*921531110*

**علی فاضلی**

**جناب آقای دکتر فیضی درخشی**

|  |
| --- |
|  |
| گزارش پروژه اولیه درس مبانی پردازش زبان گفتار |
|  |
|  |

**استاد محترم:**

گزارش پروژه اولیه درس مبانی پردازش زبان گفتار

علی فاضلی

**هدف پروژه :**

خواندن و وارد کردن یک متن فرضی با زبان فارسی یا انگلیسی با سیستم انگودینگ Unicode utf-8 ، از یک فایل اکسل و پردازش آن

پردازش اولیه در فاز اولیه شامل اعمال زیر می شود:

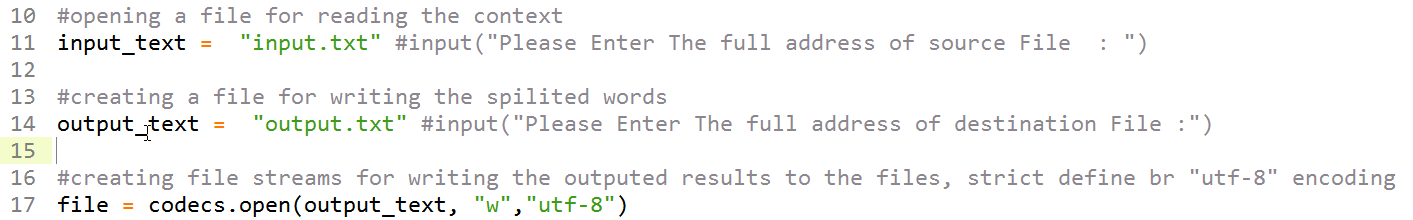
* تبدیل و خواندن متن از فایل با پسوند داده شده
* شناسایی و کار با متن فارسی و نمایش صحیح آن در کنسول
* گرفتن متن به عنوان ورودی و خواندن متن بطور کامل و تبدیل آن به کلمات تشکیل دهنده آن
* نمایش کد utf-8 مربوط به هرکلمه و متعاقبا معادل یونیکد هر حرف
* شناسایی و یا حذف علائم نگارشی یا Stop words
* شمردن تعداد کلمات موجود در متن بدون تکرار و به خروجی بردن تعداد تکرار هر کلمه در متن
* حذف کلمات تکراری موجود در متن

**توضیح کد:**

برای اینکار از زبان برنامه نویسی پایتون استفاده شد. از کتابخانه های codecs و String و Collection به منظور کارایی بیشتر استفاده گردید.

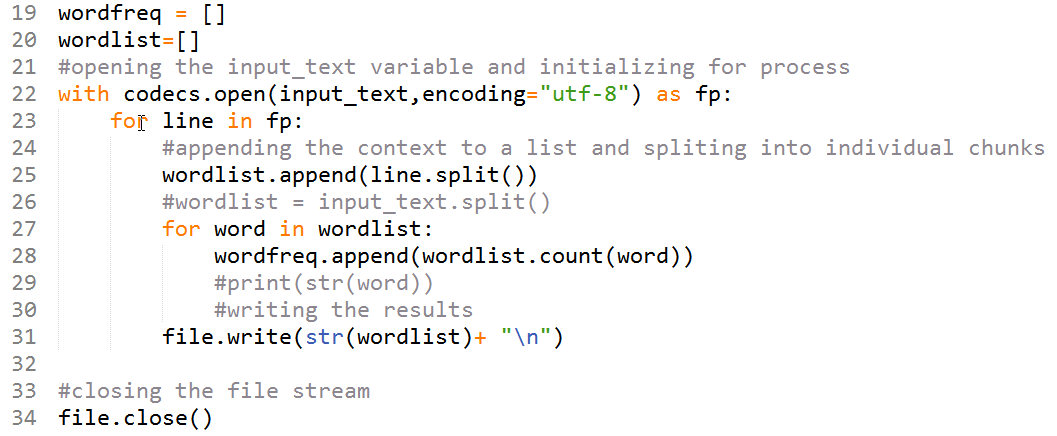
در ابتدا مسیر دقیق فایل های ورودی (Input.txt) که حاوی متن ورودی ما (در فاز اولیه پروژه) خواهد بود از کاربر گرفته میشود.

