문자열 총정리 (1)

1. 정규식

- 형태 알아두기

```
import re
바뀌는 문자열=re.sub("[지우고싶은애]' ,'',원래문자열)
예)
s="hello112, world123 !"
s=re.sub("[^\w]","",s) ## 나는 특수문자를 지우겠다. (^\w=\W)
print(s)
```

- 몇가지 더 알아두면 좋은 표현들
  - `[A-Z]` → 대문자만, 이외외도 `,.*&` 같은것도 지울 수 있음
  - `-` 을 지우고 싶을때는 맨 마지막에 쓰기 (-가 ~부터 ~까지라는 표현으로도 쓰임)
    예시) 2단계 new_id에서 알파벳 소문자, 숫자, 빼기(-), 밑줄(_), 마침표(.)를 제외한 모든 문자를 제거합니다.

```
id=re.sub('[^a-z0-9_.-]','',id)
```

  - `\w` → alphabet+ numeric , `\W=^\w` (알파벳, 숫자 제외한 나머지)
  - `\s` → 공백

2. Join, Split

- `Join`

  문자열로된 배열을 문자열로로 바꾸는 과정

  `문자열.join(문자열배열)`

```
a.join(['apple', 'pear', 'grape', 'pineapple', 'orange'])
a-> 'apple pear grape pineapple orange'
```

- `Split`
  문자열을 특정 기준을 통해 배열로 바꿔주는 것
  `split('기준')`

```
a= 'apple pear grape pineapple orange'
a가 공백을 기준으로 구분되어있으므로, 단어별로 배열을 만들어 주려면
a.split()-> ['apple', 'pear', 'grape', 'pineapple', 'orange']를 리턴

a= 'apple,pear,grape,pineapple,orange'
a가 ','를 기준으로 구분되어 있으므로, 단어별로 배열을 만들어 주려면
a.split(',')-> ['apple', 'pear', 'grape', 'pineapple', 'orange']를 리턴
```

3. 정렬

- 사전순 정렬

```
s="Would you like some coffee? Or do you want some water?"
print(re.sub('[\W]',' ',s).lower())## 특수문자를 제거-> 단어별로 공백차이가 나있음
print(re.sub('[\W]',' ',s).lower().split()) ##공백을 기준으로 스플릿진행
s=[word for word in re.sub('[\W]',' ',s).lower().split()] ##특수문자를 제거하겠다..
print(s)
s.sort(key=lambda x: x)
```

`sort` 함수 특성상, split 등을 통해서, 정렬 하기 이전에, 배열의 형태를 미리 만들어 주는 것이 포인트!

- 여러 형태의 정렬

```
s.sort(key=lambda x: (x[1:]))
```

단어별로, 두번째 글자 이후로 정렬

4. 여러가지 형태의 문자열 처리 함수들

- 문자열 바로 뒤집기
  `s=s[::-1]`
- `find`

```
s="AaBbCc,......_-==="
s.find(A) -> 가장먼저나오는 A의 index를 반환 (단순 문자 뿐만아니라, 문자열도 들어갈 수 있음)
s.find(!) -> !는 없으니, -1을 반환
```

- `replace`

  `문자열.replace(치환하고 싶은 문자열, 새로운 문자, 치환 횟수)`

  치환하고 싶은 문자열을 모두 찾고, 새로운문자로 바꿔줌, 없애고 싶으면 “” 같은거 사용

```
s=s.replace("..",".")
->  ".." 를 모두 찾고, 전부 "."로 바꿔줌, 중복되는 ".." 가 안나오게하려면
while s.find("..")!=-1: #못 찾았을 때 -1을 반환
    s=s.replace("..",".")

st='this is a  text' a뒤에 띄어쓰기 두칸일경우
st.replace("a ","a") "a " 를 "a"로 바꿔주기
```

- `lstrip,rstrip,strip`

```
s=" AaBbCc,......_-==="
s=s.lstrip() -> strip 함수에 아무것도 없을경우에는, 공백이 있을경우 제거한다라는 의미 공백이
없으면 그대로 나옴
s=s.lstrip("A") -> strip함수에 뭐가 들어있을경우, 그문자가 있으면 지워라 라는 의미,
마찬가지로, 원래문자열에 해당 문자가 없으면 그냥 skip

햇갈릴만한 부분
s=".......gaklghlka......"
s=s.lstrip(".")
-> 왼쪽에 있는 .을 모두지움 (하나만 지우는게 아님)
```

- `lower,upper, isdigit,isalpha`

  `lower` → 소문자로, `upper` → 대문자로

  `isdigit` → 문자열 전체가 숫자로 되어있는지 확인

  `isalpha` → 문자열 전체가 알파벳으로 되어있는지 확인

  코딩테스트 연습 - 신규 아이디 추천 | 프로그래머스 스쿨 (programmers.co.kr)

  ▼ code

```
import re
def solution(new_id):

    ##1 lower upper isalpha isnumeric(), isdigit()
    new_id=new_id.lower()
    ##2 정규식 형태
    new_id=re.sub('[^a-z0-9_.-]','',new_id)

    ##3 find 위치반환 없으면 -1
    while new_id.find('..')!=0:
        new_id=new_id.replace('..','.')

    ##4 lstrip rstrip
    #new_id=new_id.lstrip('.')
    #new_id=new_id.rstrip('.')
    if new_id=='.':
        new_id=""
    else:
        if new_id[0]==".":
            new_id=new_id[1:]
        if new_id[-1]==".":
            new_id=new_id[:-1]

    ##5 string 추가할때 주의
    if new_id=="":
        new_id='a'
    if len(new_id)>=16:
        new_id=new_id[:15]
        if(new_id[-1]=='.'):
            new_id=new_id[:-1]
    if len(new_id)<=2:
        while(len(new_id)<3):
          new_id+=(new_id[-1])

    answer = new_id
    return answer
```

코딩테스트 연습 - 문자열 다루기 기본 | 프로그래머스 스쿨 (programmers.co.kr)