7장

문자열 객체 소개 : 문자의 순서열

7장 문자열 객체 소개 : 문자의 순서열

7.1 문자의 순서열인 문자열

7.1. 문자의 순서열인 문자열

문자열 객체의 예

- "간단한 문자열"
- "simple“
- '또 다른 문자열'
- 'another string'
- "한글과 English와 기호(!@#@!#)가 섞인 문자열"
- "a long string with Spaces and special sym&@L5_!"
- "525600"
- "" (큰 따옴표 사이에 아무 것도 들어 있지 않으면 빈 문자열)
- '' (작은 따옴표 사이에 아무 것도 들어 있지 않으면 빈 문자열)

셀프 체크 7.1

7.2 문자열에 대한 기본 연산

7.2.1 문자열 객체 만들기

변수를 초기화하면서 문자열과 연결하면 문자열 객체를 만들 수 있다

- num_one = "one"라는 문장은 num_one 변수를 "one"이라는 값의 str 타입 객체와 연결한다.
- num_two = "2"라는 문장은 num_two라는 변수를 "2"라는 값의 str 타입 객체와 연결한다. 이때 "2"가 2라는 정수가 아니라 "2"라는 문자열임을 이해하는 것이 중요하다.

7.2.2 문자열의 인덱스 이해하기

인덱싱(indexing)

- 위치를 가지고 문자를 가리키는 것을 인덱싱(indexing)이라고 한다. 문자열에서 가장 기본적인 연산이다.

컴퓨터 과학에서는 1이 아니라 0부터 숫자를 센다. 문자열 객체를 다룰 때도 같은 원칙이 적용된다.

인덱스를 뒤에서부터 매길 수도 있다. 뒤에서부터 인덱스를 매기는 경우, 문자열의 맨 마지막 문자는 인덱스가 항상 -1이다.

- "Python rules!" 문자열을 뒤에서부터 인덱싱하는 경우, 첫 번째 문자인P의 인덱스는 -13이다.
- 문자열에서는 공백(빈칸)도 당당히 한 문자로 취급한다는 사실에 유의하자.

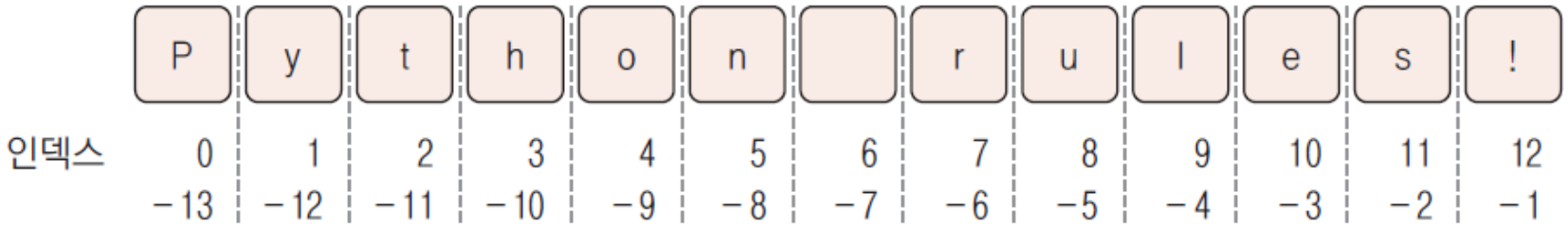

그림 7-1
"Python rules!"라는 문자열과 각 문자의 인덱스. 첫 번째 줄은 양의 정수를 사용한 인덱스를 보여 주며, 두 번째 줄은 음의 정수를 사용한 인덱스를 보여준다

7.2.2 문자열의 인덱스 이해하기

문자열에서 지정한 인덱스에 해당하는 문자를 얻는 방법 : 각괄호([ ]) 안에 원하는인덱스에 해당하는 정수 값을 넣으면 된다.

- "Python 최고!"[0]은 'P'로 계산된다.
- "Python 최고!"[7]은 '최'로 계산된다.
- "Python 최고!"[-2]는 '고'로 계산된다.

셀프 체크 7.3

7.2.3 문자열 슬라이스 이해하기

문자열의 부분 문자열(substring) 가져오기

- 예를 들어 문자열 s = "snap crackle pop"가 있을 때, "snap"은 s의 부분 문자열이다.

슬라이스(slice)

- 규칙에 따라 두 인덱스를 각괄호에 넣으면 문자열의 부분 문자열을 얻을 수 있는데, 이를 슬라이스(slice)라고 한다.
- 문자열을 슬라이스하려면 각괄호 안에 최대 세 개의 정수를 콜론(:)으로 구분해 넣으면 된다.

```
[시작_인덱스 : 끝_인덱스 : 간격 ]
```

7.2.3 문자열 슬라이스 이해하기

세 정수의 역할(그림 7-2)

- Cheer = "Python rules!" 일 때
- cheer[2:7:1]는 'thon '이다. 이 경우 왼쪽에서 오른쪽으로 차례로 한 문자씩 빠진 문자 없이 부분 문자열을 만든다. 시작 위치 인덱스는 2이고, 인덱스 7에 있는 문자('r')는 결과에 포함되지 않는다.
- cheer[2:11:3]는 'tnu'다. 왼쪽에서 오른쪽으로 처리하되 매 3번째 문자를 넣는다. 시작 인덱스는 2이고 인덱스 11에 있는 문자('s')는 결과에 포함되지 않는다.
- cheer[-2:-11:-3]는 'sun'이다. 오른쪽에서 왼쪽으로 처리하면서 매 3번째 문자를 넣기 때문이다. 시작 인덱스는 -2(오른쪽 끝에서 2번째)이며 끝 인덱스는 -11(끝에서 11번째)이지만 이 경우에도 역시 그 위치에 있는 문자(t)는 포함되지 않는다.

	P	y	t	h	o	n		r	u	l	e	s	!
인덱스	0 -13	1 -12	2 -11	3 -10	4 -9	5 -8	6 -7	7 -6	8 -5	9 -4	10 -3	11 -2	12 -1
[2:7:1]			①	②	③	④	⑤						
[2:11:3]			①			②			③				
[-2:-11:-3]						③			②			①	

7.2.3 문자열 슬라이스 이해하기

» 셀프 체크 7.4

7.3 문자열에 대한 다른 연산

7.3.1 len( )으로 문자열 안의 문자 개수 알아내기

» len() 함수를 사용하면 문자열의 전체 문자 개수를 알 수 있다. 개수에는 공백이나 기호 등도 포함된다.

- len("")는 0으로 계산된다.
- len("Boston 4 ever")는 13이다.
- len("서울의 봄")은 5로 계산된다.
- a = "어라?"인 경우 len(a)는 3이다

» 일부 명령어는 len(a)과 달리 마침표 표기 –〉 lower(a)가 아니라 a.lower()를 사용

7.3.2 upper( )와 lower( )로 대문자 소문자 변경하기

» 다양한 명령으로 대소문자를 다룰 수 있다. 문자열에서 대소문자가 있는 알파벳(꼭 영문이 아니라 러시아 키릴 문자나 그리스 문자 등에도 작용한다)만 변환한다.

- lower()는 문자열의 모든 문자를 소문자로 바꾼다. 예를 들어 "Ups AND Downs".lower()는'ups and downs'로 계산된다.
- upper()는 문자열의 모든 문자를 대문자로 바꾼다. 예를 들어 "Ups AND Downs".upper()는'UPS AND DOWNS'로 계산된다.
- swapcase()는 대문자를 소문자로, 소문자를 대문자로 바꾼다. 예를 들어 "Ups AND Downs".swapcase()는 'uPS and dOWNS'로 계산된다.
- capitalize()는 문자열의 첫 번째 문자만 대문자로 바꾸고 나머지 문자를 소문자로 바꾼다. 예를 들어 "a long Time Ago...".capitalize() 는 'A long time ago... '로 계산된다.

7.4 요약

7.4 요약

- 문자열은 문자가 하나하나 나열된 시퀀스다.
- 문자열 객체는 인용 부호(큰따옴표나 작은 따옴표)를 사용해 표현된다.
- 문자열을 처리하기 위해 여러 연산을 사용할 수 있다.

7.4 요약

» (Q7.1) 하나(또는 그 이상)의 연산을 사용해서 "Guten Morgen"에서 TEN을 계산하라. 방법은여러 가지다.

» (Q7.2) "RaceTrack"에서 Ace를 얻는 코드를 작성하라.