

제14장. 스트링 - I

목차

1. 스트링

- 스트링(string)
 - 글자들을 한 글자씩 다루는 것은 불편한 작업
 - 스트링(문자열) 객체를 사용하여 다룸
- 서브 스트링(substring)
 - 스트링의 일부분으로 인덱스를 사용하여 표현
 - 서브 스트링은 두 개의 인덱스를 사용하여 지정
 - 예) text[10:20]
 - 첫 번째 인덱스 값 : 서브 스트링의 첫번째 글자 위치
 - 두 번째 인덱스 값 : 서브 스트링의 마지막 글자 **다음** 위치
(이 위치의 글자는 포함되지 않음)

코드

1234567890

```
text = "abcdefghij"  
print(text[5:8])
```

출력

efg

2. 빈자러스

- Beans'R'US(빈자러스)웹사이트에서 실시간으로 원두커피 가격 정보를 가져옴

의 요함 | beans.itcarlow.ie/prices.html

Welcome to the Beans'R'Us Pricing Page

Current price of coffee beans = **\$5.09**

Price valid for 15 minutes from Thu Dec 5 16:13:01 2019.

3. 스타버즈 코드

- 스타버즈 코드

코드

```
import urllib.request
page = urllib.request.urlopen("http://beans.itcarlow.ie/prices.html")
text = page.read().decode("utf8")
print(text)
```

출력

```
<html> <head> <title>Welcome to the Beans'R'Us Pricing Page</title>
<link rel="stylesheet" type="text/css" href="beansrus.css" />
</head> <body>
<h2>Welcome to the Beans'R'Us Pricing Page</h2>
<p>Current price of coffee beans = <strong>$5.20</strong></p>
<p>Price valid for 15 minutes from Thu Dec 5 16:14:01 2019.</p>
</body> </html>
```

4. 웹스크래핑 라이브러리

- `import urllib.request`
 - `urllib.request` 모듈 (라이브러리) 참조
- `page = urllib.request.urlopen("http://beans.itcarlow.ie/prices.html")`
 - `urllib.request` 모듈의 `urlopen` 함수
 - 지정된 url(uniform resource locator, 웹사이트 주소)의 웹 페이지를 가져와서 `page` (`HTTPResponse` 객체)에 저장
 - 웹 페이지는 HTML 문서
 - HTML(HyperText Markup Language)
- `text = page.read().decode("utf8")`
 - `utf8`로 코드화된(`encode`) 문서를 해독(`decode`)하여 유니코드(Unicode)로 `text`에 저장

5. HTML 코드

- HTML 코드 속에 원두 커피 가격 정보가 들어 있음

```
<html> <head> <title>Welcome to the Beans'R'Us Pricing  
Page</title>
```

```
<link rel="stylesheet" type="text/css" href="beansrus.css" />
```

```
</head> <body>
```

```
<h2>Welcome to the Beans'R'Us Pricing Page</h2>
```

```
<p>Current price of coffee beans = <strong>$5.20</strong> </p>
```

```
<p>Price valid for 15 minutes from Thu Dec 5 16:14:01 2019.</p>
```

```
</body> </html>
```

6. 가격정보 추출

- HTML 전체 텍스트 에서 가격 정보는 5 글자(\$6.38)의 인덱스 [233:238] 가 필요함

코드

```
import urllib.request
page = urllib.request.urlopen("http://beans.itcarlow.ie/prices.html")
text = page.read().decode("utf8")
price = text[233:238]
print(price)
```

출력

\$6.38

7. 빈자러스 우수 고객 사이트

- Beans'R'US는 우수 고객에게 다음 사이트에 할인된 원두 커피 가격 정보를 제공

의 요약 | beans.itcarlow.ie/prices-loyalty.html



Welcome to the Beans'R'Us Pricing Page

Special Offer!!! Current price of coffee beans = **\$4.19**

Limited time only - get 'em while they're roasting!

Price valid for 15 minutes from Thu Dec 5 16:37:01 2019.

8. 빈자러스 우수 고객 사이트

- 스타버즈도 빈자러스 우수 고객이 되어 우수 고객 사이트에 접속하도록 코드를 수정

코드

```
import urllib.request
page = urllib.request.urlopen("http://beans.itcarlow.ie/prices-loyalty.html")
text = page.read().decode("utf8")
price = text[232:238]
print(price)
```

출력

e bean

9. 가격 정보 위치 변경

- 우수 고객에 대한 웹 페이지는 동적이어서 가격 정보의 위치가 수시로 변경됨
 - 이전 페이지처럼 고정된 인덱스 232:238 은 의미가 없어짐
 - 변하지 않는 위치 정보를 찾아야 함(>\$)

코드

```
import urllib.request
page = urllib.request.urlopen("http://beans.itcarlow.ie/prices-loyalty.html")
text = page.read().decode("utf8")
print(text)
```

출력

```
<html> <head> <title>Welcome to the Beans'R'Us Pricing Page</title>
<link rel="stylesheet" type="text/css" href="beansrus.css" />
</head> <body>
<h2>Welcome to the Beans'R'Us Pricing Page</h2>
<p>Special Offer!!! Current price of coffee beans =
<strong>>$4.72</strong></p>
<p>Limited time only - get 'em while they're roasting!</p>
<p>Price valid for 15 minutes from Thu Dec 5 16:43:01 2019.</p>
</body> </html>
```

10. 파이썬 스트링 라이브러리

- 파이썬의 스트링 객체를 처리하는 메서드

메서드	기능
<code>text.startswith("<HTML>")</code>	스트링 맨 앞에 서브스트링과 일치 한다면 True 반환
<code>text.endswith(".jpg")</code>	스트링 맨 뒤에 서브스트링과 일치 한다면 True 반환
<code>text.upper()</code>	대문자로 변환된 스트링 복사본을 반환
<code>text.lower()</code>	소문자로 변환된 스트링 복사본을 반환
<code>text.replace("hate","love")</code>	스트링에서 모든 특정 서브 스트링("hate")를 서브스트링("love")로 대체한 복사본을 반환
<code>text.strip()</code>	스트링의 앞뒤 공백을 제거한 스트링 복사본을 반환
<code>text.find("Python")</code>	주어진 서브스트링이 첫 번째 발견된 위치의 인덱스 값을 반환

11. 우수 고객 사이트 변경

- find 를 사용하여 동적 인덱스(>\$)를 찾아 원두 가격 정보를 추출

코드

```
import urllib.request
page = urllib.request.urlopen(" http://beans.itcarlow.ie/prices-loyalty.html ")
text = page.read().decode("utf8")
where = text.find(">$")
start_of_price = where + 1
end_of_price = start_of_price + 5
price = text[start_of_price: end_of_price]
print(price)
```

출력

\$4.57

12. 원하는 가격 알림

- 스타버즈 CEO는 커피 원두 가격이 \$4.74 이하로 내려가는 때를 알고 싶어 함
 - 가격이 내려 갈 때까지 빈자러스 사이트를 감시
 - 원하는 가격이 될 때 반복을 멈춤

코드

```
import urllib.request
price = 99.99
while price > 4.74:
    page = urllib.request.urlopen(" http://beans.itcarlow.ie/prices-loyalty.html ")
    text = page.read().decode("utf8")
    where = text.find(">$")
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    price = float(text[start_of_price: end_of_price])
print("$"+str(price))
print("Buy!")
```

출력

```
$4.21
Buy!
```

13. 빈자러스 서버 과부하

- 이런 루프를 반복하면 Beans'R'US 서버에 과부하를 유발
 - 전 세계 스타버즈 매장에서 같은 요청을 전송
 - 빈자러스 서버에 과부하 유발
 - 시간 간격을 두고 천천히 전송하는 방법 필요
- 해결 방법
 - 시간 라이브러리를 사용하여 일정 시간마다 요청

14. 파이썬 시간 라이브러리

- 파이썬의 스트링 객체를 처리하는 메서드

메서드	기능
time.clock()	호출된 이 후 부동소수점 수로 표현된 초 단위의 경과 시간
time.daylight()	현재 일광 절약 시간(daylight savings time, 서머타임)이 적용되어 있지 않으면 0을 반환
time.gmtime()	현재의 UTC(Universal Time Coordinated, 협정 세계시) 날짜와 시간을 알려줌
time.localtime()	현재의 지역 시간을 알려줌
time.sleep(secs)	지정한 시간(초) 동안 아무 일도 하지 않음
time.time()	1970년 1월 1일 이후 지나온 시간을 초 단위로 알려줌
time.timezone()	UTC 표준 시간대(런던)와 지역 표준 시간대 간의 시차를 시간단위로 알려줌

15. 웹 요청 지연

- sleep(secs)함수를 사용하여 지연

코드

```
import urllib.request
import time
price = 99.99
while price > 4.74:
    time.sleep(900)
    page = urllib.request.urlopen(" http://beans.itcarlow.ie/prices-loyalty.html ")
    text = page.read().decode("utf8")
    where = text.find(">$")
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    price = float(text[start_of_price: end_of_price])
print("$"+str(price))
print("Buy!")
```

출력

\$4.21
Buy!