

Python编程新思维及实战

嵩天

Python常用标准库解析(下)

高天

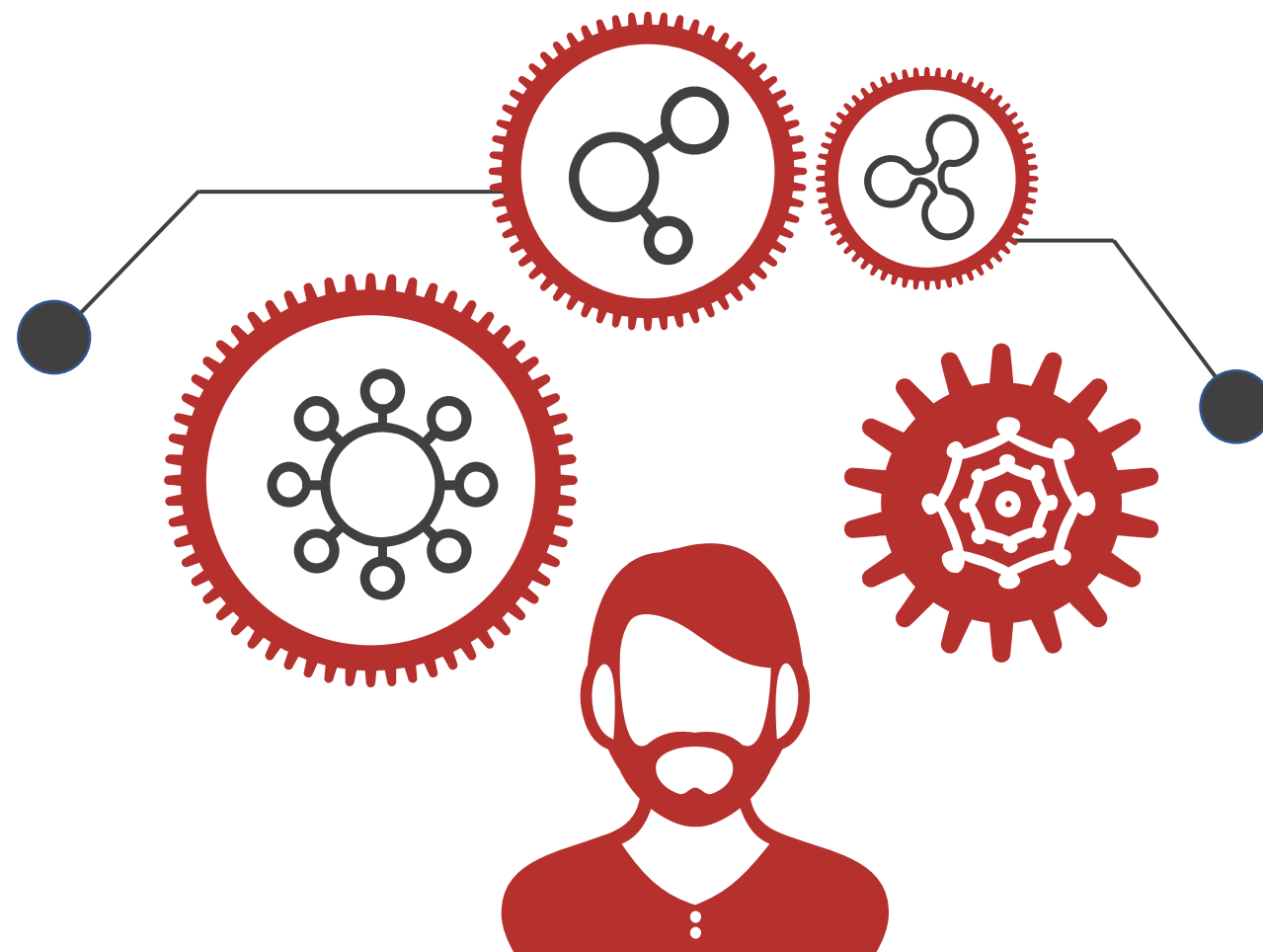


Python常用标准库解析(下)

单元开篇

单元开篇

(1) sys库的使用



(2) getopt库的使用

Python常用标准库解析(下)



Python常用标准库解析(下)

sys库的使用

sys库维护了一些与Python解释器相关的参数变量

- sys库提供的大部分是变量值，少部分函数
- sys库内容很丰富，这里仅列出常用部分，更多参数请参考手册

<https://docs.python.org/3.7/library/sys.html>

sys库常用参数

变量参数	描述
<code>sys.float_info</code>	系统浮点数类型的参数信息
<code>sys.int_info</code>	系统整数类型的参数信息
<code>sys.hash_info</code>	系统哈希函数的参数信息
<code>sys.thread_info</code>	系统线程实现的参数信息

sys库常用参数

变量参数	描述
<code>sys.byteorder</code>	字节序标识，大端返回'big'，小端返回'little'
<code>sys.executable</code>	当前解释器可执行文件的绝对路径
<code>sys.maxunicode</code>	系统支持的最大Unicode值，一般是1114111（0x10FFFF）
<code>sys.path</code>	解释器用来加载Python库的路径列表

sys库常用参数及函数

变量参数	描述
<code>sys.argv</code>	命令行参数的列表，其中argv[0]表示执行程序本身
<code>sys.dont_write_bytecode</code>	如果为True，则不生成.pyc字节码
<code>sys.getrecursionlimit()</code>	获得系统递归最大深度
<code>sys.setrecursionlimit(n)</code>	设置系统递归最大深度为n

sys库常用参数及函数

变量参数	描述
<code>sys.getdefaultencoding()</code>	获得当前默认的字符串编码名称
<code>sys.getsizeof(object)</code>	获得object的字节长度，object可以是任意类型
<code>sys.exit([arg])</code>	退出Python程序，附带状态码arg

sys库小结

- 常用参数: `sys.float_info`、`sys.int_info`、`sys.hash_info`、`sys.thread_info`
`sys.byteorder`、`sys.executable`、`sys.maxunicode`、`sys.path`
`sys.argv`、`sys.dont_write_bytecode`
- 常用函数: `sys.getrecursionlimit()`、`sys.setrecursionlimit()`
`sys.getdefaultencoding()`、`sys.getsizeof()`、`sys.exit()`

