python에서 한글 자음/모음 분해하기

3 분 소요

intro

- 네트워크 분석을 하는데, 사람 이름이 많이 나와서 이를 코드화 하려고 해요. 그냥 `person_1` 뭐 이렇게도 할 수 있겠지만, 이보다는 사람 이니셜로 변경하면 좋을 것 같더라고요.
- 그래서 한글을 영어 이니셜로 변경(이승훈 ==> LSH) 하려는데 그보다 앞서서 한글 자음/모음을 분해해야 합니다.
- 그 부분을 코드로 구현해봤습니다.

자음 모음 분해

- 저는 초성/중성/종성으로 구분할 수 있을 줄 알았는데, 그럴수없습니다. 유니코드 상에서는 예를 들어 “각”이라는 글자가 있다면 이를 [ㄱ, ㅏ , ㄱ]으로 생각하는 것이 아니라, ‘각’이라는 단일 글자에 대해서 번호를 부여하여 관리합니다.
- 즉, 번호를 보고 알아서 초성/중성/종성을 구분하는 코드를 만들어야 한다는 이야기죠.
- 이미 여기 (https://github.com/neotune/python-korean-handler/blob/master/korean_handler.py)에 잘 정리되어 있습니다만, 저는 조금 제 방식에 맞춰서 코드를 변경했습니다. 뭐 별차이는 없습니다만…

```
# 초성 리스트. 00 ~ 18
CHOSUNG_LIST = ['ㄱ', 'ㄲ', 'ㄴ', 'ㄷ', 'ㄸ', 'ㄹ', 'ㅁ', 'ㅂ', 'ㅃ', 'ㅅ', 'ㅆ', 'ㅇ', 'ㅈ', 'ㅉ', 'ㅊ', 'ㅋ', 'ㅌ', 'ㅍ', 'ㅎ']
# 중성 리스트. 00 ~ 20
JUNGSUNG_LIST = ['ㅏ', 'ㅐ', 'ㅑ', 'ㅒ', 'ㅓ', 'ㅔ', 'ㅕ', 'ㅖ', 'ㅗ', 'ㅘ', 'ㅙ', 'ㅚ', 'ㅛ', 'ㅜ', 'ㅝ', 'ㅞ', 'ㅟ', 'ㅠ', 'ㅡ', 'ㅢ', 'ㅣ']
# 종성 리스트. 00 ~ 27 + 1(1개 없음)
JONGSUNG_LIST = [' ', 'ㄱ', 'ㄲ', 'ㄳ', 'ㄴ', 'ㄵ', 'ㄶ', 'ㄷ', 'ㄹ', 'ㄺ', 'ㄻ', 'ㄼ', 'ㄽ', 'ㄾ', 'ㄿ', 'ㅀ', 'ㅁ', 'ㅂ', 'ㅄ', 'ㅅ', 'ㅆ', 'ㅇ', 'ㅈ', 'ㅊ', 'ㅋ', 'ㅌ', 'ㅍ', 'ㅎ']

def korean_to_be_englished(korean_word):
    r_lst = []
    for w in list(korean_word.strip()):
        ## 영어인 경우 구분해서 작성함.
        if '가'<=w<='힣':
            ## 588개 마다 초성이 바뀜.
            ch1 = (ord(w) - ord('가'))//588
            ## 중성은 총 28가지 종류
            ch2 = ((ord(w) - ord('가')) - (588*ch1)) // 28
            ch3 = (ord(w) - ord('가')) - (588*ch1) - 28*ch2
            r_lst.append([CHOSUNG_LIST[ch1], JUNGSUNG_LIST[ch2], JONGSUNG_LIST[ch3]])
        else:
            r_lst.append([w])
    return r_lst

korean_to_be_englished("이승훈a")
```

```
[['ㅇ', 'ㅣ', ' '], ['ㅅ', 'ㅡ', 'ㅇ'], ['ㅎ', 'ㅜ', 'ㄴ'], ['a']]
```

초성 ==> 영문 이니셜

- 이제 초성을 영문 이니셜로 변경합니다. 간단하게 딕셔너리를 만들어서 변환해줍니다.

```
def korean_word_to_initials(korean_word):
    """
    한글을 입력받아서 한글 초성에 따라서 이니셜로 변환해줍니다.
    한국 성의 경우 조금 다르게 변환되는데 '박' ==> 'Park'인 부분은 반영하지 않음
    """
    w_to_k = {'ㄱ':'K', 'ㄲ':'G', 'ㄴ':'N', 'ㄷ':'D', 'ㄸ':'D', 'ㄹ':'R', 'ㅁ':'M', 'ㅂ':'B',
              'ㅃ':'B', 'ㅅ':'S', 'ㅈ':'J', 'ㅉ':'J', 'ㅊ':'C', 'ㅌ':'T', 'ㅍ':'P', 'ㅎ':'H'}
    r_lst = []
    for i, w in enumerate(korean_to_be_englished(korean_word)):
        if w[0] in w_to_k.keys():
            r_lst.append( w_to_k[w[0]] )
        else:
            if w[1] in ['ㅑ', 'ㅕ', 'ㅛ', 'ㅠ', 'ㅖ']:
                r_lst.append('Y')
            elif w[1] in ['ㅝ', 'ㅘ', 'ㅙ', 'ㅚ', 'ㅜ', 'ㅞ', 'ㅟ']:
                r_lst.append('W')
            elif w[1] in ['ㅔ', 'ㅡ', 'ㅢ']:
                r_lst.append('E')
            elif w[1] in ['ㅏ', 'ㅐ']:
                r_lst.append('A')
            elif w[1] in ['ㅓ']:
                r_lst.append('U')
            elif w[1] in ['ㅗ']:
                r_lst.append('O')
            elif w[1] in ['ㅣ']:
                if i==0:
                    r_lst.append('L')
                else:
                    r_lst.append('I')
            else:
                return 'not applicable'
    return "".join(r_lst)
```

