

# AI Programming

Lecture 8

# Assignment 6 Solution

```
import random # random 모듈 사용 선언

tries = 0 # 시도 횟수
guess = 0 # 사용자의 추측값 (초기화)
answer = random.randint(1, 50) # 1과 50사이의 난수 발생

while guess != answer: # 사용자의 예측이 틀리면 무한루프 실행
    guess = int(input("숫자를 입력하시오: "))
    tries = tries + 1 # 시도 1회 증가

    if guess < answer:
        print("너무 작음")
    elif guess > answer:
        print("너무 큼")

print(f"시도횟수: {tries}")
```

# Spyder IDE

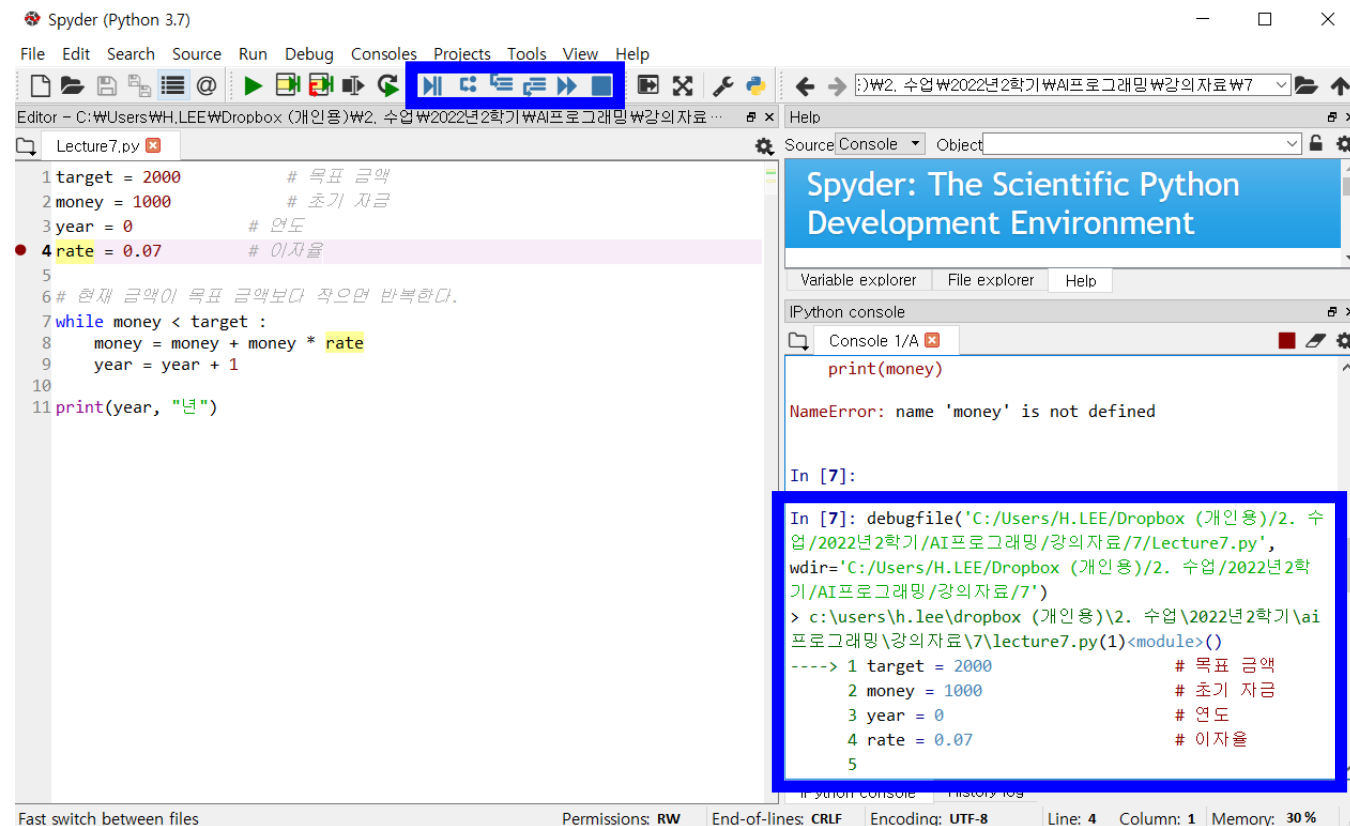
- **Download**

- Link: <https://www.spyder-ide.org/>

# Spyder IDE

- Debugging mode

Break point



중간 실행결과 확인

## 7.2 리스트의 기본

7.2.1 리스트의 개념

7.2.2 리스트의 필요성

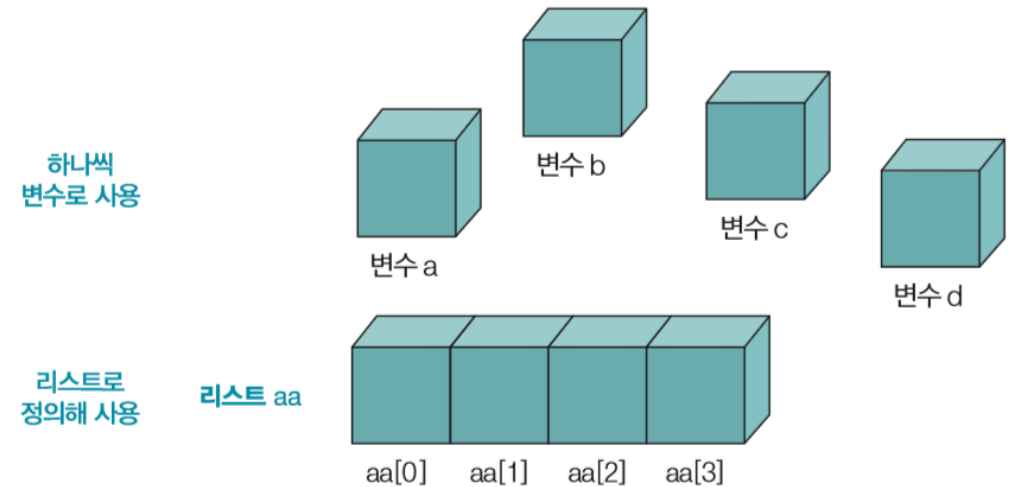
7.2.3 리스트의 일반적인 사용

7.2.4 리스트의 생성과 초기화

# Basics

- **Array and List**

- Array: C, C++, Java
  - Elements need to have the identical data type
- List: Python
  - No restrictions on data types



# Basics

- Initialization

```
❶ aa = []  
❷ bb = [10, 20, 30]  
❸ cc = ['파이썬', '공부하는', '꿀잼']  
❹ dd = [10, 20, '파이썬']
