JsonParser使用说明

实现说明

1. 完成了要求中涉及的几个基本操作接口： load/dump loadJson/dumpJson loadDict/dumpDict
2. 提供了update，以及[]操作字典接口
3. 提供了额外的一个接口 `setFileEncoding` ：设置默认的文件编码。默认情况下， 使用 `utf-8` 作为默认编码。示例：

```
jp = JsonParser()
jp.setFileEncoding('gbk')
jp.loadJson('gbk-encdoing-in.json')

jp.setFileEncoding('utf-8')
jp.dumpJson('utf-8-encoding-out.json')
```

4. 解析json的格式参考了json格式说明文档
5. 对转义字符进行了考虑，具体包括 `/" /\ // /b /f /n /r /t /uxxxx` 几种形式。有两点需要额外说明：
    1. 如果读入了 `\/` 字符，那么会直接转换为 `/` ，写入文件的时候也会转换为 `/` (参考了 simplejson 中默认处理方式)
    2. 如果读入了 `\u` 字符，后面必须跟上4个16进制字符，否则抛出 `ValueError` 异常。读入了 `\uxxxx` 之后，会使用 `unichr` 函数将16进制数字翻译成unicode字符进行保存

使用示例

```
# 重建一个解析json对象
jp = JsonParser()

# 读取json文件
jp.loadJson('and-json-file.json')

# 读取json字符串
jp.load('{"a":[1,2,3]}')

# 读取json字典（key必须为字符串）
jp.load({'a':[1,2,3]})

# 可进行update和get/set操作
print jp['foo']
jp['foo'] = 'change-foo'

hook_dict = {'foo':'bar'}
```

```
jp.update(hook_dict)

# 保存到文件中
jp.dumpJson('out.json')

# 得到json字典
json_dict_data = jp.dumpDict()

# 得到json字符串
json_str = jp.dump()
```

源码说明

1. JsonParser.py **json**解析类
2. test.py 对**JsonParser**的测试