**تصویر 1- گرفتن مسیر فایل های متنی مورد نیاز**



در دو قسمت بعدی در تصویر 1 مشاهده میکنیم که مسیر فایل خروجی و مسیر فایل خروجی با محتوای یونیکد متن دریافتی از کاربر گرفته میشود. یکی از مشکلات در این قسمت ، مربوط به Permission ها برای ایجاد و کار با فایل ها هست که در سیستم عامل لینوکس با chmod 77 مشکل رفع میشود اما در ویندوز بعضا نیاز به پاک کردن فایل های output.txt و Unicode.txt برای شروع مجدد برنامه هستیم.

سپس دو فایل استریم درست میکنیم تا با استفاده از آن به محتویات داخلی فایل های ورودی و خروجی دسترسی داشته باشیم. دقت داشته باشید که فایل input.txt بایستی با انکودینگ UTF-8 ذخیره شود تا برنامه بدرستی کار کند.



**تصویر 2- جداسازی متن به کلمه ها و کار با انکودینگ برای هماهنگی کاراکتر ها**

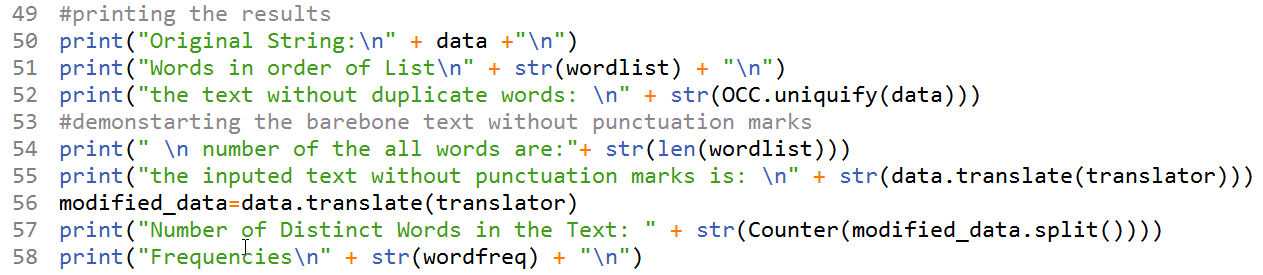
در تصویر 2 می بینیم که در ایتدا فایل input\_text در حافظه فراخوانی شده و محتویات آن ( که قبلا ان را به دلخواه پر کرده ایم) را خط به خط میخوانیم، سپس با متد split() که از متد های کلاس String می باشد متن را به کلمات تشکیل دهنده آن تقسیم کرده و آن را (بطور اتوماتیک) در یک ساختمان داده ای لیست قرار میدهد. سپس در یک حلقه تکرار دیگر به ازای تمامی کلمه های جدا شده از متن ، میزان تکرار آنها را در یک ساختمان داده ای دیگر ذخیره می کنیم.

سپس در فایل استریم file ( که مربوط به نوشتن خروجی کلمات جدا شده از متن می باشد) مقادیر معادل آن کلمات ( و به تبع آن حروف آن کلمات) در آن فایل نوشته می شوند. و در آخر با دستور Close() آنها را از حافظه سیستم پاک میکنیم.

C:\Users\K550 VX\Documents\ShareX\Screenshots\2017-02\sublime_text_2017-02-19_21-11-19.png

**تصویر 3- قسمت حذف علایم نگارشی**

در تصویر سه قابل مشاهده است که تمامی علائم نگارشی موجود در انکودینگ UTF-8 بارگذاری شده و به منظور جداسازی و حذف آنها در تمامی نقاط متن به جای آنها جای خالی جایگذاری میشود. (توضیح آرگومان اول و دوم).



**تصویر 3- چاپ کردن مقادیر محاسبه شده در کنسول خروجی**

در آخر هم مقادیر محاسبه شده اعم از مقدار اصلی متن ، کلمات جا شده یا اصطلاحا tokenize شده ، متن اصلی بدون داشتن کلمات تکراری، تعداد تکرار هر کلمه و تعداد کل کلمات و متن نهایی بدون علائم نگارشی و میزان تکرار کلمات متمایز در خروجی چاپ میشوند.

