우리는 여태까지 저장하고 싶은 새로운 정보가 있을 때마다, 변수를 새로 생성해 주었습니다. 예를 들어 저장하고 싶은 정보가 6개 있다면, 6개의 변수를 생성하여 지정해주었었습니다. 하지만 리스트(list)라는 자료형을 쓰면 하나의 변수만으로도 여러 개의 정보를 저장할 수 있습니다.

리스트를 만드는 방법은 다음과 같습니다:

```
[값1, 값2, 값3, ...]
```

에를 들어 **numbers** 라는 변수에 **1** 부터 **6** 까지의 수를 갖고 있는 리스트를 넣고 싶다면 다음과 같이 작성하면 됩니다.

```
numbers = [1, 2, 3, 4, 5, 6]
```

숫자형이 아니라, 문자열 데이터를 넣고 싶다면 어떻게 해야할까요? **names** 라는 변수에 "윤수", "혜린", "태호", "영훈" 을 갖고 있는 리스트를 넣어보겠습니다.

```
names = ["윤수", "혜린", "태호", "영훈"]
```

인덱싱 (Indexing)

지금까지는 리스트를 만드는 방법을 봤습니다. 이제 리스트에 있는 데이터를 불러오는 방법을 봅시다.

```
리스트[인덱스]
```

감이 잘 안오시죠? **numbers** 리스트로 예를 들어보겠습니다.

```
numbers = [1, 2, 3, 4, 5, 6]
print(numbers[1])
```

```
2
```

numbers[1] 을 출력하라 했는데, 숫자 **1** 이 아니라 **2** 가 나왔네요? 어찌된 일일까요?

리스트의 인덱스(index)는 **0** 부터 세기 때문입니다. **numbers[0]** 는 리스트의 첫 번째 원소인 **1** 에 해당하고, **numbers[1]** 은 리스트의 두 번째 원소인 **2** 에 해당합니다. 아래의 예시를 통해 리스트의 개념을 더 자세히 살펴봅시다.

```
# 리스트 정의
numbers = [1, 2, 3, 4, 5, 6]
names = ["윤수", "혜린", "태호", "영훈"]

# 리스트 출력하기
print(numbers)
print(names)

# 인덱싱을 통해 원소 출력하기
print(numbers[0])
print(numbers[1])
print(numbers[0] + numbers[1])
print(numbers[5])
print(numbers[6])
```

```
[1, 2, 3, 4, 5, 6]
['윤수', '혜린', '태호', '영훈']
1
2
3
6
Traceback (most recent call last):
  File "numbers_and_names.py", line 13, in <module>
    print(numbers[6])
IndexError: list index out of range
```

1. **numbers** 라는 리스트를 출력하니 **[1, 2, 3, 4, 5, 6]** 이 나왔습니다.
2. **names** 라는 리스트를 출력하니 **['윤수', '혜린', '태호', '영훈']** 이 나왔습니다.
3. 모든 리스트의 인덱스는 **0** 부터 셉니다. 따라서 **numbers[0]** 은 첫 번째 원소인 **1** 에 해당합니다.
4. **numbers[1]** 은 두 번째 원소이므로 **2** 가 출력됩니다.
5. **numbers[0] + numbers[1]** 은 **1 + 2** 의 결과값 **3** 이 출력됩니다.
6. **numbers[5]** 는 여섯 번째 원소, 즉 마지막 원소입니다. 따라서 **6** 이 출력됩니다.
7. **numbers[6]** 는 일곱번째 원소를 가리키는데, 해당 리스트에는 총 여섯 개의 원소밖에 없습니다. 따라서 실행하면 **list index out of range** 라는 오류 메시지가 나옵니다.

인덱싱(Indexing) 심화

여태까지는 **0** 과 양수 인덱스만 보셨죠? 리스트의 인덱스는 양수일 수도 있지만, 음수를 쓸 수도 있습니다.

```
numbers = [1, 2, 3, 4, 5, 6]
names = ["윤수", "혜린", "태호", "영훈"]

print(names[-1])
print(names[-2])
print(names[-4])
print(numbers[-1])
```

```
print(numbers[-2])
print(numbers[-6])
```

```
영훈
태호
윤수
6
5
1
```

names 의 길이는 **4** 이기 때문에 리스트에서 쓸 수 있는 인덱스의 범위는 **0** 에서 **3** 까지, **-1** 에서 **-4** 까지입니다. 마찬가지로 **numbers** 의 길이는 **6** 이기 때문에 리스트에서 쓸 수 있는 인덱스의 범위는 **0** 에서 **5** 까지, **-1** 에서 **-6** 까지입니다.

