

Chapter6 : 문자형 다루기

문자형(string)

- 시퀀스 자료형으로 문자형 data를 메모리에 저장
- 영문자 한 글 자는 1byte의 메모리공간을 사용

메모리 공간

- 컴퓨터는 2진수로 데이터를 저장, 2진수로 인식
- 1bit는 0 또는 1
- 1byte = 8bit = 2^8 = 256

프로그램 언어에서 데이터 타입

- 각 타입 별로 메모리 공간을 할당 받은 크기가 다름

종류	타입	크기	표현 범위 (32bit)
정수형	int	4바이트	$-2^{31} \sim 2^{31}-1$
	long	무제한	무제한
실수형	float	8바이트	약 $10^{-308} \sim 10^{+308}$

<http://goo.gl/JSI4ZP>

- 메모리 공간에 따라 표현할 수 있는 숫자범위가 다름
예) 4byte = 32bit = 2^{32} = 4,294M = -2,147M ~ + 2.147M 까지 표시
- 데이터 타입은 메모리의 효율적 활용을 위해 매우 중요

인덱싱(Indexing)

- 문자열의 각 문자는 개별 주소(offset)를 가짐
- 주소를 사용해 할당된 값을 가져오는 것이 인덱싱

Hello 각문자의오프셋은

0 1 2 3 4 왼쪽에선 0부터 오른쪽에선 -1부터 시작함

-5 -4 -3 -2 -1

```
>>> a = "abcde"
>>> print(a[0], a[4])      # a 변수의 0번째, 4번째 주소에 있는 값
a e
>>> print(a[-1], a[-5])      # a 변수의 오른쪽에서 0번째, 4번째 주소에 있는 값
e a
```

문자열 함수 (1/2)

함수명	기능
len(a)	문자열의 문자 개수를 반환
a.upper()	대문자로 변환
a.lower()	소문자로 변환
a.capitalize()	첫 문자를 대문자로 변환
a.title()	제목형태로 변환 띄워쓰기 후 첫 글자만 대문자
a.count('abc')	문자열 a에 'abc'가 들어간 횟수 반환
a.find('abc') a.rfind('abc')	문자열 a에 'abc'가 들어간 위치(오프셋) 반환
a.startswith('abc')	문자열 a는 'abc'로 시작하는 문자열여부 반환
a.endswith('abc')	문자열 a는 'abc'로 끝나는 문자열여부 반환

문자열 함수 (2/2)

함수명	기능
a.strip()	좌우 공백을 없앴
a.rstrip()	오른쪽 공백을 없앴
a.lstrip()	왼쪽 공백을 없앴
a.split()	공백을 기준으로 나눠 리스트로 반환
a.split('abc')	abc를 기준으로 나눠 리스트로 반환
a.isdigit()	문자열이 숫자인지 여부 반환
a.islower()	문자열이 소문자인지 여부 반환
a.isupper()	문자열이 대문자인지 여부 반환

다양한 문자열 표현

<It's OK 이라는 문자열은 어떻게 표현할까?>

a = 'ItW's ok.' - W를 사용

a = "It's ok." - 큰따옴표를 사용

<두줄 이상의 문자는 어떻게 구할까?>

""" 사용