

문제해결을 위한 코딩 첫걸음

5장 리스트_튜플_딕셔너리에 대해 알아보시다(1)

한성대학교 노은희 교수

서울세계도서관장보좌와 도서관방문기관으로 선정



학습 목표

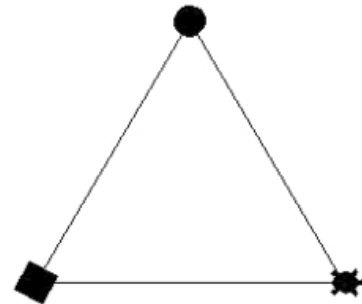
- 리스트를 이해하고 리스트의 인덱싱, 리스트 슬라이싱, 리스트 연산에 이해합니다.
- 리스트관련 메서드 함수에 대해 이해합니다.
- 튜플을 이해하고, 튜플 요소를 생성, 접근하는 방법을 이해합니다.
- 딕셔너리를 이해하고 딕셔너리 요소를 생성, 삭제, 접근하는 방법을 이해합니다.



이번 장에서 만들 프로그램

리스트 이용, 터틀 커서 모양 표시 프로그램

리스트, 터틀 그래픽 모듈 이용



딕셔너리 활용, 동물 한영사전 프로그램

딕셔너리 이용

----- 동물 한영 사전입니다. -----

호랑이, 사자, 코끼리, 토끼, 거북이 중 하나를 입력하세요:사자

사자 영어 단어는

==> Lion 입니다



5.1 리스트(list)

- 리스트
 - 여러 개의 데이터를 하나로 묶어서 처리할 수 있도록 해주는 자료형
 - 리스트 만들기
 - 비어있는 리스트 만들 경우 []만 지정
 - 항목(item)들을 쉼표(,)로 분리하여 대괄호([])안에 넣기

형식

```
리스트명 = [ ]
```

```
리스트명 = [요소1, 요소2, 요소3, ... ]
```

```
a = [ ]
```

빈 리스트

```
b = [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

숫자 가능

```
c = ["red", "orange", "yellow", "green", "blue"]
```

문자열 가능

```
d = [1, "고양이", 2, "강아지", 3, "병아리"]
```

숫자, 문자열 혼용 가능

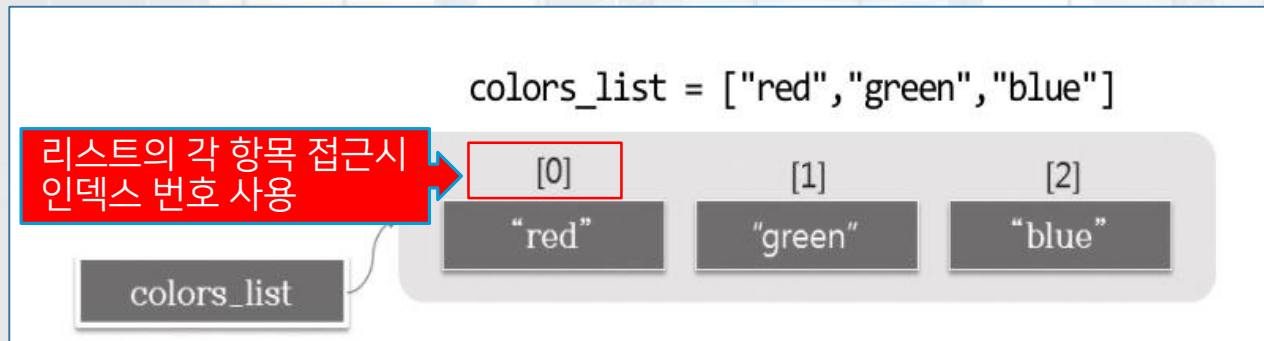
```
e = [1,3,5,7,9,[2,4,6,8,10]]
```

리스트 안에 리스트 포함 가능



리스트 인덱싱

- 리스트 인덱싱과 슬라이싱
 - 리스트 인덱싱
 - 인덱스(index) : 리스트 항목에 붙어있는 번호, 0부터 번호 시작
 - "red","green","blue" 문자열 데이터를 갖는 colors_list 생성하기



[소스코드] 5-1.py

```
colors_list = ["red", "green", "blue"]  
print(colors_list)  
print("----- 가장 좋아하는 색상은? -----")  
print(colors_list[0])
```

[실행결과]

```
['red', 'green', 'blue']  
### 가장 좋아하는 색상은? ###  
red
```



따라 해보기 : 리스트 활용

- 리스트에 저장된 데이터를 이용하여 커서 모양 찍기

[소스코드] 5-2.py

```
import turtle
```

```
t=turtle.Turtle()
```

```
s=["turtle","circle","square"]
```

← 리스트

```
t.fd(200)
```

```
t.shape(s[0])
```

← 리스트의 0번 데이터 "turtle"

```
t.stamp()
```

```
t.left(120)
```

```
t.fd(200)
```

```
t.shape(s[1])
```

← 리스트의 1번 데이터 "circle"

```
t.stamp()
```

```
t.left(120)
```

```
t.fd(200)
```

```
t.shape(s[2])
```

← 리스트의 2번 데이터 "square"

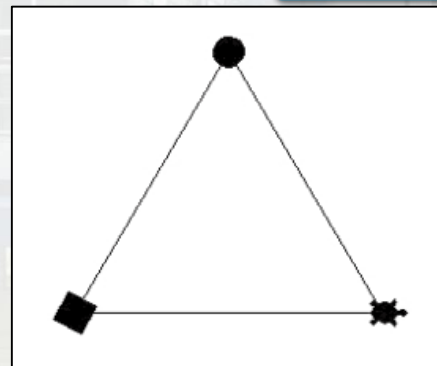
```
t.stamp()
```

