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

먼저 띄어쓰기를 기준으로 주어진 `document` 를 `split` 하여 `tokens` 이라는 list를 만듭니다.

In [5]:

```
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'''
tokens = document.split(" ")
print(tokens)
```

```
["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']
```

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

for문을 사용하여 list `tokens` 에 있는 각 element가 문제의 조건을 만족시키는지 여부를 확인합니다. 먼저 따옴표로 시작해서 따옴표로 끝나는지 확인합니다. 이 조건을 만족시키는 경우 `strip` 을 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.

In [6]:

```
for (idx, token) in enumerate(tokens):
    #'로 시작해서 '로 끝나는 단어
    if token.startswith("'") and token.endswith("'"):
        token = token.strip("'")
    #단어 도중에 '가 나오는 경우 뒤의 글자들 모두 삭제
    location = token.find("'")
    if location >= 0:
        token = token[:location]
    tokens[idx] = token
print(tokens)
```

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

2.2. ".com"으로 끝나는 단어는 토큰화되지 않도록 하시오.

for문을 사용하여 list `tokens` 에 있는 element가 ".com"으로 끝나는지 확인합니다. 각 token이 ".com"인지 확인하여 그 경우에는 토큰화되지 않도록 `token` 를 그대로 대입합니다.

In [7]:

```
#.com으로 끝나는 단어는 토큰화하지 않는다는 것을 명시
for (idx, token) in enumerate(tokens):
    if token.endswith(".com"):
        tokens[idx] = token
print(tokens)
```

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

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.)

`tokens`에 있는 element가 문제의 조건에 부합하는지 확인한 후 조건에 맞게 토큰화 합니다. `all` 함수를 이용해서 split된 모든 문자열의 길이가 1인지 확인합니다. `token`를 마침표를 기준으로 `split`한 list를 중간에 element 간의 구분이 없이 `join`하면 마침표가 삭제됩니다.

In [4]:

```
for (idx, token) in enumerate(tokens):
    #"."로 split
    parts = token.split(".")
    #모두 길이가 1인 경우에 한하여 "."을 지운다.
    if all(len(part) == 1 for part in parts):
        token = "".join(parts)
    tokens[idx] = token
print(tokens)
```

```
['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']
```