0 또는 자연수의 인덱스에 대해서는 앞서 배웠습니다. 그렇다면 음수의 인덱스를 쓸 수 있다는 것은 무슨 의미일까요?

1. **names[-1]** 은 마지막에서 첫번째 원소, 즉 가장 마지막 원소를 뜻합니다. 따라서 **"영훈"** 이 출력됩니다.
2. **names[-2]** 는 마지막에서 두번째 원소이므로, **"태호"** 가 출력됩니다.
3. **names[-4]** 은 마지막에서 네번째 원소, 즉 가장 첫번째 원소에 해당합니다. 따라서 **"윤수"** 가 출력됩니다.
4. 같은 방식으로 **numbers[-1]** 은 가장 마지막 원소입니다. 따라서 **6** 이 출력됩니다.
5. **numbers[-2]** 는 마지막에서 두번째 원소이므로, **5** 가 출력됩니다.
6. **numbers[-6]** 은 첫번째 원소에 해당하므로, **1** 이 출력됩니다.

슬라이싱 (Slicing)

```
numbers = [1, 2, 3, 4, 5, 6]

# 슬라이싱을 통해 원소 출력하기
print(numbers[0:4])
print(numbers[0:-3])
print(numbers[2:])
print(numbers[:3])
```

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

numbers[a:b] 는 인덱스 **a** 부터 인덱스 **b - 1** 까지의 원소를 읽어옵니다. **numbers[a:]** 는 인덱스 **a** 부터 마지막 인덱스까지의 원소를 읽어옵니다. **numbers[:b]** 는 첫번째 인덱스(**0**)부터 인덱스 **b - 1** 까지의 원소를 읽어옵니다.

1. **numbers[0:4]** 은 인덱스 **0** 부터 **3** 까지 총 4개의 원소이며, 이는 **[1, 2, 3, 4]** 에 해당됩니다.
2. **numbers[0:-3]** 은 인덱스 **0** 부터 **-4** 까지, 즉 인덱스 **0** 부터 **2** 까지 총 3개의 원소이며, 이는 **[1, 2, 3]** 에 해당됩니다.
3. **numbers[2:]** 는 인덱스 **2** 부터 마지막 인덱스까지 총 4개의 원소이며, 이는 **[3, 4, 5, 6]** 에 해당됩니다

4. **numbers[:3]** 은 인덱스 **0** 부터 **2** 까지 총 3개의 원소이며, 이는 **[1, 2, 3]** 에 해당됩니다.

원소 바꾸기

```
numbers = [1, 2, 3, 4, 5, 6]
names = ["윤수", "혜린", "태호", "영훈"]

numbers[0] = 7
print(numbers)

numbers[1] = numbers[1] + numbers[2]
print(numbers)


[7, 2, 3, 4, 5, 6]
[7, 5, 3, 4, 5, 6]
```

1. **numbers[0] = 7** 에 의해 리스트의 첫 번째 원소가 **7** 로 바뀝니다. 따라서 **[7, 2, 3, 4, 5, 6]** 이 출력됩니다.
2. **numbers[1] = numbers[1] + numbers[2]** 에 의해 리스트의 두번째 원소가 리스트의 두번째 원소 **2** 와 세번째 원소 **3** 의 합으로 바뀝니다. 즉 **numbers[1]** 이 **5** 가 됩니다.

☑ 수업을 완료하셨으면 체크해주세요.

수강생 Q&A 보기 (/questions?assignment_id=91&sort_by=popular)

질문하기 (/questions/new?assignment_id=91&op1=%ED%94%84%EB%A1%9C%EA%B7%B

이전 강의 리스트 (/assignments/90)

다음 강의 리스트 인덱싱 연습 (/assignments/92)