

이터러블(iterable) 객체

반복 가능한 객체

iterate : 반복하다.

- 객체 내부에 요소가 여러 개 저장되어 있고 **한 번에 하나씩 꺼내서 사용할 수 있는 객체.**
 - 문자열, 리스트, 튜플, 딕셔너리, 셋.
- 객체 내부를 확인하는 dir()함수를 사용했을 때 `"__iter__"`메소드(method)가 있는 객체.
 - iterable 객체를 반환하는 함수: range, enumerate, reversed, filter, map, zip



`dir()` 함수 : 인수로 지정한 객체가 어떤 변수와 메소드를 가지고 있는지 표시한다.

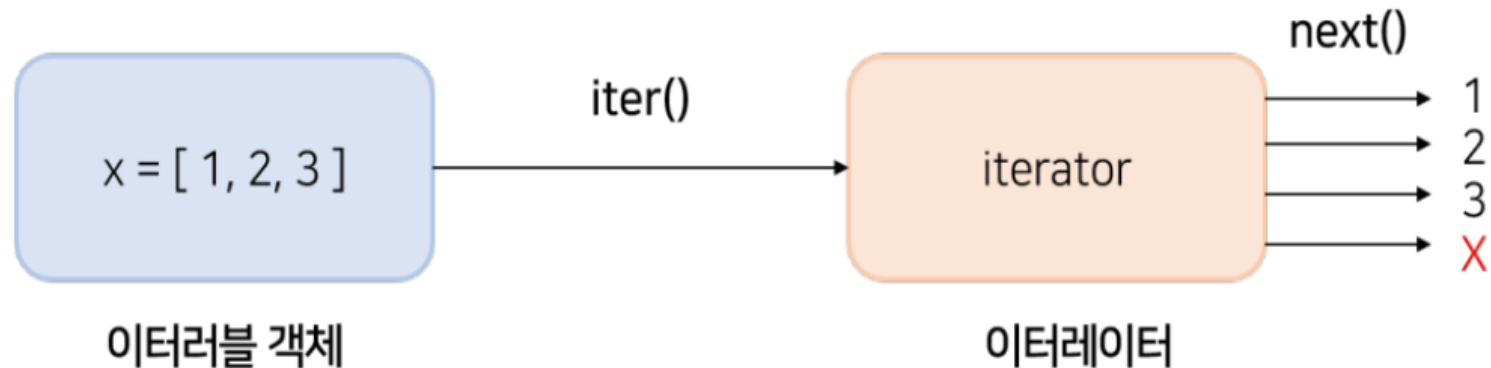
이터레이터(iterator) 객체

반복자

iterate : 반복하다.

이터레이터(iterator)란?

- 반복자라는 뜻으로 값을 차례대로 꺼내는 일을 하는 객체.
- 객체 내부를 확인하는 `dir()` 함수를 사용했을 때 `"__next__"` 메소드가 있는 객체.
- 이터레이터는 `__iter__()` 메소드로 이터레이터를 생성하고 `__next__()` 메소드로 다음 요소를 꺼낼 수 있다. 더 이상 꺼낼 요소가 없으면 `StopIteration` 예외가 발생한다



이터레이터 생성하고 하나씩 꺼내기

코드

```
li=[1,2,3,4]  
print(dir(li))
```

```
# 이터레이터 생성  
li_iter=iter(li)  
print(dir(li_iter))
```

```
# 이터레이터 변수에 next()함수로 요소 꺼내기  
print(next(li_iter))  
print(next(li_iter))  
print(next(li_iter))  
print(next(li_iter))  
print(next(li_iter))
```



```
iter() == __iter__()  
next() == __next__()
```

```
li_iter=li.__iter__()  
li_iter=iter(li)
```

```
1  
2  
3  
4  
StopIteration
```