# 비교과 겨울 방학 특강 파이썬 4일 - 수업자료



양 자영



# 문자열 다루기

## 문자열

- 작은 따옴표나 큰 따옴표로 표현
- 반드시 같은 따옴표로 열고 닫아야 함

"Hello World" = 'Hello World' 같은 또한

```
msg = 'She said 'Hi' '
print(msg)

msg = "She said 'Hi' "
print(msg)

She said 'Hi'

She said 'Hi'
```

#### (추가로 해보기)

• 다음 문자열을 변수에 저장하여 출력하시오. He said, "Hello."

## 문자열



#### • 여러 줄 문자 표현 방법

```
poem = "이렇게 정다운₩n너 하나 나 하나는₩n어디서 무엇이 되어₩n다시 만나랴." print(poem)
```

개행문자("\n") 사용

이렇게 정다운 너 하나 나 하나는 어디서 무엇이 되어 다시 만나랴.

```
poem = "'이렇게 정다운
너 하나 나 하나는
어디서 무엇이 되어
다시 만나랴.'"
```

print(poem)

작은 따옴표 3개("") 또는 큰 따옴표 3개(""")로 열고 닫기

# 문자열 연산( + \* )



```
Python 3.8.2 Shell
                                                                                     X
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = "문자열"
>>> b = "현산
>>> a + b
'문자열연산'
>>>
Python 3.8.2 Shell
                                                                                      X
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> "문자열" * 10
'문자열문자열문자열문자열문자열문자열문자열문자열문자열문자열
>>>
>>> "'문자열'" * 10
>>>
                                                                                  Ln: 9 Col: 4
```

# 문자열 변수 확인해보기



Ln: 12 Col: 4

```
Python 3.8.2 Shell
                                                                                                   \times
                                                                                             \Box
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = "Python'
                      int, float, double long - 숫자형
>>> type(a)
                      str - 문자형
<class 'str'>
>>>
                                                                                              Ln: 7 Col: 4
>>> a = "Python is easy"
>>> b = "to handle strings"
                                변수끼리 + 사용하기
>>> a+b
'Python is easyto handle strings
>>>
>>> a = "문자열"
>>> a * 3
                     변수끼리 * 사용하기
'문자열문자열문자열'
>>>
```

# 문자열 변수 확인해보기



```
>>> a = """
First line
Second line"""
>>> print(a)

First line
Second line
>>>
```

#### 숫자 변수와 문자열 변수 연결하기

```
>>> age = 21
>>> line = "살 입니다."
>>> age + line
Traceback (most recent call last):
   File "<pyshell#22>", line 1, in <module>
        age + line
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> str(age) + line
'21살 입니다.'
>>>
```

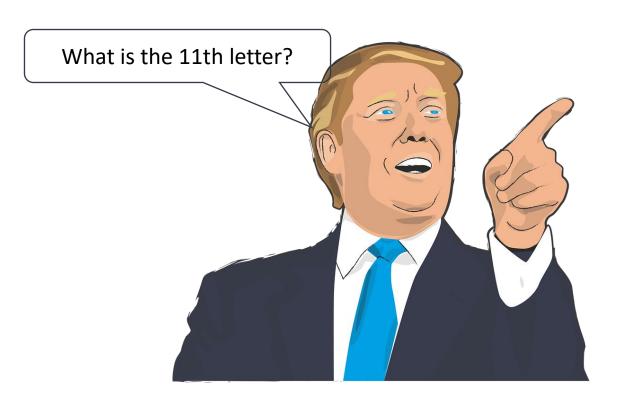
# 문자열 인덱싱



S	t	r	i	n	g	S	i	n		P	y	t	h	0	n
[0]							[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]

#### 첫번째 글자는 0번째





```
>>> a = "Strings in Python"
>>> a[0]
'S'
>>> a[7]
>>> a[16]
>>> a[-2]
>>> a[-15]
>>> a[-17]
>>>
```