Python常用标准库解析(下)

getopt库的使用

getopt库介绍

getopt库提供解析命令行参数sys.argv的功能

- Unix命令行解析风格，与getopt()对应的C函数库风格一致
- 通过getopt库可以快速构建专业的命令行参数
- getopt库提供了1个函数和1个异常类型

<https://docs.python.org/3.7/library/getopt.html>

getopt库详解

`getopt.getopt(args, shortopts, longopts=[])`

- **args** : 程序的命令行参数, 不包括程序名称本身, 一般是`sys.argv[1:]`
- **shortopts** : 定义`-x`或`-x <值>`形式的短参数, 带值的增加冒号(:)
例: `"abc:d:"` 表示命令行可解析`-a -b -c<值> -d <值>`参数
- **longopts** : 定`--name` 或 `--name <值>`形式的长参数, 带值加等号(=)
例: `["name1", "name2="]`定`--name1` 和 `--name2 <值>`

getopt库详解

```
getopt.getopt(args, shortopts, longopts=[])
```

函数返回两个参数：optlist和args

其中，optlist包含由 (option, value) 组成的列表

args是未解析参数

getopt.GetoptError

参数解析错误时反馈的异常

实例展示

```
import getopt, sys
try:
    # -h -v -i <文件.ico> --distpath <目录> --clean
    opts, args = getopt.getopt(sys.argv[1:], "hi:v", ["distpath=", "clean"])
    for o, a in opts:
        if o == "-h":
            print("发现-h参数")
        elif o == "-v":
            print("发现-v参数")
        elif o == "-i":
            print("发现-i参数及其对应值 {}".format(a))
        elif o == "--distpath":
            print("发现--distpath参数及其对应值 {}".format(a))
        elif o == "--clean":
            print("发现--clean参数")
        else:
            print("发现未知参数")
    except getopt.GetoptError:
        print("命令行参数解析错误")
        sys.exit(2)
```

实例展示



```
import getopt, sys
try:
    # -h -v -i <文件.ico> --distpath <目录> --clean
    opts, args = getopt.getopt(sys.argv[1:], "hi:v", ["distpath=", "clean"])
    for o, a in opts:
        if o == "-h":
            print("发现-h参数")
        elif o == "-v":
            print("发现-v参数")
        elif o == "-i":
            print("发现-i参数及其对应值 {}".format(a))
        elif o == "--distpath":
            print("发现--distpath参数及其对应值 {}".format(a))
        elif o == "--clean":
            print("发现--clean参数")
        else:
            print("发现未知参数")
    except getopt.GetoptError:
        print("命令行参数解析错误")
        sys.exit(2)
```

解析命令行

实例展示

```
import getopt, sys
try:
    # -h -v -i <文件.ico> --distpath <目录> --clean
    opts, args = getopt.getopt(sys.argv[1:], "hi:v", ["distpath=", "clean"])
    for o, a in opts:
        if o == "-h":
            print("发现-h参数")
        elif o == "-v":
            print("发现-v参数")
        elif o == "-i":
            print("发现-i参数及其对应值 {}".format(a))
        elif o == "--distpath":
            print("发现--distpath参数及其对应值 {}".format(a))
        elif o == "--clean":
            print("发现--clean参数")
        else:
            print("发现未知参数")
    except getopt.GetoptError:
        print("命令行参数解析错误")
        sys.exit(2)
```

逐个判断参
数并处理

实例展示

```
import getopt, sys
try:
    # -h -v -i <文件.ico> --distpath <目录> --clean
    opts, args = getopt.getopt(sys.argv[1:], "hi:v", ["distpath=", "clean"])
    for o, a in opts:
        if o == "-h":
            print("发现-h参数")
        elif o == "-v":
            print("发现-v参数")
        elif o == "-i":
            print("发现-i参数及其对应值 {}".format(a))
        elif o == "--distpath":
            print("发现--distpath参数及其对应值 {}".format(a))
        elif o == "--clean":
            print("发现--clean参数")
        else:
            print("发现未知参数")
    except getopt.GetoptError:
        print("命令行参数解析错误")
        sys.exit(2)
```

异常处理

getopt库小结

- 处理函数: `getopt.getopt(args, shortopts, longopts=[])`
- 定义异常: `getopt.GetoptError`

Python常用标准库解析(下)

单元小结

单元小结

(1) sys库的使用
解释器相关参数和函数

(2) getopt库的使用
1个函数和1个异常

Python常用标准库解析(下)

sys库小结

- 常用参数: `sys.float_info`、`sys.int_info`、`sys.hash_info`、`sys.thread_info`
`sys.byteorder`、`sys.executable`、`sys.maxunicode`、`sys.path`
`sys.argv`、`sys.dont_write_bytecode`
- 常用函数: `sys.getrecursionlimit()`、`sys.setrecursionlimit()`
`sys.getdefaultencoding()`、`sys.getsizeof(object)`

getopt库小结

- 处理函数: `getopt.getopt(args, shortopts, longopts=[])`
- 定义异常: `getopt.GetoptError`

 Python ▶ 123

Thank you