

example 1: james, jason, john으로 이루어진 list a를 만든다. bob, james, sam으로 이루어진 list b를 만든다.

- 1) list a에 새로운 사람인 timmermann을 추가한다.
- 2) 이 list에 새로운 사람인 robert를 영수와 덕규사이에 추가한다.
- 3) list b의 원소들을 list a에 추가한다.
- 3) 이 list에 james가 중복되어있으니 가장 앞에 위치한 영수를 삭제한다.

In [2]:

```
a = ["james", "jason", "john"]
b = ['bob', 'james', 'sam']

a.append("timmerman")

a.insert(1,"robert")

a = a+b

a.remove('james')

['robert', 'jason', 'john', 'timmerman', 'bob', 'james', 'sam']
['bob', 'james', 'sam']
```

In [ ]:

ex2: enumerate, zip 를 활용한 exmaple

1. google, samsung, apple이 들어있는 list를 만들고 각각의 원소를 순서대로 key를 주는 dict를 만든다.
2. 그후 이를 다시 각각 회사명과 index로 이루어진 2개의 tuple로 나눠준다.

In [38]:

```
b = ["google","samsung","apple"]
mapping = list((v,i)for i, v in enumerate(b))

print mapping

name,index = zip(*mapping)
print name
print index

[('google', 0), ('samsung', 1), ('apple', 2)]
('google', 'samsung', 'apple')
(0, 1, 2)
```

ex3) math, meteorology, politics, statistics, computer\_science, psychology로 이루어진 list

로부터 각 단어의 앞글자를 key로하는 Dict를 만들어보자.

In [25]:

```
words = ['math', 'meteorology', 'politics', 'statistics', 'computer science', 'psychology']
by_letter = {}
by_letter2 = {}
for word in words:
    letter = word[0]
    if letter not in by_letter:
        by_letter[letter] = [word]
    else:
        by_letter[letter].append(word)

print by_letter

by_letter2.setdefault(letter, [ ]).append(word)

print by_letter2
```

```
{'p': ['politics', 'psychology'], 's': ['statistics'], 'm': ['math', 'meteorology'], 'c': ['computer science']}
{'p': ['psychology']}
```

### Isit comprehension 이용하기

example4: 1~20까지의 숫자에서

1. 2의 배수로 이루어진 list를 생성
2. 3의 배수로 이루어진 list를 생성
3. 2, 3의 공배수로 이루어진 list를 생성

In [28]:

```
A = [a for a in range(20) if a % 2 == 0]
B = [b for b in range(20) if b % 3 == 0]
C = [c for c in range(20) if c in A and c in B]
```

Out[28]:

```
([0, 2, 4, 6, 8, 10, 12, 14, 16, 18], [0, 3, 6, 9, 12, 15, 18], [0, 6, 12, 18])
```

example5: set comprehension 을 이용하여 1~100사이의 소수를 모두 구하여 하나의 set를 만들어준다.

\*소수: 양수인 1과 자신만을 약수로 가지는 수

In [46]:

```
no_primes = {a * multiplier for multiplier in range(2, 100) for a in range(2, 100)}

no_primes.add(1)

primes = {p for p in range(1, 100) if p not in no_primes}

print len(primes)
```

25

example6: 위의 소수 set에서 첫번째 부터 25번째 소수를 순서를 key로 가지고 값을 value로 가지는 dict를 만들고 1~100사이의 소수중 10번째 소수의 값을 출력한다.

In [52]:

```
prime_dict = {index+1:val for index, val in enumerate(primes)}
```

```
print prime_dict  
print prime_dict[10]
```

```
{1: 2, 2: 3, 3: 5, 4: 7, 5: 11, 6: 13, 7: 17, 8: 19, 9: 23, 10: 29, 11: 31, 12: 37, 13: 41, 14: 43, 15: 47, 16: 53, 17: 59, 18: 61, 19: 67, 20: 71, 21: 73, 22: 79, 23: 83, 24: 89, 25: 97}  
29
```

example7:

lambda function 에 대한 이해:

work, eat, sleep, repeat으로 이루어진 list를 만들고 lambda function을 이용해 각각의 동사의 뒤에 ing를 붙인 새로운 list를 만든다.

In [ ]:

```
verbs = ["work", "eat", "sleep", "repeat"]  
verb_converter = lambda verb: verb + "ing"  
verbs_with_ing = [verb_converter(verb) for verb in verbs]  
print(verbs_with_ing)
```