```

# Basics

- **List operations**

- Length: `len(myList)`
- Concatenation: `[1, 2] + [3, 4] = [1, 2, 3, 4]`
- Repetition: `[1]*4 = [1, 1, 1, 1]`
- Membership: `3 in [1, 2, 3] → True or False`
- Iteration: `for i in [1, 2, 3]:`



# Basics

- List operations

```
>>> aa = [10, 20, 30]
>>> bb = [40, 50, 60]
>>> len(aa)
3
>>> aa + bb
[10, 20, 30, 40, 50, 60]
>>> aa * 3
[10, 20, 30, 10, 20, 30, 10, 20, 30]
>>> 30 in aa
True
>>> 40 in aa
False
```

# Basics

- `append()` **method**

```
aa = []  
aa.append(0)  
aa.append(0)  
aa.append(0)  
aa.append(0)  
print(aa)
```

**출력 결과**

`[0, 0, 0, 0]`

```
aa = []  
for i in range(0, 100) :  
    aa.append(0)  
len(aa)
```

**출력 결과**

`100`

# Basics

## • Practice

```
1 aa = []
2 for i in range(0, 4) :
3     aa.append(0)
4 hap = 0
5
6 for i in range(0, 4) :
7     aa[i] = int(input(str(i + 1) + "번째 숫자 : " ))
8
9 hap = aa[0] + aa[1] + aa[2] + aa[3]
10
11 print("합계 ==> %d" % hap)
```

→ aa = [0]\*4

→ for i in range(4):  
hap += aa[i]