# 문자열 슬라이싱(자르기)



S	t	r	i	n	g	S		i	n		P	y	t	h	0	n
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]

n g s i n

# 문자열 슬라이싱(자르기)



Ln: 7 Col: 4

```
File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>> a = "Hello Python"

>>> a[0:5]

'Hello'

O 1 2 3 4
```

a[0:5] -> a 변수의 0번째 이상, 5번째 미만의 문자열 추출 -> 0 <= a < 5

# 문자열 슬라이싱(자르기, 음수)



```
>>> a = "Hello Python"
>>> a[-1:-6]
''
>>> a[-6:-1]
'Pytho'
>>>
```

#### 음수로 끝까지 자르고 싶을때는

## 실습 – 온점 판별하기



• 문장을 입력받은 후 마지막에 온점(.)이 있는 확인하여 판별 출력하기

# 문자열 함수(1)



#### 문자열 분리하기 - split

```
>>> a = "arc bug can dream elephant"
>>> a.split(" ")
['arc', 'bug', 'can', 'dream', 'elephant']
>>> a = "arc-bug-can-dream-elephant"
>>> b = a.split("-")
>>> b
['arc', 'bug', 'can', 'dream', 'elephant']
>>> b
```

#### 문자열 공백제거-strip

```
>>> a="아 정말 제가 할게요."
>>> a.strip("아")
' 정말 제가 할게요.'
>>> a=" 정답은 사과 입니다. "
>>> a.strip()
'정답은 사과 입니다.'
```

# 문자열 함수(2)



#### 문자열 바꾸기 .replace

슬라이스 연산으로도 가능 ex) a[:2]+"님"+a[3:]

```
>>> a = "사장니도 새해 복 많이 받으세요."
>>> a.replace("니","님")
'사장님도 새해 복 많이 받으세요.'
>>>
```

#### 문자열 대소문자 바꾸기

```
>>> a = "python"
>>> B = "PYTHON"
>>> a.upper()
'PYTHON'
>>> B.lower()
'python'
>>> a.capitalize()
'Python'
```

# 문자열 함수(3)



#### 문자열 개수 세기.count()

```
>>> a = "간창공창공장장" 
>>> a.count("장")
4
```

#### 문자열 위치 찾기 .find() .index()

a.index("Boys")

ValueError: substring not found

```
>>> a = "Ladies and Gentlemen, Welcome to Python world"
>>> a.find("\")
22
>>> a.index("\")
22
```

```
>>> a = "Ladies and Gentlemen, Welcome to Python world"
>>> a.find("Boys")
-1
>>> a.index("Boys")
Traceback (most recent call last):
File "<pyshell#6>", line 1, in <module>
```

٠,

Ln: 17 Col: 4

# 문자열 함수(4) -문자열 값 판별하기 .is



#### 숫자인지 확인 .isdigit() / 문자확인 .isalpha()

```
>>> a = "12345"
>>> a.isdigit()
True
>>> a.isalpha()
False
```

#### 대문자 판별 .isupper() / 소문자 판별 .islower()

```
>>> a = "hello"
>>> a.islower()
True
>>> a = "HELLO"
>>> a.isupper()
```

#### 공백 여부 .issapce()

True

```
>>> a = " "
>>> a.isspace()
True
>>> a = " 5a "
>>> a.isspace()
False
>>>
```

## 실습 - 자동 기사



몇 가지 입력(승리팀, 패배팀, 스코어, 경기유형, MVP선수) 으로 자동으로 기사 완성

경기 유형1, "불꽃 튀는" 경기 유형2, "단조로운" 경기 유형 3, "일방적인"

# 실습 – 성별 판별

• 주민번호를 입력받아 여자인지 남자인지 구별하는 프로그램을 작성하시오.

(단, 주민번호 자릿수가 부족하거나 성별 부분에 1, 2, 3, 4가 아닌 다른 것이 오는 경우는 오류 메시지를 출력할 것

- 1. jumin변수에 주민번호를 입력받는다.
- 2. 7번째 글자를 number 변수에 저장한다.
- 3. number가 1또는 3이면 '남자'라고 출력한다.
- 4. number가 2또는 4이면 '여자'라고 출력한다.
- 5. 그외면 잘못된 번호라고 출력한다.

