

# Intro to Computer Science

**Local Laboratory**

**\* Udacity – Intro to Computer Science**

## Introduction

### Unit 3: How to Manage Data

### Structured Data

#### String

sequence of characters

```
s = 'yabba!'
s[0] -> 'y'
s[2:4] -> 'bb'
```

#### List

sequence of anything

```
p = ['y', 'a', 'b', 'b', 'a', '!']
p[0] -> 'y'
p[2:4] -> ['b', 'b']
```

<list> -> [ <Expression>, <Expression>, ... ]

[ ]

[3]

## Quiz: Stooges

세명의 Stooges 이름에 대한 List를 값으로 갖는 `stooges` 라는 변수를 정의하시오.

Moe, Larry, Curly

```
stooges = ['Moe', 'Larry', 'Curly']
```

## Quiz: Days in a Month

주어진 변수

```
days_in_month = [ 31, 28, 31, 30, 31, 30,  
                  31, 31, 30, 31, 30, 31 ]
```

를 이용하여, 달을 나타내는 숫자를 입력으로 하여 그 달의 일 수를 리턴하는  
how\_many\_days라는 프로시저를 정의하시오.

```
how_many_days(1) -> 31
```

```
how_many_days(9) -> 30
```

## Days in a Month

code

```
def how_many_days(month):  
    days_in_month = [  
        31, 28, 31, 30, 31, 30,  
        31, 31, 30, 31, 30, 31 ]  
    return days_in_month[month - 1]  
  
print(how_many_days(1))  
print(how_many_days(9))
```

result

```
31  
30
```

code

```
def how_many_days(month):  
    days_in_month = [  
        31, 28, 31, 30, 31, 30,  
        31, 31, 30, 31, 30, 31 ]  
  
    return days_in_month[month - 1]  
  
print(how_many_days(13))
```

result

```
Traceback (most recent call last): File  
"/Users/lastland/PycharmProjects/cs101_4/test.py", line 8,  
  in <module>  
    print(how_many_days(13)) File  
"/Users/lastland/PycharmProjects/cs101_4/test.py", line 6,  
  in how_many_days  
    return days_in_month[month-1]  
IndexError: list index out of range
```

## Nested Lists

```
mixed_up = ['apple', 3, 'oranges', 27,  
            [1, 2, ['alpha', 'beta'] ] ]
```

```
beatles = [ ['John', 1940],  
            ['Paul', 1942],  
            ['George', 1943],  
            ['Ringo', 1940] ]
```

### code

```
beatles = [ ['John', 1940], ['Paul', 1942],  
            ['George', 1943], ['Ringo', 1940] ]  
  
print(beatles)  
print(beatles[3])  
print(beatles[3][0])  
print(beatles[3][1])
```

### result

```
[['John', 1940], ['Paul', 1942], ['George', 1943], ['Ringo', 1940] ]  
[ 'Ringo', 1940 ]  
Ringo  
1940
```

## Quiz: Countries

주어진 변수

```
contries = [ ['China', 'Beijing', 1350],  
             ['India', 'Delhi', 1210],  
             ['Romania', 'Bucharest', 21],  
             ['United States', 'Washington', 307]]
```

를 이용하여,

country name	capital	population (millions)
대한민국	서울	50.5
미국	워싱턴	325.7
중국	베이징	1400.0
인도	뉴델리	1380.0
일본	토쿄	126.5
러시아	모스크바	146.5
브라질	브라질리아	214.0
인도네시아	자카르타	200.0
호주	캔버라	25.5
남아프리카	프렐토리아	59.0
나이지리아	아부자	190.0
에티오피아	아디스아바바	110.0
케냐	나이로비	40.0
우간다	카무루	35.0
탄자니아	다레스살라મ	55.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.0
잠비아	루사카	8.0
부룬디	부주무라	10.0
루안다	루안다	2.5
모잠비크	마푸투	26.0
말라위	리롱위	20.

a) India의 수도를 출력하는 코드를 작성하시오.