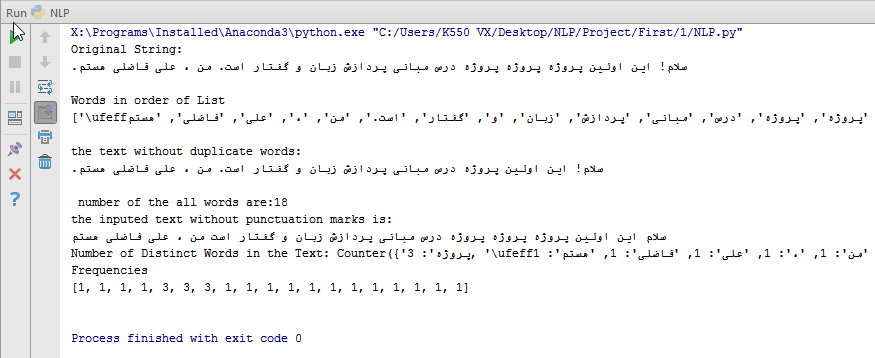
**اجرای آزمایشی پروژه :**

در ابتدا یک متن فرضی به عنوان متن آزمایشی به برنامه داده میشود تا از صحت کارکرد آن اطمینان حاصل شود.

متن زیر به عنوان متن آزمایشی به برنامه ( فایل Input.txt) داده شد.

" سلام! این اولین پروژه پروژه پروژه درس مبانی پردازش زبان و گفتار است. من ، علی فاضلی هستم."

متن مفروض دارای علائم نگارشی و کلمات تکراری می باشد که تعمدا برای آزمایش صحت کارکرد برنامه در نظر گرفته شده است.

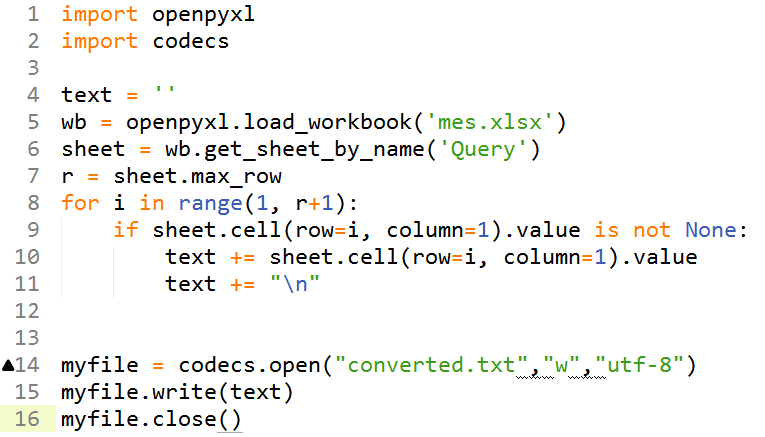


**تصویر 4- خروجی برنامه به ازای رشته ورودی مفروض**

همانطور که در تصویر مشخص است برنامه همانطور که انتظار میرفت بدرستی کار میکند و خروجی های توابع مختلف برنامه قابل مشاهده می باشد.

**اجرای پروژه :**

در این مرحله، بدلیل اینکه برنامه بنده با فایل هایی با پسوند .txt کار میکرد، اقدام به تبدیل فایل اکسل به فایل متنی کردم.



**تصویر 5-کد تبدیل فایل با پسوند xls به .txt**

سپس فایل متنی تبدیل شده (Converted.txt) را به عنوان ورودی Input\_text برنامه داده شد. فایل داده شده دارای انواع کاراکتر های یونیکد بوده و از این جهت کامل محسوب میشود. برنامه حدود 15 دقیقه برای اجرا زمان برد. فایل خروجی (output\_text) بدلیل آنکه به ازای هر کلمه جدا شده دو عدد (") به منظور جدا کردن کلمات استفاده میکند از حجم قابل توجهی برخوردار بود و خود این حجم از مشکلات این پروژه (کُد) به حساب می اید.

نتایج در فایلResults.txt در ضمایم قابل دسترسی می باشد.