주민번호를 입력하세요 : 901213-1255555 남자입니다

주민번호를 입력하세요 : 901010-2521313 여자입니다

주민번호를 입력하세요 : 031214-3111111 남자입니다

주민번호를 입력하세요 : 031010-4222222 여자입니다

# 실습 – 연도로 마스크 구입일



출생연도를 입력받아 마스크를 구입할 수 있는 날을 출력하는 프로그램을 작성하시오

출생연도	0.5	1 6	2.7	2 0	4.0
끝자리	0,5	1,6	2,7	3,8	4,9
요일	월	화	수	목	금

# 실습 – 암호문 만들기

- 암호 프로그램 만들기1
  - 사용자에게 입력받은 평문을 거꾸로 출력하는 암호문을 만들어보자.

```
str = input("평문 : ")
crypto =
print(crypto)
```

평문 : abcdefg gfedcba

평문 : 정문에서만나요 요나만서에문정



# 리스트와 튜플

# 여러 개의 데이터 다루기



• 일주일간의 최고 기온을 프로그램에서 다루려면?

월	화	수	목	금	토	일
20	21	25	24	27	22	23

temp1 = 20 temp2 = 21 temp3 = 25 temp4 = 24 temp5 = 27 temp6 = 22 temp7 = 23

- 일주일이 아니라 일년 간의 최고 기온을 다루려면?
  - 많은 양의 자료를 사용해야 할 경우 일일이 변수로 선언하여 처리하는 것은 비효율적
  - 여러 개의 데이터를 묶어서 저장하고 처리하자 → 리스트

temp = [20, 21, 25, 24, 27, 22, 23] 리스트명 = [항목1, 항목2, 항목3, ...]

# 리스트(list)

- 여러 개의 값을 가지는 변수
- 대괄호[] 안에 자료들을 넣어 선언
- 리스트 예

```
empty = []# 빈 리스트score = [80, 90, 60, 70]# 숫자 리스트name = ["홍길동", "박문수", "김삿갓"]# 문자열 리스트num_str = [1, 2, 3, "아이언맨", "토르"]# 수와 문자열로 구성된 리스트list_inlist = [[1, 3, 5], [2, 4, 6]]# 리스트를 포함하는 리스트(2차원 리스트)
```

리스트명 = [항목1, 항목2, 항목3, ...]

# 리스트의 생성



• 리스트의 생성

```
slist = ["영어", "수학", "사회", "과학"]
print(slist)

cart = []

cart.append("사과")
cart.append("세제")
print(cart)
```

```
alist = list()
blist = list(range(5))
clist = list(range(1, 5))

(3) dlist = list(range(1, 5, 2))
print(alist)
print(blist)
print(blist)
[0, 1, 2, 3, 4]
print(clist)
[1, 2, 3, 4]
print(dlist)
[1, 3]
```

# 리스트 항목에 접근과 변경

- 리스트 항목의 접근
  - 인덱스(index) : 리스트의 항목 의 위치를 알려주는 번호
  - 리스트의 인덱스는 0번부터 시 작(음수 인덱스도 가능)

slist = ["영어", "수힉	-", "사호	l", "과학"]
print(slist[0])	영어	인덱스
print(slist[1])	수학	[0] [1] [2] [3]
print(slist[2])	사회	"영어" "수학" "사회" "과학"
print(slist[-2])	사회	[-4] [-3] [-2] [-1]

• 리스트 항목의 변경

리스트명[인덱스] = 값

```
slist = ["영어", "수학", "사회", "과학"]
slist[10] = "음악"
print(slist) 인덱스 범위를 벗어나지 않도록 주의!
```

Traceback (most recent call last):

File "C:/Users/khmoon/Desktop/test.py", line 2, in <module> slist[10] = "음악"

IndexError: list assignment index out of range

# 리스트와 반복문의 궁합



• 다음 리스트 내의 원소 값을 출력하는 프로그램을 작성하시오.



```
num = [100, 96, 209, 22, 30, 117]

for i in num :

print(i, end = " ")
```

```
num = [100, 96, 209, 22, 30, 117]
for i in range(6) :
    print(num[i], end = " ")
```

# 실습 – 별표 출력하기



• 리스트에 저장된 값으로 다음과 같이 별표를 출력하여 막대 그래프를 그리는 프로그램을 작성하시오.

```
numbers = [20, 1, 12, 9, 18]
```

- 1. 리스트의 개수만큼 반복한다
  - 1) 리스트의 값에 "\*"를 곱해서 화면에 출력한다.