## Countries

code

```
countries = [ ['China', 'Beijing', 1350],  
              ['India', 'Delhi', 1210],  
              ['Romania', 'Bucharest', 21],  
              ['United States', 'Washington', 309]]
```

```
print(countries[1])  
print(countries[1][1])
```

result

```
[ 'India', 'Delhi', 1210 ]  
Delhi
```



## Quiz: Relative Size

주어진 변수

```

contries = [ ['China', 'Beijing', 1350],
              ['India', 'Delhi', 1210],
              ['Romania', 'Bucharest', 21],
              ['United States', 'Washington', 307]]

```

를 이용하여,

country name	capital	population (millions)
--------------	---------	--------------------------

b) 중국의 인구는 루마니아의 몇 배인지 계산하시오.

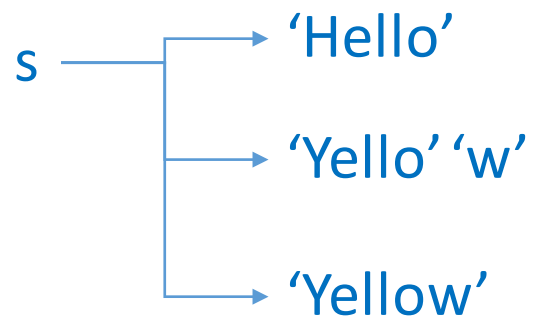
```
print( countries[0][2] / countries[2][2] )
```

## Mutation

s = 'Hello'

s = 'Yello'

s = s + 'w'



[mutate](#) 미국식 [ˈmjuːtɜrt] 영국식 [mjuːˈtɜrt] [다른 뜻\(2건\)](#) | [예문보기](#)

1. 돌연변이가 되다; 돌연변이를 만들다 2. (새로운 형태로) 변형되다

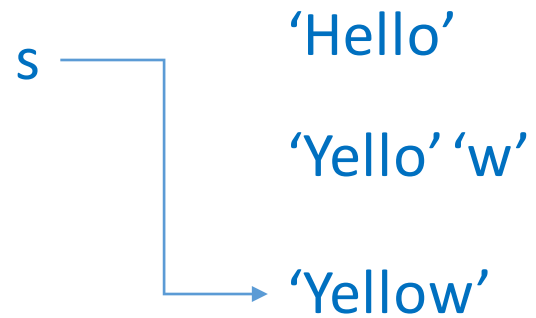
### 이탈리아어사전

[mutare](#) (mutate) TTS [예문보기](#)

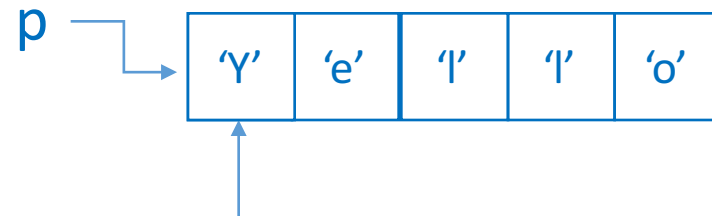
[타동사] 1. 바꾸다, 변경하다. 2. 교환하다, 변하다. [재귀동사] 1. (mutarsi) 2. 바뀌다. 3. 변모하다.

## A List of Strings

```
s = 'Hello'  
s = 'Yello'  
s = s + 'w'
```



```
p = ['H', 'e', 'l', 'l', 'o']  
p = ['Y', 'e', 'l', 'l', 'o']  
p[0] = 'Y'
```



## A List of Strings

code

```
p = ['H', 'e', 'l', 'l', 'o']
```

```
print(p)
```

```
p[0] = 'Y'
```

```
print(p)
```

```
p[4] = 'l'
```

```
print(p)
```

result

```
['H', 'e', 'l', 'l', 'o']
```

```
['Y', 'e', 'l', 'l', 'o']
```

```
['Y', 'e', 'l', 'l', 'l']
```

## Quiz: Different Stooges

이전에

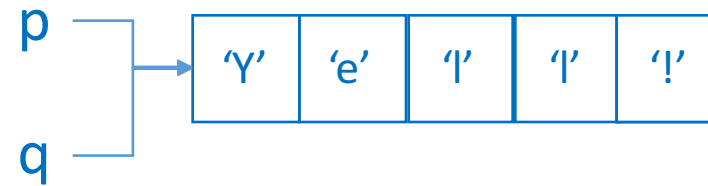
`stooges = ['Moe', 'Larry', 'Curly']`  
를 정의하였다. 이후 Stooges 영화에서 Curly가 Shemp로 교체되었다.  
stooges가 ['Moe', Larry, 'Shemp']가 되도록 하는 한 줄의 명령어를 쓰시오.