### 출력 결과

1번째 숫자 : 10

2번째 숫자 : 20

3번째 숫자 : 30

4번째 숫자 : 40

합계 ==> 100

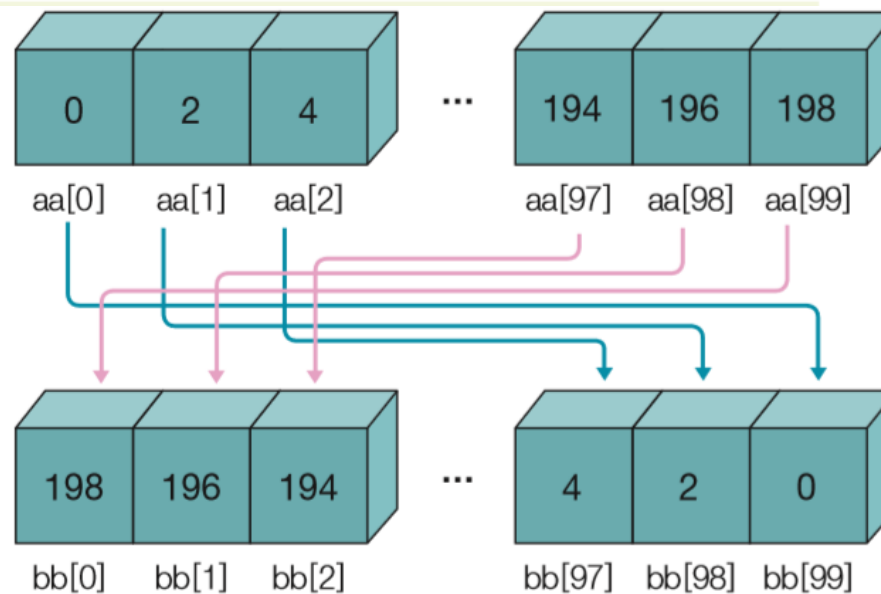
# Basics

## • Practice

```
1 aa = []
2 bb = []
3 value = 0
4
5 for i in range(0, 100) :
6     aa.append(value)
7     value += 2
8
9 for i in range(0, 100) :
10     bb.append(aa[99 - i])
11
12 print("bb[0]에는 %d이, bb[99]에는 %d이 입력됩니다." % (bb[0], bb[99]))
```

❶ 리스트 aa를  
짝수로 초기화

❷ 리스트 bb에  
역순으로 대입



## 7.2 리스트의 기본

### 7.2.5 리스트 값에 접근하는 다양한 방법

# Indexing & Slicing

- Indexing

- `myList[index]`

- `index = 0, 1, 2, ... , len(myList)-1`

```
>>> aa = [10, 20, 30, 40]
```

```
>>> aa[0]
```

```
10
```

```
>>> aa[1]
```

```
20
```

```
>>> aa[2]
```

```
30
```

```
>>> aa[3]
```

```
40
```

```
>>> aa[4]
```

```
Traceback (most recent call last):
```

```
  File "<pyshe11#12>", line 1, in <module>
```

```
    aa[4]
```

```
IndexError: list index out of range
```

# Indexing & Slicing

- Indexing

- `myList[index]`

- `index = -1, -2, ... , -len(myList)`

```
>>> aa = [10, 20, 30, 40]
```

```
>>> aa[-1]
```

```
40
```

```
>>> aa[-2]
```

```
30
```

```
>>> aa[-3]
```

```
20
```

```
>>> aa[-4]
```

```
10
```

```
>>> aa[-5]
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#47>", line 1, in <module>
```

```
    aa[-5]
```

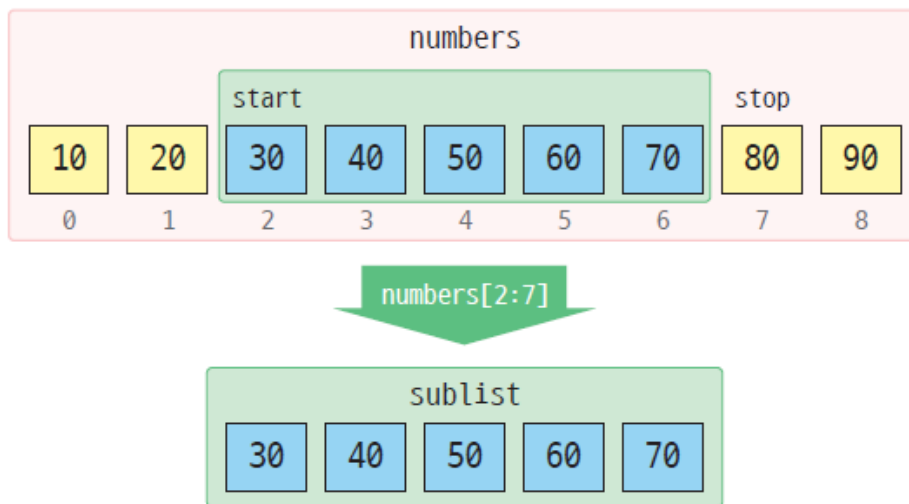
```
IndexError: list index out of range
```