# 리스트 슬라이싱(리스트 자르기)

- stop 인덱스 항목은 추출에 포함되지 않음
- 처음 항목부터 추출할 때는 start 생략 가능
- 마지막 항목까지 추출할 때는 stop 생략 가능
- step 생략 가능(기본값 1)
- step이 음수이면 뒤에서부터 추출

```
['영어', '사회']
['영어', '수학']
['영어', '수학']
['영어', '수학']
['수학', '사회', '과학']
['영어', '수학', '사회', '과학']
['과학', '사회', '수학', '영어']
```

리스트명[start:stop:step]

```
slist = ["영어", "수학", "사회", "과학"]
print(slist[0:3:2])
print(slist[0:2])
print(slist[:2])
print(slist[1:])
print(slist[:])
```

# 기타 숫자형 리스트 도구



min(a)	리스트 a에 포함된 원소 중 가장 작은 원소를 찾아줌
max(a)	리스트 a에 포함된 원소 중 가장 큰 원소를 찾아줌
len(a)	리스트 a에 포함된 원소의 개수를 알려줌
sum(a)	리스트에 포함된 원소의 합을 계산함

```
>>> a
[9, 7, 6, 5, 4, 3, 3, 2, 1]
>>>
min(a)
1
>>> max(a)
9
>>> len(a)
9
>>> sum(a)
40
```

평균?

# 실습 - 리스트



- 월요일부터 일요일까지 온도
- temp =[20, 21, 25, 24, 27, 22, 23]
- 월요일의 온도는?
- 주말(금, 토, 일요일)의 온도는?
- 수요일에 온도가 잘못 기입되었다. 수요일 온도는 26도. 이를 반영하시오.
- 일주일 전체 가장 높은 온도는?
- 일주일의 평균 온도은?

# 리스트의 연산



#### • + : 리스트 연결

```
num1 = [1, 2, 3]
num2 = [4, 5, 6]
plus_num = num1 + num2
print(plus_num)
```

[1, 2, 3, 4, 5, 6]

#### • \* : 리스트 반복

```
num1 = [1, 2, 3]
mul_num = num1 * 3
print(mul_num)
```

[1, 2, 3, 1, 2, 3, 1, 2, 3]

# 리스트 전용함수



# • 리스트 변수 뒤에 점(.) 찍고 호출

리스트 관련 전용함수	설명
append(x)	리스트의 마지막 위치에 x를 추가
insert(index, x)	리스트의 index 위치에 x를 삽입
remove(x)	리스트에서 처음 발견된 x를 삭제
non(indov)	지정한 위치에서 항목을 삭제하고 반환. Pindex를 생략하면 리스트의 마지막 항목을 삭제
pop(index)	하고 반환
clear()	리스트의 모든 항목을 삭제
index(x)	리스트에 x가 있다면 그 위치를 반환
sort()	리스트 항목들을 오름차순으로 정렬. reverse 인자의 값이 True이면 내림차순으로 정렬
count(x)	리스트에 포함된 x의 개수를 반환
extend([x1,x2])	리스트 끝에 [x1,x2] 리스트를 붙여서 확장
reverse()	리스트 순서 뒤집기
copy()	리스트 복사

# 예제



• 다음은 회의 참석자 명단입니다. 참석자 명단을 오름차순과 내림차순으로 정렬해 봅시다.

```
name = ["차은우", "아이유", "장원영", "홍길동", "강감찬"]
```

```
오름차순: ['강감찬', '아이유', '장원영', '차은우', '홍길동']
내림차순: ['홍길동', '차은우', '장원영', '아이유', '강감찬']
```

# 리스트 항목의 추가



• append(x) 함수 - 리스트의 마지막 위치에 추가

```
slist = ["영어", "수학", "사회", "과학"]
slist.append("물리")
print(slist)
['영어', '수학', '사회', '과학', '물리']
맨 뒤에 추가
```

• insert(index, x) 함수 - 리스트의 인덱스 위치에 삽입

```
slist = ["영어", "수학", "사회", "과학"]
slist.insert(2, "체육")
print(slist)
['영어', '수학', '체육', '사회', '과학']
```

# 실습 – 장바구니 추가



장바구니 항목에 살 것을 계속 추가한 후, "그만"을 입력하면 장바구니 항목을 출력해 주는 프로그램을 작성하시오.