`stooges[2] = 'Shemp'`

## Yello Mutation

```
s = 'Hello'  
s = 'Yello'  
s = s + 'w'  
t = s  
s = s + 'ish'
```

```
p = ['H', 'e', 'l', 'l', 'o']  
p[0] = 'Y'  
q = p  
q[4] = 'i'
```



## Yello Mutation

code

```
s = 'Hello'
s[0] = 'Y'    immutable
```

result

```
TypeError: 'str' object does not support item assignment
```

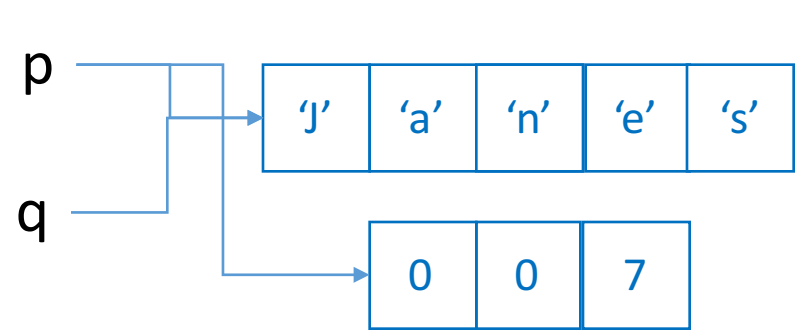
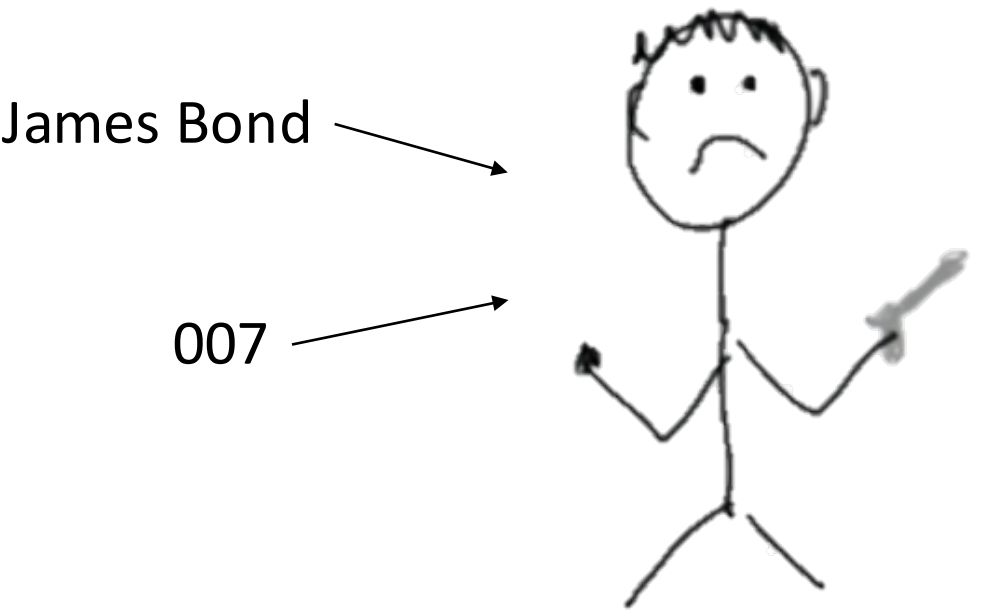
code

```
p = ['H', 'e', 'l', 'l', 'o']
q = p
print(p, q)
p[0] = 'Y'
print(q)
print(p)
```

result

```
[ 'H', 'e', 'l', 'l', 'o' ] [ 'H', 'e', 'l', 'l', 'o' ]
[ 'Y', 'e', 'l', 'l', 'o' ]
[ 'Y', 'e', 'l', 'l', 'o' ]
```

Aliasing



p[2] = 'n'  
p = [0,0,7]



## Quiz: Secret Agent Man

아래 코드 실행 후, `agent[2]` 의 값은 무엇인지 쓰시오.

```
spy = [0, 0, 7]  
agent = spy  
spy[2] = agent[2] + 1
```

8

## Quiz: Secret Agent Man

code

```
spy = [0, 0, 7]
agent = spy
print(spy, agent)
print(agent[2])

spy[2] = agent[2] + 1
print(agent, spy)
```

result

```
[0, 0, 7] [0, 0, 7]
7
[0, 0, 8] [0, 0, 8]
```

## Quiz: Replace Spy

3개의 숫자를 요소로 가지는 list를 입력으로 받아서,  
입력한 list를 수정하여 세번째 요소를 1만큼 증가시키게 하는  
`replace_spy` 프로시저를 정의하시오.