# Indexing & Slicing

Syntax: 슬라이싱 #1

**형식** 리스트[ start : stop ]

**예** numbers = [ 10, 20, 30, 40, 50, 60, 70, 80, 90 ]  
sublist = numbers[2:7]





# Indexing & Slicing

- **Slicing**

- `myList[start:end+1]`
  - Default value of `start`: 0
  - Default value of `end+1`: `len(myList)`

```
>>> aa = [10, 20, 30, 40]
>>> aa[0:3]
[10, 20, 30]
>>> aa[2:4]
[30, 40]
```

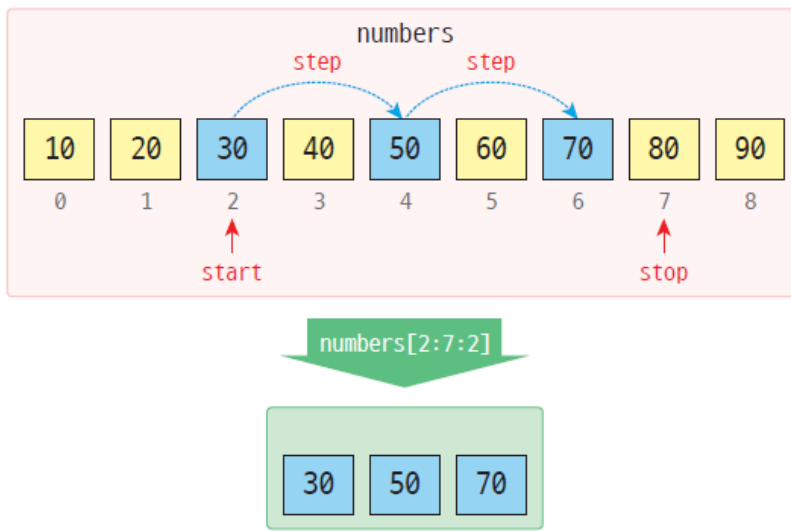
```
>>> aa[2:]
[30, 40]
>>> aa[:2]
[10, 20]
>>> aa[:]
[10, 20, 30, 40]
```

# Indexing & Slicing

Syntax: 슬라이싱 #2

**형식** 리스트[ start : stop : step ]

**예** numbers = [ 10, 20, 30, 40, 50, 60, 70, 80, 90 ]  
sublist = numbers[2:7:2]



# Indexing & Slicing

- **Slicing**

- `myList[index]`: not a list
- `myList[start:end+1]`: list

```
>>> aa = [10, 20, 30]
>>> aa[1]
20
>>> type(aa[1])
<class 'int'>
>>> bb = ['Python', 'List', 'Example']
>>> bb[1]
'List'
>>> type(bb[1])
<class 'str'>
```

```
>>> aa[1:2]
[20]
>>> type(aa[1:2])
<class 'list'>
>>> bb[1:2]
['List']
>>> type(bb[1:2])
<class 'list'>
```

# Indexing & Slicing

- **Slicing**

- `myList[start::step]`

- Default value of `start`: 0

- Default value of `step`: 1

```
>>> aa=[10, 20, 30, 40, 50, 60, 70, 80]
```

```
>>> aa[::2]
```

```
[10, 30, 50, 70]
```

```
>>> aa[::3]
```

```
[10, 40, 70]
```

```
>>> aa[2::2]
```

```
[30, 50, 70]
```

```
>>> aa[1::2]
```

```
[20, 40, 60, 80]
```

```
>>> aa[2::]
```

```
[30, 40, 50, 60, 70, 80]
```

```
>>> aa[5::]
```

```
[60, 70, 80]
```

# Indexing & Slicing

- **Slicing**

- `myList[start::-step]`
  - Default value of `start`: `len(myList)-1`

```
>>> aa=[10, 20, 30, 40, 50, 60, 70, 80]
```

```
>>> aa[::-2]
```

```
[80, 60, 40, 20]
```

```
>>> aa[::-3]
```

```
[80, 50, 20]
```

```
>>> aa[7::-2]
```

```
[80, 60, 40, 20]
```

```
>>> aa[5::-2]
```

```
[60, 40, 20]
```

```
>>> aa[::-1]
```

```
[80, 70, 60, 50, 40, 30, 20, 10]
```

