

파이썬에서 바이너리 데이터를 처리할 때는 bytes와 bytearray를 사용합니다. 바이너리는 2진수를 의미하는데, 바이너리 데이터는 컴퓨터가 처리하는 데이터 형식을 뜻합니다.

47.3.1 bytes

bytes는 1바이트 단위의 값을 연속적으로 저장하는 시퀀스 자료형입니다(보통 1바이트는 8비트로 정의하며 0~255(0x00~0xFF)까지 정수를 사용합니다).

bytes로 바이트 객체를 만드는 방법은 3가지가 있습니다.

- **bytes(**길이)****: 정해진 길이만큼 0으로 채워진 바이트 객체를 생성
- **bytes(**반복가능한객체)****: 반복 가능한 객체로 바이트 객체를 생성
- **bytes(**바이트객체)****: 바이트 객체로 바이트 객체를 생성

```
>>> bytes(10)      # 0이 10개 들어있는 바이트 객체 생성
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
>>> bytes([10, 20, 30, 40, 50])  # 리스트로 바이트 객체 생성
b'\n\x14\x1e(2'
>>> bytes(b'hello')  # 바이트 객체로 바이트 객체 생성
b'hello'
```

파이썬에서 b'hello'처럼 '(작은따옴표)나 "(큰따옴표) 앞에 b를 붙이면 바이트(bytes) 객체가 됩니다. 간단하게 'hello'와 문자열 str의 관계라고 생각하면 됩니다.

47.3.2 bytearray

bytearray도 1바이트 단위의 값을 연속적으로 저장하는 시퀀스 자료형인데, bytes와 차이점은 요소를 변경할 수 있느냐의 차이입니다. bytes는 요소를 변경할 수 없고, bytearray는 요소를 변경할 수 있습니다.

- **bytearray(**)****: 빈 바이트 배열 객체를 생성
- **bytearray(**길이)****: 정해진 길이만큼 0으로 채워진 바이트 배열 객체를 생성
- **bytearray(**반복가능한객체)****: 반복 가능한 객체로 바이트 배열 객체를 생성
- **bytearray(**바이트객체)****: 바이트 객체로 바이트 배열 객체를 생성

다음은 바이트 배열 객체의 첫 번째 요소를 변경합니다.

```
>>> x = bytearray(b'hello')
>>> x[0] = ord('a')      # ord는 문자의 ASCII 코드를 반환
>>> x
bytearray(b'aello')
```

bytearray의 요소에 값을 할당할 때는 정수(int)를 할당해야 합니다. 문자를 넣고 싶으면 ord를 사용하여 문자의 ASCII 코드(정수)를 넣어주어야 합니다.

47.3.3 바이트 자료형과 인코딩

파이썬에서 문자열(str)의 기본 인코딩은 UTF-8인데, b'hello'와 같이 문자열을 바이트 객체로 만들면 각 문자를 ASCII 코드로 저장합니다. 보통 문자열을 UTF-8이 아닌 ASCII 코드로 처리하고 싶을 때 바이트 객체를 사용합니다.

문자열(str)을 바이트 객체로 바꾸려면 encode 메서드를 사용합니다.

- **문자열.encode()**

```
>>> 'hello'.encode()      # str을 bytes로 변환
b'hello'
```

이때 인코딩을 지정해주면 해당 인코딩으로 된 바이트 객체를 만듭니다. 다음은 한글 문자열 '안녕'을 EUC-KR 인코딩과 UTF-8 인코딩으로 된 바이트 객체로 만듭니다.

- 문자열.encode('인코딩')

```
>>> '안녕'.encode('euc-kr')
b'\xbe\x88\xb3\xe7'
>>> '안녕'.encode('utf-8')
b'\xec\x95\x88\xeb\x85\x95'
```

반대로 바이트 객체를 문자열(str)로 바꾸려면 decode 메서드를 사용하면 됩니다.

- 바이트객체.decode()

```
>>> b'hello'.decode()     # bytes를 str로 변환
'hello'
```

물론 바이트 객체가 특정 인코딩으로 되어 있다면 decode에 인코딩을 지정해주면 됩니다.

- 바이트객체.decode('인코딩')

```
>>> x = '안녕'.encode('euc-kr')
>>> x.decode('euc-kr')
'안녕'
>>> y = '안녕'.encode('utf-8')
>>> y.decode('utf-8')
'안녕'
```

bytes, bytearray는 인코딩을 지정하여 객체를 생성할 수 있습니다.

- bytes(**값, encoding='인코딩')**
- bytearray(**값, encoding='인코딩')**

다음은 한글 문자열 '안녕'을 EUC-KR, CP949 인코딩으로 된 바이트 객체로 생성합니다.

```
>>> bytes('안녕', encoding='euc-kr')
b'\xbe\x88\xb3\xe7'
>>> bytearray('안녕', encoding='cp949')
bytearray(b'\xbe\x88\xb3\xe7')
```

EUC-KR은 한글 표준 인코딩입니다. 그리고 CP949는 마이크로소프트에서 만든 한글 인코딩의 한 종류이며 EUC-KR의 확장형입니다(Windows에서 사용).