test case

- 간단하게 test case를 돌려봅니다.
    - 박씨를 B로 번역해주는 문제가 있습니다만, 제가 볼 때는 이정도면 쓸만해요 하하핫.

```
### test case
test_cases = ['이승훈', '전지현', '하정우', '정우성', '박상원', '정유미', '김연우', '윤종신']
for case in test_cases:
    print("{} ==> {}".format(case, korean_word_to_initials(case)))
```

```
```

```
이승훈 ==> LSH
전지현 ==> JJH
하정우 ==> HJW
정우성 ==> JWS
박상원 ==> BSW
정유미 ==> JYM
김연우 ==> KYW
윤종신 ==> YJS
```

reference

- https://github.com/neotune/python-korean-handler/blob/master/korean_handler.py (https://github.com/neotune/python-korean-handler/blob/master/korean_handler.py)

raw code

```
def korean_to_be_englished(korean_word):
    """
    한글 단어를 입력받아서 초성/중성/종성을 구분하여 리턴해줍니다.
    """
    ###################################
    # 초성 리스트. 00 ~ 18
    CHOSUNG_LIST = ['ㄱ', 'ㄲ', 'ㄴ', 'ㄷ', 'ㄸ', 'ㄹ', 'ㅁ', 'ㅂ', 'ㅃ', 'ㅅ', 'ㅆ', 'ㅇ', 'ㅈ', 'ㅉ', 'ㅊ', 'ㅋ
', 'ㅌ', 'ㅍ', 'ㅎ']
    # 중성 리스트. 00 ~ 20
```

```
    JUNGSUNG_LIST = ['ㅏ', 'ㅐ', 'ㅑ', 'ㅒ', 'ㅓ', 'ㅔ', 'ㅕ', 'ㅖ', 'ㅗ', 'ㅘ', 'ㅙ', 'ㅚ', 'ㅛ', 'ㅜ', 'ㅝ', 'ㅞ', 'ㅟ', 'ㅠ', 'ㅡ', 'ㅢ', 'ㅣ']
    # 종성 리스트. 00 ~ 27 + 1(1개 없음)
    JONGSUNG_LIST = [' ', 'ㄱ', 'ㄲ', 'ㄳ', 'ㄴ', 'ㄵ', 'ㄶ', 'ㄷ', 'ㄹ', 'ㄺ', 'ㄻ', 'ㄼ', 'ㄽ', 'ㄾ', 'ㄿ', 'ㅀ', 'ㅁ', 'ㅂ', 'ㅄ', 'ㅅ', 'ㅆ', 'ㅇ', 'ㅈ', 'ㅊ', 'ㅋ', 'ㅌ', 'ㅍ', 'ㅎ']
    ####################################
    r_lst = []
    for w in list(korean_word.strip()):
        if '가'<=w<='힣':
```

```
            ch1 = (ord(w) - ord('가'))//588
            ch2 = ((ord(w) - ord('가')) - (588*ch1)) // 28
            ch3 = (ord(w) - ord('가')) - (588*ch1) - 28*ch2
            r_lst.append([CHOSUNG_LIST[ch1], JUNGSUNG_LIST[ch2], JONGSUNG_LIST[ch3]])
        else:
            r_lst.append([w])
    return r_lst

def korean_word_to_initials(korean_word):
```

```
    """
    한글을 입력받아서 한글 초성에 따라서 이니셜로 변환해줍니다.
    한국 성의 경우 조금 다르게 변환되는데 '박' ==> 'Park'인 부분은 반영하지 않음
    """
    w_to_k = {'ㄱ':'K', 'ㄲ':'G', 'ㄴ':'N', 'ㄷ':'D', 'ㄸ':'D', 'ㄹ':'R', 'ㅁ':'M', 'ㅂ':'B',
              'ㅃ':'B', 'ㅅ':'S', 'ㅈ':'J', 'ㅉ':'J', 'ㅊ':'C', 'ㅌ':'T', 'ㅍ':'P', 'ㅎ':'H'}
    r_lst = []
    for i, w in enumerate(korean_to_be_englished(korean_word)):
        if w[0] in w_to_k.keys():
```

```
            r_lst.append( w_to_k[w[0]] )
        else:
            if w[1] in ['ㅑ', 'ㅕ', 'ㅛ', 'ㅠ', 'ㅖ']:
                r_lst.append('Y')
            elif w[1] in ['ㅝ', 'ㅘ', 'ㅙ', 'ㅚ', 'ㅜ', 'ㅞ', 'ㅟ']:
                r_lst.append('W')
            elif w[1] in ['ㅔ', 'ㅡ', 'ㅢ']:
                r_lst.append('E')
            elif w[1] in ['ㅏ', 'ㅐ']:
```

```
            r_lst.append('A')
        elif w[1] in ['ㅓ']:
            r_lst.append('U')
        elif w[1] in ['ㅗ']:
            r_lst.append('O')
        elif w[1] in ['ㅣ']:
            if i==0:
                r_lst.append('L')
            else:
```

```
                r_lst.append('I')
            else:
                return 'not applicable'
    return "".join(r_lst)
```

태그: korean python-lib python string

카테고리: python

업데이트: July 23, 2018

댓글남기기