Problem 2. 아래 3가지 조건들을 만족하도록 word_tokenizer 함수를 작성하시오.

먼저 띄어쓰기를 기준으로 주어진 `document` 를 `split` 하여 `r` 이라는 list를 만들고 list의 길이를 `l` 이라고 합니다. 길이 `l` 은 이하에 있는 코드의 for문에서 사용될 예정입니다. 문제에 주어진 조건을 모두 만족시킨 후 list `r` 을 return합니다. 다음은 각 조건을 만족하도록 작성한 코드입니다.

In [1]:

```
document = '''i've 'hello' 'hello'world' imlab's PH.D I.B.M snu.ac.kr 127.0.0.1 galago.gif
ieee.803.99 naver.com gigabyte.tw pass..fail'''
r = document.split(" ")
l = len(r)
```

2-1. 따옴표로 시작해서 따옴표로 끝나는 단어는 따옴표만 없애시오. 그리고 단어 도중에 따옴표가 나오는 경우 따옴표를 포함한 뒤의 글자들을 모두 삭제하시오.

for문을 사용하여 list `r` 에 있는 각 element가 문제의 조건을 만족시키는지 여부를 확인합니다. 먼저 따옴표로 시작해서 따옴표로 끝나는지 확인합니다. 이 조건을 만족시키는 경우 strip을 사용하여 맨 앞과 맨 끝에 있는 따옴표를 삭제합니다.

그 다음 `find` 함수를 사용하여 따옴표의 위치를 `location` 에 대입합니다. 따옴표가 element에 없는 경우 `location` 에는 -1이 대입되고 따옴표가 있는 경우에는 해당 위치가 대입 됩니다. 따라서 `location` 값이 0이상인 경우에 대해서 따옴표보다 앞에 있는 글자들을 취하여 단어 도중에 나오는 따옴표를 포함한 뒤의 글자들을 삭제합니다. 이를 위해 element를 따옴표를 기준으로 `split` 한 list의 맨 처음에 있는 element만을 취합니다.

In [2]:

```
for i in range(l):
    #'로 시작해서 '로 끝나는 단어
    if r[i][0]=="'" and r[i][len(r[i])-1]=="'":
        r[i] = r[i].strip("'")
    #단어 도중에 '가 나오는 경우 뒤의 글자들 모두 삭제
    location=r[i].find("'")
    if location >= 0:
        r[i] = r[i].split("'")[0]
```

2.2. ".com"으로 끝나는 단어는 토큰화되지 않도록 하시오.

for문을 사용하여 list `r` 에 있는 element가 ".com"으로 끝나는지 확인합니다. i번째 element의 마지막 네 글자(`len(r[i])-4`)가 ".com"인지 확인하여 그 경우에는 토큰화되지 않도록 `r[i]` 를 그대로 대입합니다.

In [3]:

```
#2-2
#.com으로 끝나는 단어는 토큰화하지 않는다는 것을 명시
for i in range(l):
    if r[i][(len(r[i])-4):] == ".com":
        r[i] = r[i]
```

2.3. 마침표(.)로 연결된 단어에서, 마침표 앞, 뒤, 및 사이에 있는 글자가 모두 1개일 경우 마침표를 삭제하고, 0개 혹은 2개 이상일 경우 토큰화되지 않도록 하시오. (* 2-3 조건을 만족하면 2-2도 자동적으로 만족하지만, .com 으로 끝나는지 확인하는 코드를 2-3과는 별도로 코드에 명시해야함.)

for문을 사용하여 list `r` 에 있는 element가 문제의 조건에 부합하는지 확인한 후 조건에 맞게 토큰화 합니다. 마침표를 기준으로 `split` 한 list를 `r_i` , 그 list의 길이를 `l_i` 라고 합니다. `all_length_one` 에는 `True` 를 대입한 후 이를 추후에 마침표 앞, 뒤 및 사이에 있는 글자가 모두 1개인 경우인지 확인하는 `bool` 변수로 사용합니다. for문을 사용하여 list `r_i` 의 각 element의 길이가 1인지 확인하여 아니면 `all_length_one` 에 `False` 를 대입하고 `break` 합니다. `break` 되지 않고 for문을 통과하게 되면 `all_length_one` 은 `True` 로 유지되고 이 경우에 마침표를 삭제합니다. `r[i]` 를 마침표를 기준으로 `split` 한 list를 중간에 element 간의 구분이 없이 `join` 하면 마침표가 삭제됩니다.

In [4]:

```
for i in range(l):
```

```
    #"."로 split한 후 각 element의 길이가 1이면 True, 아니면 False
    r_i=r[i].split(".")
    l_i=len(r_i)
    all_length_one = True
    for j in range(l_i):
        if len(r_i[j]) != 1:
            all_length_one = False
            break
    #모두 길이가 1인 경우에 한하여 "."을 지운다.
    if all_length_one == True:
        r[i]="".join(r[i].split("."))
```

In [5]:

```
print(r)
```

```
['i', 'hello', 'hello', 'imlab', 'PH.D', 'IBM', 'snu.ac.kr', '127.0.0.1', 'galago.gif',
'ieee.803.99', 'naver.com', 'gigabyte.tw', 'pass..fail']
```