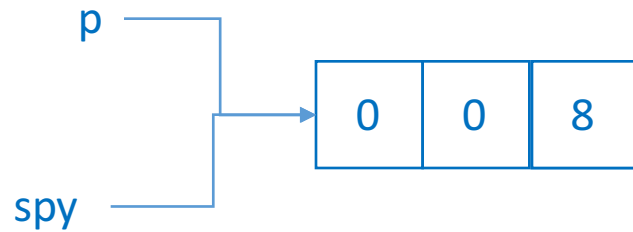
```
spy = [0, 0, 7]  
replace_spy(spy)  
print(spy)
```

```
[0, 0, 8]
```

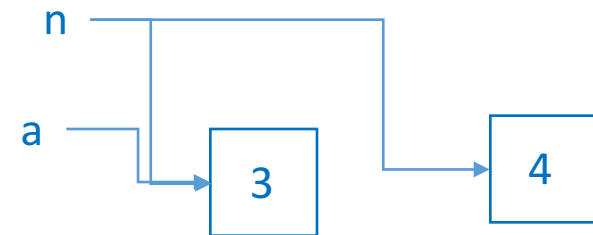
## Replace Spy

```
def replace_spy( p ) :  
    p[2] = p[2] + 1
```

```
spy = [0, 0, 7]  
→ replace_spy(spy)  
print(spy)
```



```
def inc(n):  
    n = n + 1  
    return n  
a = 3  
inc(a)
```



## List Operations

`<list>.append(<element>)`

```
stooges = ['Moe', 'Larry', 'Curly']  
stooges.append('Shemp')
```



## List Addition and Length

<list> + <list>

[0,1] + [2,3] -> [0, 1, 2, 3]

len(<list>)

len([0,1]) -> 2

len(['a',['b',['c']]]) -> 2

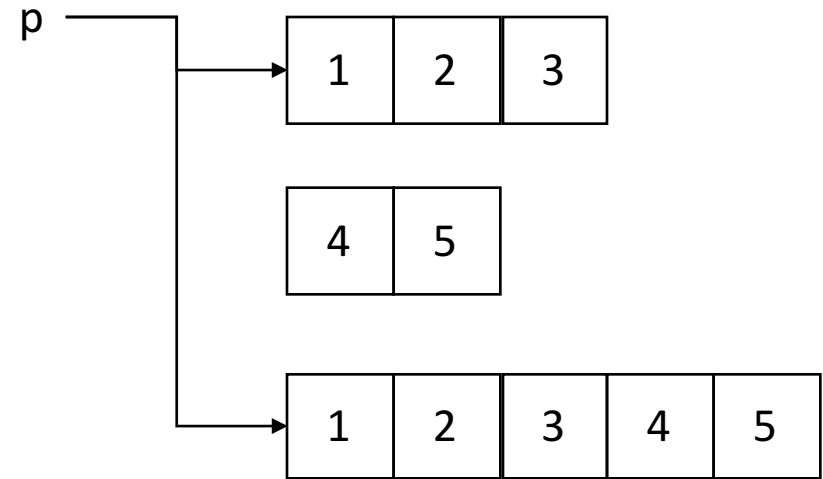
len("Udacity") -> 7

## Quiz: Len Quiz

아래 코드 실행 후, `len(p)` 의 값은 무엇인지 쓰시오.

```
p = [1, 2]  
p.append(3)  
p = p + [4, 5]
```