```
t.ht()
```

P.59 터틀모양 지정 t.shape()

종류	형태	예시 명령
classic(기본형태)		turtle.shape("classic")
arrow		turtle.shape("arrow")
turtle		turtle.shape("turtle")
circle		turtle.shape("circle")
square		turtle.shape("square")
triangle		turtle.shape("triangle")

[실행 결과]





[실행결과]

5-2.py - C:\Users\Hansu\Documents\python\source\chapter05\5-2.py (3.8.2)

File Edit Format Run Options Window Help

```
import turtle
t=turtle.Turtle()

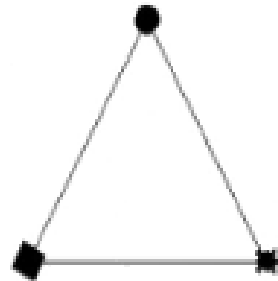
s=["turtle","circle",

t.fd(200)
t.shape(s[0])
t.stamp()      # 거북이
t.left(120)    # 왼쪽으로

t.fd(200)
t.shape(s[1])
t.stamp()      # 원모양
t.left(120)    # 왼쪽으로

t.fd(200)
t.shape(s[2])
t.stamp()      # 사각형!
t.ht()         # 숨기
```

Python Turtle Graphics



Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

>>>

```
- RESTART: C:\Users\노은희\Docume
py
>>> . 'blue'
>>> 색상은?
```

```
Users\노은희\Docume
py
>>> . 'blue'
>>> 색상은?
```

```
Users\노은희\Docume
py
>>> . 'blue'
>>> 색상은?
```

```
Users\노은희\Docume
ce\chapter05\5-2.p
```



00:01.35



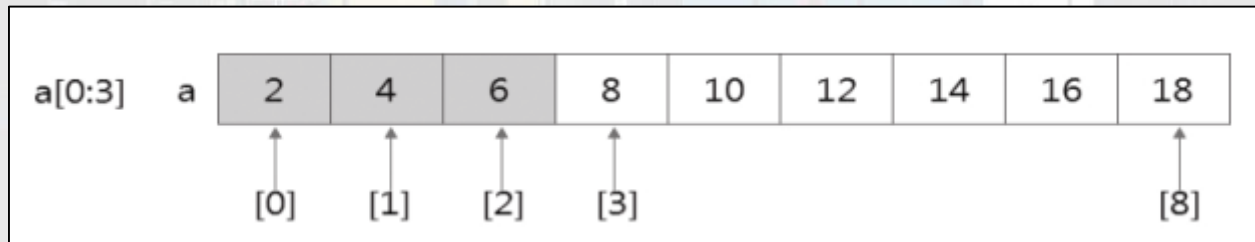


리스트 슬라이싱

- 리스트 슬라이싱
 - []안에 인덱스로 범위를 지정하여 리스트의 일부를 잘라줌

형식 리스트[시작 인덱스:끝인덱스]

끝인덱스는 가져오려는 내용보다 1 크게 지정



[소스코드] 5-3.py

```
a = [2,4,6,8,10,12,14,16,18]
print(a)
print(a[0:3])
```

[실행결과]

```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
[2, 4, 6]
```

가져오려는 끝 인덱스보다 1 크게 지정



따라 해보기 : 리스트 슬라이싱

- 리스트 슬라이싱
 - []안에 인덱스로 범위를 지정하여 리스트의 일부를 잘라줌

```
>>> a = [3,6,9,12,15,18,21,27]
```

```
>>> a[:3] ← 0부터 2까지
```

```
[3, 6, 9]
```

```
>>> a[3:] ← 3부터 끝까지
```

```
[12, 15, 18, 27]
```

```
>>> a[:] ← 0부터 끝까지
```

```
[3, 6, 9, 12, 15, 18, 27]
```



리스트(list) 연산자

- 리스트 연산자
 - + : 리스트 결합
 - * : 리스트 반복

[소스코드] 5-4.py

```
a = [1,3,5,7,9]
```

```
b = [2,4,6,8,10]
```

```
print(a + b) ← 리스트 결합
```

```
print(a * 2) ← 리스트 반복
```

[실행결과]

```
[1, 3, 5, 7, 9, 2, 4, 6, 8, 10]  
[1, 3, 5, 7, 9, 1, 3, 5, 7, 9]
```

- 리스트 요소의 개수 구하기 : len(list)

```
>>> a = [1,3,5,7,9]
```

```
>>> len(a) ← a리스트 요소의 개수
```

```
5
```

```
>>> len([1,3,5,7,9]) ← [] 안의 요소의 개수
```

```
5
```



리스트 요소 값 수정

- 리스트 요소 값 수정하기

```
>>> a = [3,6,9,13,15,18,21,27]
```

```
>>> a[3] = 12
```

a리스트 3번 인덱스 값을 수정

```
>>> a
```

```
[3,6,9,12,15,18,21,27]
```

12로 수정됨



마무리

이번 차시 수업에서는 리스트를 이해하고 리스트의 인덱싱,
리스트 슬라이싱, 리스트 연산에 대해 알아보았습니다.

