

项目	语法	关键代码段
系统环境变量	<p>1.os.getenv(key,default=None) 获取环境变量的值</p> <p>2.environ属性 获取当前环境变量的值 (字符串)</p> <p>3.environb属性 获取当前环境变量的值 (字节)</p>	<pre>result = os.getenv('wind1', default='未定义该环境变量! ') result = os.getenv("path") # 获取PATH环境变量的路径 result = os.getenv('PATHEXT') # 可执行文件 result = os.getenv('SYSTEMROOT') # 系统主目录 result = os.getenv('LOGONSERVER') # 机器名 result = os.getenv("OS") # 当前系统名称 digits = os.getenv("PROCESSOR_ARCHITECTURE") # 读取当前处理器架构 result = os.getenv("programfiles") # 应用程序的默认安装目录 result = os.getenv("USERNAME") # 当前Windows账户名 data1 = os.environ # 获取当前操作系统的环境变量的映射对象 os.environ[newvar] = r'C:\Java\jdk1.8.0_181' # 创建环境变量 data2 = os.environ['WORKON_HOME']='test' # 设置环境变量 data3 = os.environ['HOMEPATH'] # 当前用户主目录 data4 = os.environ['LOGONSERVER'] # 机器名 os.environb # 获取以字节表示的当前环境变量的映射对象</pre>

json 模块应用超级魔卡 —— 读取、写入及转换

项目	语法	关键代码段
从JSON文件中读取数据	<p>json.load(fp)</p> <p>fp: 表示一个支持read()方法或包含一个JSON格式字符串的文本文件或二进制文件</p>	<pre>with open('./tmp/mr1.json','w+') as f: f.write(['a': 'A', 'c': 3.0, 'b': [2, 4]]) print(json.load(f)) data = json.load(f) for line in data: print(line['city'])</pre>
将JSON格式转换成Python对象	<p>json.loads(s,encoding)</p> <p>s: 包含JSON格式字符串、字节或字节数组 encoding: 编码方式</p>	<pre>with open('./tmp/city.json', 'r') as json_file: data = json_file.read() result = json.loads(data) file = requests.get(url,timeout=3) list1 = json.loads(file.text) # 将JSON格式的网页源码转换为列表</pre>
将Python对象转换为JSON格式字符串后写入到文件	<p>json.dump(obj, fp, ensure_ascii=True, allow_nan=True, indent=None, sort_keys=False)</p> <p>obj: Python对象 fp: 一个支持write()方法的文件对象</p>	<pre>data1 = [{'a': 'MR-SOFT', 'b': (8, 88), 'c': 8.8, 'd': 33}] with open('mr.json', 'w') as f: json.dump(data1, f, allow_nan=False, sort_keys=True, indent=4) csvf = open('./tmp/mr2.csv', 'r') # 读取CSV文件 reader = csv.DictReader(csvf) with open('./tmp/mr2.json', 'w') as f: # 转换为JSON文件 for r in reader: data = json.dump(r, f, ensure_ascii=False, indent=4)</pre>
将Python对象转换为JSON字符串	<p>json.dumps(obj, ensure_ascii=True, allow_nan=True, indent=None, sort_keys=False)</p>	<pre>data = [{'a': 'MR-SOFT', 'b': (8, 88), 'c': 8.8, 'd': 33}] json.dumps(data, allow_nan=False, sort_keys=True, indent=2) # 缩进显示 a = dict(name='sandy', age=17, QQ='null') b = json.dumps(a) # 将字典转换为JSON格式 data = ['ID', [1,2,3], {'name': '明日科技'}] # 创建一个列表 data = json.dumps(data, ensure_ascii=False) # 将列表转换为JSON格式</pre>