`len(p)` →

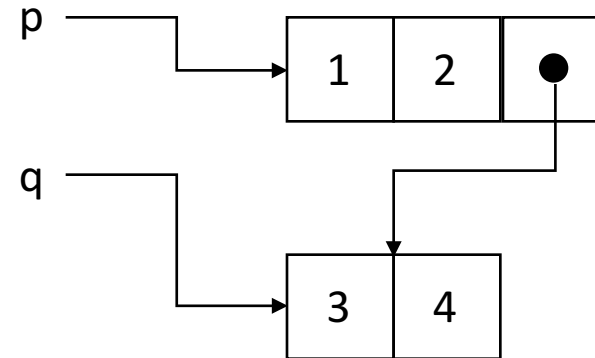


## Quiz: Append Quiz

아래 코드 실행 후, `len(p)` 의 값은 무엇인지 쓰시오.

```
p = [1, 2]  
q = [3, 4]  
p.append(q)
```

`len(p)` →





## Append Quiz

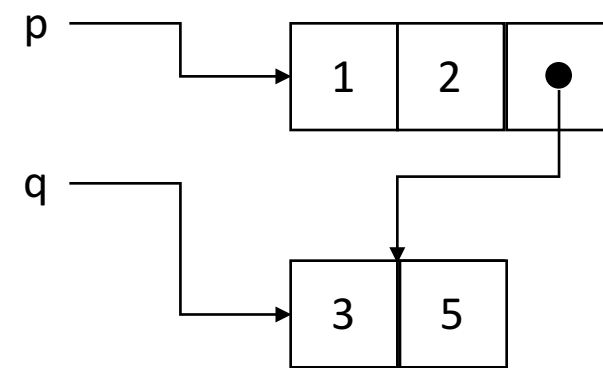
code

```
p = [1, 2]
q = [3, 4]
p.append(q)
print(p)
print(len(p))

q[1] = 5
print(p)
```

result

```
[1, 2, [3, 4]]
3
[1, 2, [3, 5]]
```



## Quiz: Loops on Lists

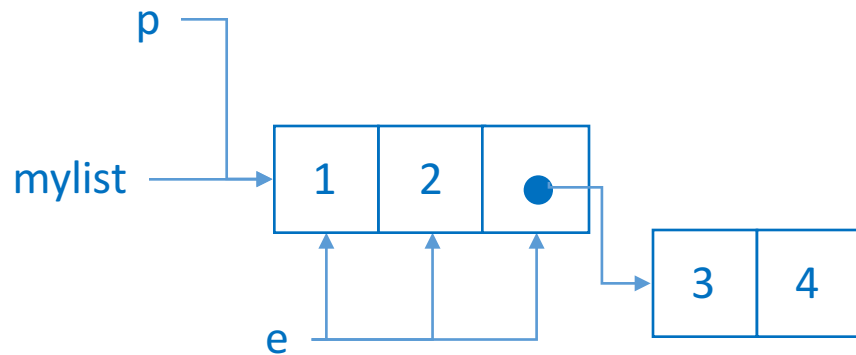
Loop ← List

```
def print_all_elements(p):  
    i = 0  
    while len(p) > i :  
        print(p[i])  
        i = i + 1
```

## For Loops

```
for <name> in <list>:  
    <Block>
```

```
def print_all_elements(p):  
    → for e in p:  
        print(e)
```



```
mylist = [1, 2, [3, 4]]  
print_all_elements(mylist)
```

## Quiz: Sum List

숫자들을 요소로 가지는 리스트를 입력으로 하여, 입력 리스트의 모든 요소의 합을 리턴하는 `sum_list` 라는 프로시저를 정의하시오.

`sum_list([1,7,4]) -> 12`

## Sum List

code

```
def sum_list(p):  
    result = 0  
    for e in p:  
        result = result + e  
    return result  
  
print(sum_list([1, 4, 7]))  
print(sum_list([]))
```

result

```
12  
0
```

## Quiz: Measure Udacity

문자열들을 요소로 가지는 리스트를 입력으로 하여, 요소 중에 첫번째 문자가 대문자 'U'인 요소들의 개수를 리턴하는 `measure_udacity` 라는 프로시저를 정의하시오.

```
measure_udacity(['Dave', 'Sebastian', 'Katy']) -> 0
```

```
measure_udacity(['Umika', 'Umberto']) -> 2
```

## Measure Udacity

code

```
def measure_udacity(U):  
    count = 0  
    for e in U:  
        if e[0] == 'U':  
            count = count + 1  
    return count  
  
print(measure_udacity(['Dave', 'Sebastian', 'Katy']))  
print(measure_udacity(['Umika', 'Umberto']))  
measure_udacity(['udacity', 'United States', 'union' , 'U2'])
```

result

```
0  
2  
2
```

## Quiz: Find Element

리스트와 임의의 타입을 가진 값 두개의 입력으로, 입력된 두번째 값과 동일한 값을 가진 리스트의 요소 중 첫번째로 매칭되는 요소의 인덱스를 리턴하는 `find_element` 라는 프로시저를 정의하시오. 만약 동일한 값을 가지는 요소가 없을 경우, `-1`을 리턴하시오.