- 1. cart를 빈 리스트로 초기화한다.
- 2. 무한반복한다.
  - ① 사고싶은 항목을 product변수에 입력받는다.
  - ② Product변수의 값이 "그만"이면 반복문을 멈춘다.
  - ③ 아니면 cart리스트에 추가한다.
- 3. cart를 출력한다.

사고 싶은 항목을 입력하세요 : 사과

사고 싶은 항목을 입력하세요 : 딸기

사고 싶은 항목을 입력하세요 : 바나나

사고 싶은 항목을 입력하세요 : 그만

['사과', '딸기', '바나나']

# 리스트 항목의 삭제



• remove(x) 함수 - 리스트에서 처음 발견된 해당 항목을 삭제

```
slist = ["영어", "수학", "사회", "과학", "사회"]
                                                          ['영어', '수학', '과학', '사회']
slist.remove("사회")
print(slist)
slist = ["영어", "수학", "사회", "과학"]
                                                Traceback (most recent call last):
                                                 File "C:/Users/khmoon/Desktop/test.py", line 8, in <module>
slist.remove("체육")
                                                  slist.remove("체육")
                                                ValueError: list.remove(x): x not in list
print(slist)
slist = ["영어", "수학", "사회", "과학"]
if "체육" in slist :
                                                      ['영어', '수학', '사회', '과학']
   slist.remove("체육")
print(slist)
```

#### 리스트 항목의 삭제

- pop(index) 함수
  - 지정한 위치에서 항목을 삭제하고 반환. 등 하면 리스트의 마지막 항목을 삭제하고 반환

- clear() 함수
  - 리스트의 모든 항목을 삭제

```
slist = ["영어", "수학", "사회", "과학"]
subject = slist.pop()
                                          ['영어', '수학', '사회']
                                          과학
print(slist)
print(subject)
slist = ["영어", "수학", "사회", "과학"]
                                          ['영어', '수학', '과학']
subject = slist.pop(2)
                                          사회
print(slist)
print(subject)
slist = ["영어", "수학", "사회", "과학"]
slist.clear()
print(slist)
```

#### 리스트 항목의 탐색



• index(x) 함수 – 첫번째 일치 항목의 인덱스 반환

```
slist = ["영어", "수학", "사회", "과학", "사회"]
number = slist.index("사회")
print(number)
slist = ["영어", "수학", "사회", "과학"]
                                             Traceback (most recent call last):
                                              File "C:/Users/khmoon/Desktop/test.py", line 2, in <module>
number = slist.index("체육")
                                               number = slist.index("제육")
                                             ValueError: '체육' is not in list
print(number)
slist = ["영어", "수학", "사회", "과학"]
if "체육" in slist:
   number = slist.index("제육")
   print(number)
```

#### 리스트의 기타 함수



• 리스트의 특정 항목의 개수 – count(x) 함수

```
grade = ["A", "B", "C", "A", "B", "A"]

number = grade.count("A")

print("A 등급 수 :", number)
```

• 리스트의 확장 – extend([x1, x2]) 함수

```
grade1 = ["A", "B", "C"]
grade2 = ["A", "B", "A"] ['A', 'B', 'C', 'A', 'B', 'A']
grade1.extend(grade2)
print(grade1)
```

#### 리스트 + 연산자와 extend 함수의 차이점

- list1 + list2는 두개의 리스트를 연결해서 새로운 리스트 list3을 반환
- list1.extend(list2)는 list1객체에 list2를 붙이고, list1 객체를 그대로 사용

#### 실습 – 학생 명단관리



리스트를 이용해서 학급의 학생 명단을 관리하는 프로그램으로, 총 7개의 시나리오에 따라 프로그램을 작성하고 결과를 확인하는 프로그램을 작성하시오.

