

```
In [2]: til = "Today I Learned"
        til
```

Out[2]: 'Today I Learned'

```
In [3]: til.lower()
```

Out[3]: 'today i learned'

```
In [4]: til.upper()
```

Out[4]: 'TODAY I LEARNED'

## list

```
In [5]: lang = []
        lang
```

Out[5]: []

```
In [6]: lang.append("python")
        lang.append("java")
        lang.append('c')
        lang
```

Out[6]: ['python', 'java', 'c']

```
In [7]: lang[0]
```

Out[7]: 'python'

```
In [8]: lang[1]
```

Out[8]: 'java'

```
In [9]: lang[-1]
```

Out[9]: 'c'

## Control Flow

```
In [10]: for i in lang:
          print(i)
```

```
python
java
c
```

In [12]:

```
for i in lang:
    if i == 'python':
        print('python')
    else:
        print('기타')
```

python  
기타  
기타

```
In [22]: for i in range(5): # 기본적으로 0부터 출력
          print (i)
```

0  
1  
2  
3  
4

```
In [23]: for i in range(1,10,2): #1부터 9까지 간격은 2
          print (i)
```

1  
3  
5  
7  
9

```
In [26]: for i, val in enumerate (lang): # i는 index val은 그 인덱스에 들어있는 값
          print(i,val)
```

0 python  
1 java  
2 c

```
In [27]: address = " 경기도 성남시 분당구 불정로 6 NAVER 그린팩토리 16층 "
          address
```

Out[27]: ' 경기도 성남시 분당구 불정로 6 NAVER 그린팩토리 16층 '

```
In [28]: address.strip()
          #앞뒤 공백을 제거
          #데이터 전처리 시 주로 사용
```

Out[28]: '경기도 성남시 분당구 불정로 6 NAVER 그린팩토리 16층 '

```
In [29]: address
          #출력결과를 보면 원본은 바뀌지 않음
          #원본 바꾸려면 재대입 해줘야함
```

Out[29]: ' 경기도 성남시 분당구 불정로 6 NAVER 그린팩토리 16층 '

```
In [30]: len(address)
```

Out[30]: 35

```
In [31]: address.split()
```

```
#디폴트 값은 공백
#출력값이 list형식으로 나온 그러면 list관련 함수 사용 가능할 듯? append해보니까 결과는
```

Out[31]: ['경기도', '성남시', '분당구', '불정로', '6', 'NAVER', '그린팩토리', '16층']

```
In [35]: address_list = address.split()
address_list
```

Out[35]: ['경기도', '성남시', '분당구', '불정로', '6', 'NAVER', '그린팩토리', '16층']

```
In [36]: len(address_list)
```

Out[36]: 8

```
In [39]: #슬라이싱으로 문자 가져오기
address = address.strip()
address[:2]
```

Out[39]: '경기'

```
In [40]: #특정 문자로 시작하는지 여부 확인
address.startswith("경기")
#원본은 False나오고 앞뒤 공백을 제거한거는 True가 나온 왜지;;
#이거는 시작이었음 " 경기" != "경기"
```

Out[40]: True

```
In [44]: "경기" in address
```

Out[44]: True

```
In [41]: gu = address_list[2]
street = address_list[3]
```

```
In [42]: address_list[-1]
```

Out[42]: '16층'

```
In [43]: " ".join(address_list)
#리스트를 공백으로 구분하여 전부 합쳤음
#" " 쓰면 공백 없이, "." 마침표로 구분. 이것저것 가능
```

Out[43]: '경기도 성남시 분당구 불정로 6 NAVER 그린팩토리 16층'

```
In [45]: "경기" in address_list
#list는 원소를 비교할때 text가 전부 매칭이 되어야함
```

Out[45]: False

```
In [46]: gu in address_list
```

Out[46]: True

In [ ]: