Ask Django

문자열 인코딩과 유니코드

bit and byte

- 컴퓨터 데이터의 크기를 나타내는 단위
- 1 bit : 0과 1, 2가지 데이터를 표현 가능
- 1 byte : 8bit, 256(2^8)가지 데이터를 표현 가능

Character Encoding (문자 인코딩)

- 문자나 기호들의 집합을 부호화 (인코딩) 하는 방법 (위키피디아)
- 인코딩의 2가지 의미
  - 변환하는 방법 : ascii, cp949, utf8, utf16, utf32 등
  - 변환하는 행위
- 하나의 동영상을 avi, mp4, mkv 등으로 변환(인코딩)할 수 있듯이, 하나의 문자열도 다양한 인코딩으로 변환할 수 있습니다.
- 각 인코딩마다 표현가능한 글자와 범위가 다릅니다.

Encoding & Decoding

- 일반적인 인코딩 의미 : 어떠한 값을 특정 룰에 맞춰 다른 형식으로 변환
  - 디코딩 : 역변환
- 파이썬 유니코드 문자열(str)에서의 인코딩 의미
  - 하나의 문자를 하나의 숫자로서 표현하는 다양한 Mapping Rule (인코딩따라 Rule이 다름)
  - 해당 Mapping Rule에 맞춰 변환하는 것

- 파이썬 바이트(bytes) 에서의 디코딩 의미
  - 해당 바이트를 유니코드로 디코딩할 수 있음을 알고 있다.
  - 해당 바이트가 인코딩된 인코딩에 따라, 디코딩을 수행하여 유니코드 문자열(str)을 획득

다양한 인코딩

- ascii : 7비트를 사용한 인코딩
- utf8, utf16, utf32 : **유니코드**를 따르는 인코딩 방식
- code page 949 : Microsoft의 한국어 문자 인코딩 테이블
- code page 932 : Microsoft의 일본어 문자 인코딩 테이블

Unicode (유니코드)

- 기존 인코딩의 한계를 극복하고, 전 세계의 모든 문자를 일관되게 표현할 수 있도록 설계된 산업표준
- UTF-8 (위키피디아)
  - 모든 유니코드 문자 표현 가능
  - 가변 길이 문자 인코딩 방식 : 1바이트 ~ 4바이트

출처 : Unicode 이해의 다양한 단계들

1. 최종 사용자. ㅅㅂ 한글 깨지네.
2. 습관적으로 ... charset=cp949 따위의 기존 레거시 코드를 ... 복사함.
3. UTF-8을 쓰니 Unicode 완비되었다고 생각하는 사람.
4. 세상에는 여러 종류의 인코딩이 존재하고 있다는 것을 아는 사람.
5. 특정 문자셋을 사용하는 문자(열)을 바이트열로 인코딩하는 방식이 ...
6. Unicode에 여러 평면(plane)이나 카테고리(category), ...
7. Unicode 전문가. 각종 Unicode 정규화 형식에 대해 잘 알고 있고, ...

5. 특정 문자셋을 사용하는 문자(열)을 바이트열로 인코딩하는 방식이 인코딩이며,

UTF-8이 곧 Unicode가 아니라는 것을 아는 사람. Python에서 unicode 타입과 str 타입이 왜 함께 있는지 이해하며 잘 사용한다. (생략)

- 부연설명
  - Python 2 : unicode타입(유니코드) 과 str타입(특정 인코딩)
  - Python 3 : str타입(유니코드) 과 bytes타입(특정 인코딩)

ascii code

- 7비트를 사용한 인코딩

```
>>> for i in range(128):
        print(i, repr(chr(i)))
0 '\x00'
1 '\x01'
... 중략
122 'z'
123 '{'
124 '|'
125 '}'
126 '~'
127 '\x7f'
```

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

파이썬에서의 인코딩/디코딩

- 유니코드 문자열 (str타입) > 인코딩 > bytes 타입 문자열
- bytes 타입 문자열 > 디코딩 > 유니코드 문자열 (str타입)

```
unicode_string = '가'  # str type
utf8_string = unicode_string.encode('utf-8')  # bytes type : b'\xea\xb0\x80'
cp949_string = unicode_string.encode('cp949') # bytes type : b'\xb0\xa1'
unicode_string = cp949_string.decode('cp949') # str type   : '가'
```

- 참고 : 바이너리 데이터를 파이썬으로 읽어들이면, bytes 타입
  - PSD, PNG, JPG, XLSX, 세이브 데이터 등

엉뚱한 인코딩으로 디코딩하면 UnicodeDecodeError

인코딩한 인코딩으로 디코딩하지 않고, 다른 인코딩으로 디코딩을 하면

```
>>> unicode_string = '가'  # str type
>>> utf8_string = unicode_string.encode('utf-8')   # bytes type
>>> utf8_string.decode('cp949')
---------------------------------------------------------------------------
UnicodeDecodeError                        Traceback (most recent call last)
<ipython-input-25-60100cdf68a5> in <module>()
---- 1 utf8_string.decode('cp949')

UnicodeDecodeError: 'cp949' codec can't decode byte 0x80 in position 2: incomplete multibyte sequence
```

