데이터처리프로그래밍

문자열 (string)

강원대학교 교육혁신원 송혜정

<hjsong@kangwon.ac.kr>

String

✓ 학습목표
- 문자열(string)을 이해하고 활용한다.

✓ 학습내용
- 문자열(string)
- 문자코드
- 문자열 encoding
- 문자열 인덱싱
- 문자열 연산
- 문자열 formatting
- 문자열 함수

강의에 앞서서..

- 본 강의자료는 아래의 자료들을 참고하여 만들어 졌음을 알립니다

1. 데이터과학을 위한 파이썬 프로그래밍, 최성철, 한빛아카데미,2019
2. Python (https://docs.python.org)
3. 점프 투 파이썬 (https://wikidocs.net/book/1)
4. 파이썬 for Beginner, 우재남, 한빛아카데미

String

- 문자열(string)
  - 큰따옴표(“”)나 작은따옴표(‘’)안에 묶여있는 문자들

    'hello' , "hello", 'I have a dream', '123'

```
s1='python'
s2="python"
n1 = '1234'
print(s1,s2)
print(n1)
```

```
python python
1234
```

```
f1, f2, f3 = "Apple", "Orange", "Banana"  #여러변수에 값 할당
print(f1,f2,f3)
f1 = f2 = f3 = "Apple"                    # 동일한 값 할당
print(f1,f2,f3)
```

```
Apple Orange Banana
Apple Apple Apple
```

  - 삼중 따옴표 (" " ")를 사용하여 여러 줄 문자열 할당

```
s6 = """i
have
a
dream"""
print (s6)
```

```
i
have
a
dream
```

String

- 문자열(string)
  - 문자열에 큰따옴표(“”)나 작은따옴표(‘’) 표현
    - 문자열에 작은 따옴표가 있고 큰 따옴표가 없으면 큰 따옴표로 묶어처리
    - 문자열에 큰 따옴표가 있고 작은 따옴표가 없으면 작은 따옴표로 묶어처리

```
s3="python's string"
s4='python "string"'
print(s3,s4)
```

```
python's string python "string"
```

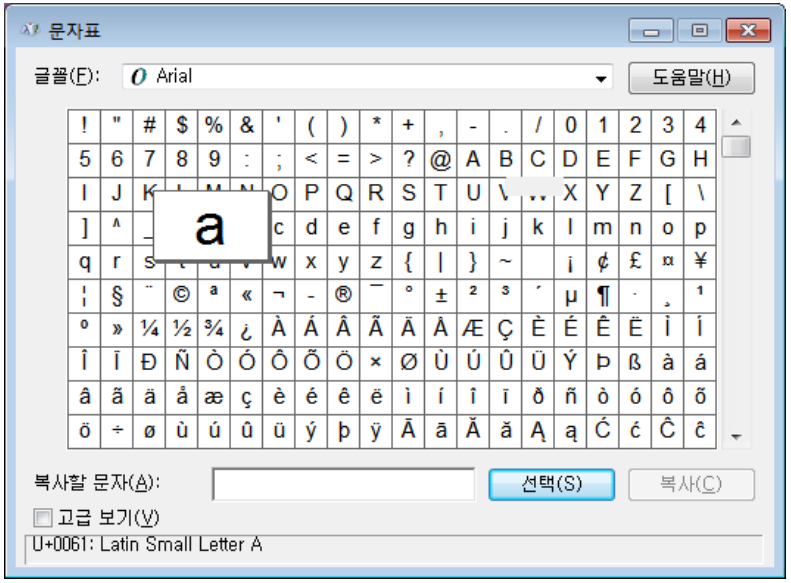
  - 제어 문자(escape character, escape sequence)
    - 백 슬래시(₩) 뒤에 제어용 문자 사용
    - ₩n : newline, ₩t : tab, ₩₩ : backslash, ₩’ : single quote

```
s5='₩₩First ₩t line.₩nSecond line. ₩t I₩'m happy !!'
print(s5)
```

```
₩First    line.
Second line.      i'm happy !!
```

문자코드

- 문자는 인코딩(encoding)하여 문자코드로 변경하여 저장
- 유니코드(Unicode)
  - 전 세계의 모든 문자를 컴퓨터에서 일관되게 표현하고 다룰 수 있도록 설계된 산업 표준
    - https://home.unicode.org/
  - UTF-8 (UTF: Unicode Transformation Format)
    - 유니코드를 위한 가변 길이 문자 인코딩 방식
    - 한 문자를 나타내기 위해 1바이트에서 4바이트까지 사용



문자코드

- 유니코드(Unicode) 예 (https://unicode-table.com)

문자코드

- 유니코드(Unicode) 사용
  - 문자와 대응되는 유니코드를 '₩u', '₩U'로 표현

    '₩uAC00 ' : 16-bit hex value (16진수 4자리)

    '₩U00002665' : 32-bit hex value (16진수 8자리)
  - 코드 변환 함수

    chr(): 코드(숫자)를 입력하여 문자로 반환

    ord(): 문자를 코드(숫자)로 반환

```
#유니코드로 문자표현
a = '₩u0041'        #16bit 16진수, 2 바이트 코드
b = '₩uAC00'
c = '₩U0001F60A'    #32bit 16진수, 4바이트 코드
d = '₩U0001F48E'
e = '₩U00002665'

print(a,b,c, d, e)

#하나의 문자 코드 확인
ch = '*'
a1 = ord(ch)  #문자를 숫자로 변환
a2 = chr(a1)  #숫자를 문자로 변환
print(a1,hex(a1), a2)  #캐드값을 1-진수, 16진수로 출력

#리스트에 담긴 여러문자를 확인
chs = ['%', 'b', '₩u0042', '₩uB098', '₩U00002168']
print('chs = ', chs)
for ch in chs:
    a1 = ord(ch)
    a2 = chr(a1)
    print(a1, hex(a1), a2)
```

```
A 가 😉 💎 ♥
42 0x2a *
chs =  ['%', 'b', 'B', '나', 'Ⅸ']
37 0x25 %
98 0x62 b
66 0x42 B
45208 0xb098 나
8552 0x2168 Ⅸ
```

문자열 encoding

- 문자열 인코딩
  - 문자열.encode(인코딩방식) : 문자열을 바이트 코드로 변환
    인코딩 방식 :
      - utf-8 : 유니코드로 인코딩
      - euc-kr : 8bit 방식 인코딩, 완성형 한글
      - ansi(ascii) : 미국표준(ascii) 코드로 인코딩
  - 문자열.decode() : 바이트 코드를 문자열로 변환

```
s = '가나다'
e1 = s.encode('utf-8')   #utf-8 방식으로 인코딩
e2 = s.encode('euc-kr')  #euc-kr 방식으로 인코딩
e3 = s.encode('ansi')    #ansi(ascii) 방식으로 인코딩
print ('s = ', s)
print ('utf-8 :', e1)
print ('euc-kr : ', e2)
print ('ansi : ', e3)

s1 = e1.decode()         #default UTF-8
s2 = e2.decode('euc-kr')
s3 = e3.decode('ansi')
print (s1)
print (s2)
print (s3)
```

```
s =  가나다
utf-8 : b'₩xea₩xb0₩x80₩xeb₩x82₩x98₩xeb₩x8b₩xa4'
euc-kr :  b'₩xb0₩xa1₩xb3₩xaa₩xb4₩xd9'
ansi :  b'₩xb0₩xa1₩xb3₩xaa₩xb4₩xd9'
가나다
가나다
가나다
```