- 학생수가 5명인 출석부 리스트를 만든다.(이승엽, 박찬호, 신석기, 배민규, 전민수)
- 2. 가나다 순으로 정렬하여 보여준다.
- '박찬호가 전학을 가게 되었다. 출석부에서 삭제하고 전체 학생과 학생수를 출력한다.
- 4. 선생님을 돕는 학생 2명을 앞에서부터 데려와서 출 력한다.
- 5. 새로운 학생이 전학을 왔다. 이름이 '홍길동' 이다.
- 6. 자리를 바꾸려고 역순으로 뒤집었다.
- 7. 전민수 학생이 전민규로 이름을 개명했다.

```
1 : ['이승엽', '박찬호', '신작기', '배민규', '전민수']

2 : ['박찬호', '배민규', '신석기', '이승엽', '전민수']

3 : ['배민규', '신석기', '이승엽', '전민수']

4 : ['배민규', '신석기']
```

5 : ['배민규', '신석기', '이승엽', '전민수', '홍길동']

6: ['홍길동', '전민수', '이승엽', '신석기', '배민규']

7: ['홍길동', '전민규', '이승엽', '신석기', '배민규']

### 실습 – 학생 성적 리스트



- 다음은 우리 반 5명에 대한 성적(score)과 이름(name) 리스트이다.
- 80점 이상이면 "우수"일 때 성적이 우수인 학생들의 이름을 출력하는 프로그램을 작성하시오.

```
score = [80, 90, 70, 60, 100]

name = ["차은우", "아이유", "장원영","홍길동","강감찬"]

print('--- 성적 우수자 ---')
```

- 1. 리스트의 수만큼 반복한다.
  - ① 점수가 80점 이상이면 이름과, 성적을 출력한다.

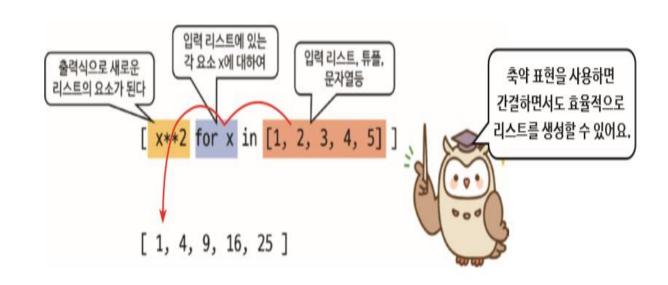
```
-- 성적 우수자 ---
차은우 80
아이유 90
강감찬 100
```

# 리스트 함축(list comprehension)



• 리스트 함축 표현을 사용하면 입력 리스트의 각 항목을 하나하나 방문 하며 지정된 연산을 하고, 연산의 결과로 생성된 항목값을 가지는 리스 트를 생성할 수 있다.

```
s = []
for x in [1, 2, 3, 4, 5]:
    s.append(x**2)
```



# 리스트 함축(list comprehension)

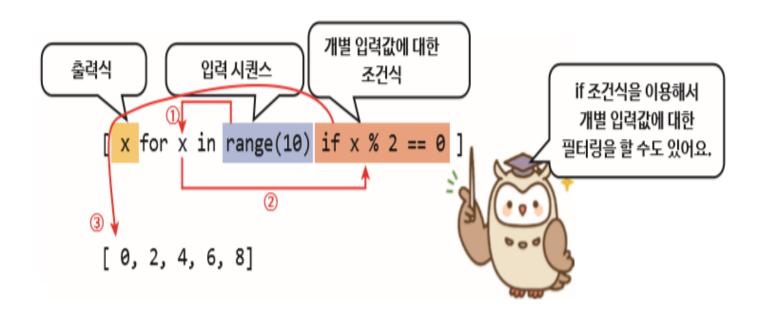


```
>>> [x for x in range(10)] # 0에서 9까지 숫자를 포함하는 리스트
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> [x * x for x in range(10)] # 0에서 9까지 숫자의 제곱 값
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
>>> st = 'Hello World'
>>> [x.upper() for x in st] # 문자열 각각에 대해 upper() 메소드 적용
['H', 'E', 'L', 'L', 'O', ' ', 'W', 'O', 'R', 'L', 'D']
```

만일 ['welcome', 'to', 'the', 'python', 'world']라는 항목을 가지는 리스트 a에서 첫번째 문자를 대문자로 바꾸어보자.