## 7.2 리스트의 기본

### 7.2.6 리스트 값의 변경

# Replace & Delete

- **Replace (substitution)**

- `myList[index] = newValue`
- `myList[start:end+1] = newList`

```
>>> aa = [10, 20, 30]
>>> aa[1] = 200
>>> aa
[10, 200, 30]
>>> aa[2] = 201
>>> aa
[10, 200, 201]
```

```
>>> aa = [10, 20, 30]
>>> aa[1:3] = [200, 201]
>>> aa
[10, 200, 201]
>>> aa[:] = [10, 20, 30]
>>> aa
[10, 20, 30]
```

# Replace & Delete

- **Replace**

- `myList[start:end+1] = newList`

```
>>> aa = [10, 20, 30]
>>> aa[1:2] = [200, 201]
>>> aa
[10, 200, 201, 30]
>>> len(aa)
4
>>> aa[1]
200
>>> aa[2]
201
```

```
>>> aa = [10, 20, 30]
>>> aa[1:3] = 0
Traceback (most recent call last):
  File "<pyshell#167>", line 1, in <module>
    aa[1:3] = 0
TypeError: can only assign an iterable
>>> aa[1:3] = [0]
>>> aa
[10, 0]
```

```
>>> aa = [10, 20, 30]
>>> aa[1] = [200, 201]
>>> aa
[10, [200, 201], 30]
>>> aa[1]
[200, 201]
>>> aa[1][0]
200
>>> aa[1][1]
201
```



# Replace & Delete

- **Delete**

- `del(myList[index])`
- `del(myList[start:end+1])`

```
>>> aa = [10, 20, 30]
>>> del(aa[0])
>>> aa
[20, 30]
>>> del(aa[0])
>>> aa
[30]
```

```
>>> aa = [10, 20, 30]
>>> del(aa[0:2])
>>> aa
[30]
>>> type(aa)
<class 'list'>
```

# Replace & Delete

- **Delete**

- `myList[start:end+1] = []`

- `[]`: empty list

```
>>> aa = [10, 20, 30, 40, 50]
>>> aa[1:4] = []
>>> aa
[10, 50]
>>> aa = [10, 20, 30, 40, 50]
>>> aa[2:4] = []
>>> aa
[10, 20, 50]
```

```
>>> aa = [10, 20, 30, 40, 50]
>>> aa[0] = []
>>> aa
[[], 20, 30, 40, 50]
>>> aa = [10, 20, 30, 40, 50]
>>> aa[0:1] = []
>>> aa
[20, 30, 40, 50]
```

# Replace & Delete

- `del()`, `[]`, `None`

```
❶ aa = [10, 20, 30]; aa = []; aa  
❷ aa = [10, 20, 30]; aa = None; aa  
❸ aa = [10, 20, 30]; del(aa); aa
```

## 출력 결과

`[]`

아무것도 안 나옴

오류 발생

# Summary

- **List operations**

- Length: `len(myList)`
- Concatenation: `[1, 2] + [3, 4] = [1, 2, 3, 4]`
- Repetition: `[1]*4 = [1, 1, 1, 1]`
- Membership: `3 in [1, 2, 3] = True`
- Iteration: `for i in [1, 2, 3]:`

# Summary

- **Indexing & slicing**

- `myList[index]`
  - `myList[0], myList[1], ...`
  - `myList[-1], myList[-2], ...`
- `myList[start:end+1]`

# Assignment 7

## • 막대그래프 그리기

- Step 1) 사용자에게 항목의 수를 입력받음
- Step 2) 각 항목의 이름 및 막대 길이를 입력받음
  - 이름과 막대의 길이를 리스트로 관리
  - for, append()
- Step 3) 각 항목의 이름 출력
  - for, indexing, print(f"...", end="")
- Step 4) 각 원소의 값만큼의 \* 출력
  - for, print("\*", end="")

항목의 수를 입력하시오: 5

0번째 항목의 이름을 입력하시오: 딸기

0번째 항목의 막대 길이를 입력하시오: 3

1번째 항목의 이름을 입력하시오: 포도

1번째 항목의 막대 길이를 입력하시오: 5

2번째 항목의 이름을 입력하시오: 참외

2번째 항목의 막대 길이를 입력하시오: 2

3번째 항목의 이름을 입력하시오: 사과

3번째 항목의 막대 길이를 입력하시오: 7

4번째 항목의 이름을 입력하시오: 메론

4번째 항목의 막대 길이를 입력하시오: 9

딸기: \*\*\* 3개

포도: \*\*\*\*\* 5개

참외: \*\* 2개

사과: \*\*\*\*\* 7개

메론: \*\*\*\*\* 9개