

Course: CS562  
Name: Yanshen Wang  
Date: 01/22/16  
Proposal of Project

## 1. Identify a Python library

In this Project, I will plan to use the TSTL tool to test a python library which is use for translating the pronunciation of Chinese characters into Latin alphabet. With the help of this python library, after users input some Chinese characters, they can easily get some results including the letters of pinyin, the tone marks and the initial of each characters. However, in currently the technique of Chinese characters' conversion is not very mature, in other words, users will always meet many different problems when they utilize this kind of applications. Therefore, I want to use some software testing techniques, which I have learned in this class, to try to find out some bugs from this python library.

## 2. The function calls under this python library

Since the purpose of this python library is to convert Chinese characters into Latin letters, there exits some function calls to help users to get those results.

***get\_pinyin()***: When users implement this function, after they input some Chinese characters, the result will return some Latin letters.

e.g.

```
get_pinyin(u"北京")  
'bei-jing'
```

In addition, if users apply the parameters '*show\_tone\_marks*', the result will not only return the letters, but also return pronunciation marks of those letters.

e.g.

```
get_pinyin(u"上海", show_tone_marks=True)  
'sh àng-hǎi '
```

Actually the default splitter of this program is ` `, but if users want to change this kind of splitter, just add the whitespace character in this function.

e.g.

```
get_pinyin(u"深圳", ' ')  
'shen zhen'
```

***get\_initial()***: In this function, after users input a signal Chinese character, the result will return the initial capital letter of this character.

e.g.

```
get_initial(u"广")  
'G'
```

***get\_initials()***: In this function, after users input a bunch of Chinese character, the result will return the initial capital letters of each character.

e.g.

```
get_initials(u"广州")  
'G-Z'
```

### 3. What I want to test about this python library

In order to try to find bugs as many as possible in this python library, I will plan to test it in the following parts:

- a. First of all, in the syntax of Chinese pronunciation, one character may have two or more pronunciations. For example, in the phrase '阿姨', the pronunciation of character '阿' is 'a'; while in the phrase '阿胶', the pronunciation of character '阿' is 'e'. Therefore, I want to try to test this python library to find out whether or not this program can detect precisely about the character which has two or more different pronunciations.
- b. In addition, I will try to input some mixed strings into the function to find out if this library will meet some bugs. For instance, I may combine some English words and Chinese characters together as the input strings such as 'Hello 北京', and then find out what kind of result I will get.
- c. Furthermore, I want to test the maximum length of input strings in those functions, and for example in the ***get\_pinyin*** function, I will input a long Chinese poem to detect whether or not this python program will meet any problem.
- d. Usually, the Chinese characters include two kinds of different types: one is called simplified Chinese characters, and the other is called Traditional Chinese characters.

For example, '开' represents the simplified Chinese characters, and '開' represents the Traditional Chinese characters. Therefore, I want to input several Traditional Chinese characters to detect if this program can recognize them.

- e. Finally, I will try to pick up some uncommon used Chinese characters (such as '饕餮') to test if this program can recognize those characters accurately.

### 4. Where I found this python library

Actually I have searched lots of python libraries in the GitHub, and the original website of this python library is <https://github.com/lxneng/xpinyin>. After I install this kind of third python library, I can successfully utilize each function calls from this library. Therefore, I think this is a good third python library to be implemented by the TSTL tools.