`find_element([1, 2, 3], 3) -> 2`

`find_element(['alpha', 'beta'], 'gamma') -> -1`



## Find Element

```
def find_element(p, t):  
    i = 0  
    while i < len(p):  
        if p[i] == t:  
            return i  
        i = i + 1  
    return -1
```

```
def find_element(p, t):  
    i = 0  
    for e in p:  
        if e == t:  
            return i  
        i = i + 1  
    return -1
```

## Index

`<list>.index(<value>)`

만약 `<value>`가 `<list>`에 존재한다면,  
`<value>`가 존재하는 첫번째 위치를  
리턴

code

```
p = [0, 1, 2]
print(p.index(2))
p = [0, 1, 2, 2, 2]
print(p.index(2))
p = [0, 1, 2]
print(p.index(3))
```

result

```
2
2
ValueError: 3 is not in list
```

## Index

<value> in <list>

만약 <value>가 <list>에 존재하면,  
True 값을 출력  
존재하지 않으면 False 값을 출력

3 in p

code

```
p = [0, 1, 2]  
print(3 in p)  
print(1 in p)
```

result

```
False  
True
```

## Quiz: Index

이전에 정의한 `find_element` 프로시저를 `index` 메소드를 이용하여 새로 정의하시오.

`<list>.index(<value>)`

만약 `<value>`가 `<list>`에 존재한다면,  
`<value>`가 존재하는 첫번째 위치를  
리턴

`<value> in <list>`

만약 `<value>`가 `<list>`에 존재하면,  
True 값을 출력  
존재하지 않으면 False 값을 출력

`<value> not in <list>`  $\equiv$  `not <value> in <list>`

## Quiz: Index

```
def find_element(p, t):  
    if t in p:  
        return p.index(t)  
    else:  
        return -1
```

```
def find_element(p, t):  
    if t not in p:  
        return -1  
    else:  
        return p.index(t)
```

## Quiz: Union

두개의 리스트를 입력으로 받아서, 두 리스트의 합집합을 만드는 `union` 이라는 프로시저를 정의하시오.

```
a = [1, 2, 3]
b = [2, 4, 6]
union(a, b)
print(a) -> [1, 2, 3, 4, 6]
print(b) -> [2, 4, 6]
```

```
def union(p, q):
    for e in q:
        if e not in p:
            p.append(e)
```

## Pop

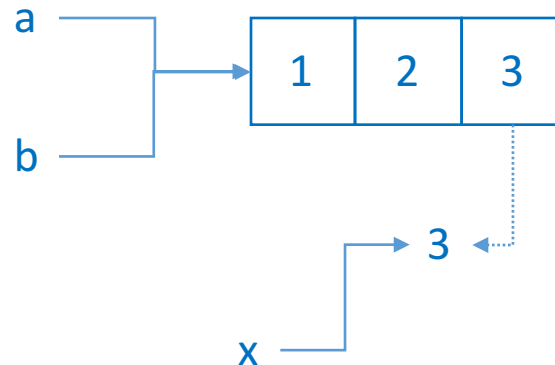
`<list>.pop() -> element`

list의 마지막 요소를 지우고(mutate) 그 값을 반환(return)함

```
a = [1, 2, 3]
```

```
b = a
```

```
x = a.pop()
```



## Quiz: Pop Quiz

`p` 는 적어도 두개의 요소를 가지는 리스트라고 가정한다.  
아래의 코드들 중 코드의 마지막에서 `p`의 값이 변하지 않는 것들을 모두 고르시오.

☐ `p.append(3)`  
`p.pop()`

☐ `x = p.pop()`  
`p.append(x)`

☐ `x = p.pop()`  
`y = p.pop()`  
`p.append(x)`  
`p.append(y)`

☐ `x = p.pop()`  
☐ `y = p.pop()`  
☐ `p.append(y)`  
☐ `p.append(x)`