파이썬에서의 인코딩/디코딩 Tip

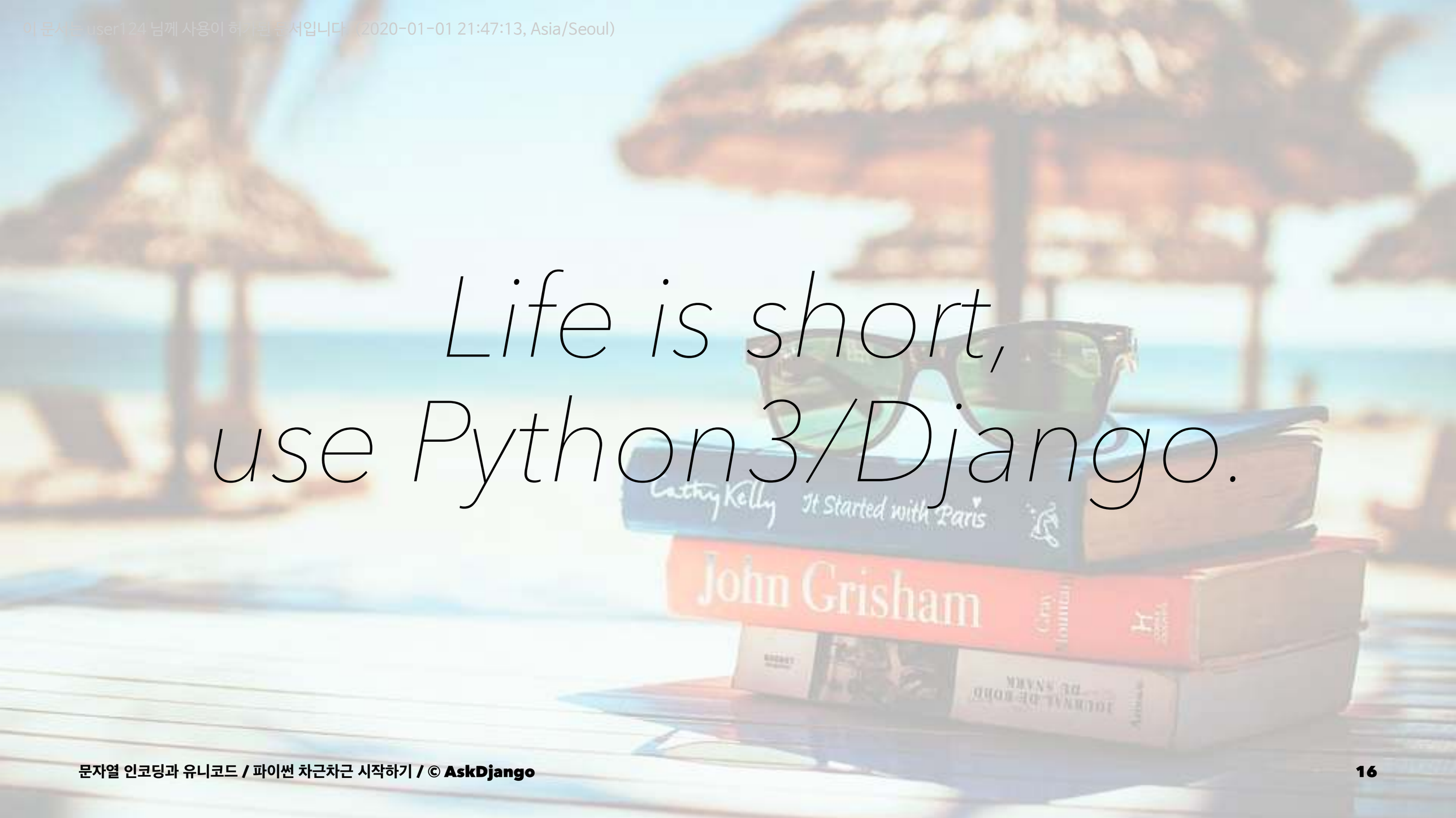
- 파이썬 코드 안에서는 모두 유니코드로 처리
  - 유니코드로 문자열을 처리하면, 한글처리에 불편함이 없습니다. 글자수 세기도 쉬움.

```
unicode_ga = '가나다'
utf8_ga = unicode_ga.encode('utf8')

print(len(unicode_ga))  # 3 : 글자 수
print(len(utf8_ga))     # 9 : 바이트 수

# 처음 2글자만 보기
print(unicode_ga[:2])
print(utf8_ga[:6])   # 인코딩따라서 참조하는 인덱스가 다름.
```

- 현재 파이썬 프로그램 밖과 문자열 데이터를 주고 받을 때에는
  - 줄 때 : 최대한 늦게 **특정 인코딩으로 인코딩**한 후에 전송
  - 받을 때 : 최한 빨리 **특정 인코딩으로 디코딩**하여, 유니코드로 처리
- 어떤 경우 ?
  - 문자열을 파일에 저장 & 읽어오기
  - 데이터베이스 통신
  - Android&iOS 앱과 통신
  - 그 외, 다수 상황



Life is short, use Python3/Django.