# 리스트 함축(list comprehension) - 조건





```
>>> [x for x in range(10) if x % 2 == 0]
[0, 2, 4, 6, 8]
>>> [x**2 for x in range(10) if x % 2 == 0] # 출력식에 제곱을 할 수 있다
[0, 4, 16, 66, 64]
```

# 리스트 함축(list comprehension) - 조건

```
s = ["Hello", "12345", "World", "67890"]
numbers = [x for x in s if x.isdigit()]
print(numbers)
```

```
>>> [int(x) for x in input('정수를 여러개 입력하세요 : ').split()]
정수를 여러개 입력하세요 : 100 200 300
[100, 200, 300]
>>> [int(x) for x in input('정수를 여러개 입력하세요 : ').split() if x.isdigit()]
정수를 여러개 입력하세요 : 100 이백 300 400
[100, 300, 400]
```



# 튜플

# 튜플(tuple)



- 리스트와 비슷하게 데이터를 묶어서 처리하는 컨테이너 자료형임
- 튜플에 포함된 <u>아이템을 수정할 수 없음</u>
- 구조가 단순, 리스트에 비해 접근 속도가 빠름
- 프로그램이 실행되는 동안 값을 유지해야 한다면 튜플을, 수시로 값을 변경해야 하면 리스트 사용

다음 중 튜플로 선언하는 것이 가장 좋은 것은

- ① 내가 좋아하는 색상: 빨간색, 노란색, 보라색
- ② 내가 좋아하는 음식 : 김치, 라면, 짜장면
- ③ 학생의 혈액형: A형, B형, O형, AB형
- ④ 학생 명단 : 홍길동, 홍철수, 강감찬

# 튜플(tuple) 선언

- 소괄호()를 사용하여 선언함
- 튜플은 1개의 요소만을 가질 때는 요소 뒤에 콤마(,)를 만드시 붙여야 함 (예) t3 = (1, )
- 괄호()를 생략해도 무방함(예)t5 = 1, 2, 3
- tuple함수를 이용해 리스트를 튜플로 변환

```
t1 = ()

t2 = tuple()

t3 = (1,)

t4 = (1,2,3)

t5 = 1,2,3

t6 = tuple([1,2,3])

t7 = ('a', 'b', ('a','b'))
```

```
t1 ()
t2 ()
t3 (1,)
t4 (1, 2, 3)
t5 (1, 2, 3)
t6 (1, 2, 3)
t7 ('a', 'b', ('a', 'b'))
```

### 예제



• '학생' 이 저장된 튜플을 생성하라.

• 1 부터 99까지의 정수 중 짝수만 저장된 튜플을 생성하라.

# 튜플(tuple) 원소 접근

- 튜플은 아이템의 수정이 불가능하기 때문에 아이템의 삽입, 삭제, 정렬 등의 기능은 없고 조회 기능만 주로 사용함
- 리스트와 마찬가지로 인덱스를 이용하여 아이템에 접근함
- 튜플 간의 결합연산인 +와 반복연산자 \*도 사용할 수 있다

```
fruit = ('사과', '포도', '수박')
fruit[2]
'수박'
fruit[3] = '딸기'
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    fruit[3] = '딸기'
TypeError: 'tuple' object does not support item assignment
```

# 튜플을 이용한 데이터 교환



- a,b = 1,2
- 두 변수의 값을 서로 교환하여 바꾸는 일
- A, B = B, A

### 튜플의 함수



• 튜플 내 아이템조회– index(x) 함수

• 아이템이 있는지 확인: in

#### 실습 – 과목코드 조회



• 튜플에서 사용자가 과목명을 입력하면 과목에 해당하는 인덱스를 출력해보자

slist = ("국어","영어", "수학", "사회", "과학")
code = ("K001", "E002", "M001", "S003", "S004")

- 1. 검색하려는 과목을 입력받는다.
- 2. 검색하려는 이름이 과목리스트에 있다면
  - ① 과목리스트에서 과목의 인덱스를 구한다.
  - ② 화면에 과목명과 코드를 출력한다.
- 3. 없다면
  - ① 과목이 없습니다. 라고 출력한다.

검색하려는 과목을 입력하세요 : 영어 과목 : 영어 , 과목코드 E002

검색하려는 과목을 입력하세요 : 미술 과목이 없습니다.