹并保存日志信息	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例69_Python保留格式复制多个excel工作表到汇总表并...	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例70_Python批量将公众号文章保留原格式下载为PDF	2021-09-12 10:43	文件夹	-
<input type="checkbox"/>	实例71_Python从Excel表中批量复制粘贴数据到新表	2021-09-12 10:43	文件夹	-

五、面试资料

我们学习Python必然是为了找到高薪的工作，下面这些面试题是来自阿里、腾讯、字节等一线互联网大厂最新的面试资料，并且有阿里大佬给出了权威的解答，刷完这一套面试资料相信大家都能找到满意的工作。



alibaba面试真题及答案.docx



滴滴面试题.docx



抖音+快手面试题.docx



华为+小米+vivo+oppo面试题.docx



京东面试题集合.docx



蚂蚁金服面试题.docx



美团点评面试题.docx



顺丰4面试题及答案.docx



腾讯面试题.docx



网易+拼多多面试题.docx



小红书面试题.docx



字节跳动三面真题.docx

Python面试宝典

- 第一章 Python基础
 - 第一节 Python基础语法
 - 第二节 文件操作
 - 第三节 模块与包
 - 第四节 数据类型
 - 第五节 企业面试题
 - 1. 分别从前端、后端、数据库
 - 2. 我们都知道现在Python很火
 - 3. 如果你困在了死循环里，怎
 - 4. 请谈谈.pyc文件和.py文件的
 - 5. 请反转字符串'aStr'?
 - 7. 将字符串'k1|k1|k2|k3|k3
 - 8. 利用切片操作，实现一个trir
 - 9. 请设计一个decorator，它
 - 10. 解释一下python的and-or
 - 11. how do I iterate over a s
 - 12. 你在爬虫过程中，都是怎么
 - 13. 爬取这些数据都用来做什么
 - 14. 有没有一个工具可以帮助查
 - 15. 讲一下 mysql 的原理?
 - 16. 数据库索引
 - 17. 方案
 - 18. 数据库怎么优化查询效率?
 - 19. 简单谈谈ACID，并解释每
 - 20. Python的主要功能是什么?
 - 21. 装饰器的写法以及应用场景
 - 22. 异常处理写法以及如何主动
 - 23. isinstance作用以及应用场
 - 24. json序列化时，可以处理的
 - 25. 线上服务可能因为种种原
 - 26. 如何提高python的运行效
 - 27. 线上服务可能因为种种原
 - 28. 如何提高python的运行效
 - 29. 字符串的操作题目
 - 30. 介绍Cython, Pypy Cpythc
 - 31. 手写一个判断时间的装饰
 - 32. 生成器，迭代器的区别?
 - 33. Python里面如何拷贝一个?
 - 34. 如何判断是函数还是方法?
 - 35. 为什么是数字名字可以当做参
 - 36. 遇到bug如何处理
 - 37. 分别从前端、后端、数据
 - 38. python新式类和经典类的区
 - 39. 线上服务可能因为种种原
 - 40. 如何提高python的运行效

7. Python的列表的截取与字符串操作类型相同,如下所示

```
L = ['span', 'Span', 'SPAM']
print(L[-1]) # ['SPAM']
```

8. Python列表操作的函数和方法

```
len(a) # 列表元素的个数
max(a) # 返回列表元素最大值
min(a) # 返回列表元素最小值
list(tuple) # 将一个可迭代对象转换为列表
```

列表常用方法总结

```
a.append(4)
a.count(3)
a.extend([4, 5, 6])
a.index(3)
a.insert(0, 2)
a.remove()
a.pop()
a.reverse()
a.sort()
```

元组:

```
T = tuple('abc')
对元组进行排序
注意
当对元组进行排序的时候,通常先将其转换为列表并使其成为一个可迭代对象,或者使用sorted方法,它接收任何序列对象.
```

```
T = ('c', 'a', 'd', 'b')
tmp = list(T)
tmp.sort() ==> ['a', 'b', 'c', 'd']
T = tuple(tmp)
sorted(T)
```

字典:

以下实例展示了 frozenset()函数的使用方法。

```
实例(Python 2.0+)
#!/usr/bin/python
# -*- coding: UTF-8 -*-
seq = ('Google', 'Runoob', 'Taobao')
dict = dict.fromkeys(seq)
print "字典为: %s" % str(dict)
dict = dict.fromkeys(seq, 10)
print "字典为: %s" % str(dict)
以上实例输出结果为:
```

```
字典为: {'Google': None, 'Taobao': None, 'Runoob': None}
字典为: {'Google': 10, 'Taobao': 10, 'Runoob': 10}
通过zip函数构造字典
D = dict(zip(keyslist, valueslist))
通过赋值表达式元组构造字典(键必须是字符串,因为如果不是字符串,构造的时候也会当成是字符串处理)
D = dict(name='Bob', age=42) ==> {'name': 'Bob', 'age': 42}
```

列出所有的键、值,注意得到的是一个可迭代对象,而不是列表,用的时候需要转换

```
D.keys()
D.values()
D.items() --> 键 + 值
删除字典(根据键)以及长度
D.pop(key)
len(D)
del D[key]
新增或者是修改键对应的值
D[key] = value # 如果key已经在列表,如果不存在就创建.
字典推导式
D = {x:x**2 for x in range(10) if x % 2 == 0}
```

第五节 企业面试题

1. 分别从前端、后端、数据库阐述web项目的性能优化

该题目网上有很多方法,我不想截图网上的长串文字,看的头疼,按我自己的理解说几点

前端优化:

- 减少http请求,例如制作精灵图
- html和css放在页面上部, javascript放在页面下面,因为js加载比HTML和css加载慢,所以要优先加载html和css以防页面显示不全,性能差,也会影响用户体验

后端优化:

- 缓存存储读写次数高,变化少的数据,比如网站首页的信息,商品的值等等,应用程序读取数据时,一般是先从缓存中读取,如果读取不到或数据已失效,再访问磁盘数据库,并将数据再次写入缓存。
- 异步方式,如果有耗时操作,可以采用异步,比如Celery
- 代码优化,避免循环和判断次数太多,如果多个if else判断,优先判断最有可能发生的情况

数据库优化:

- 如有条件,数据可以存放于redis,读取速度快
- 建立索引、外键等

2. 我们都知道现在Python很火,但是对于一门技术不光要知道它的优点,也要知道它的缺点,请谈谈Python的不足之处。

Python有以下缺陷:

- Python的可解释特征会拖累其运行速度。
- 虽然Python在很多方面都性能良好,但在移动计算和浏览器方面表现不够好。
- 由于是动态语言,Python使用鸭子类型,即duck-typing,这会增加运行时的错误。

3. 如果你困在了死循环里,怎么打破它?

出现了这种问题,我们可以按Ctrl+C,这样可以打断执行程序,我们创建一个死循环来解释一下。

```
>>> def counterFunc(n):
>>>     while(n==2):print(n)
>>>     counterFunc(7)
>>> 7
>>> 7
```


这份完整版的Python全套学习资料已经[上传CSDN](#)

朋友们如果需要可以微信扫描下方二维码[免费领取【保证100%免费】](#)。



图片违规！