

## 3장

R e g u l a r E x p r e s s i o n s

## 문자 집합으로 찾기

이 장에서는 문자 집합을 어떻게 사용하는지 알아본다. 2장에서는 어떤 문자와도 일치하는 마침표(.) 문자를 배웠는데, 이와 달리 문자 집합을 사용하면 특정 문자들과 문자 범위를 일치시킬 수 있다.

### 여러 문자 중 하나와 일치시키기

2장에서 배웠듯이, 마침표(.)는 어떤 문자든지 문자 하나와 일치한다. 2장 마지막 예제에서 na와 sa를 찾고자 .a 패턴을 사용했다. 하지만 만약 캐나다 판매 정보를 알려 주는 파일 가운데 이름이 ca1.xls인 것도 있는데, 여전히 na와 sa만을 찾고 싶을 때는 어떻게 할 것인가? ca1.xls라는 파일명도 패턴에 해당하기 때문에 마침표(.)는 c와도 일치할 것이다.

모든 문자가 아니라 다만 n이나 s와 일치하는 파일명만 찾고 싶을 때 정규 표현식에서는 메타 문자인 대괄호([])를 사용해 문자 집합을 표현한다. 대괄호([])는 문자 집합을 정의한다. 대괄호([]) 안에 있는 문자는 모두 집합의 구성원이 되며, 집합에 속한 문자 가운데 하나가 일치한다. 집합에 속한 문자가 모두 일치할 필요는 없다.

2장 예제에 나왔던 정규 표현식을 수정해 보았다.

예문
sales1.xls orders3.xls sales2.xls sales3.xls apac1.xls europe2.xls na1.xls na2.xls sa1.xls ca1.xls
정규 표현식
[ns]a.\.xls
결과
sales1.xls orders3.xls sales2.xls sales3.xls apac1.xls europe2.xls na1.xls na2.xls sa1.xls ca1.xls
분석

정규 표현식이 [ns]로 시작한다. 이 패턴은 n이나 s 중 한 문자와 일치하며, c나 다른 문자와는 일치하지 않는다. 대괄호 ([ ])는 집합을 의미하기 때문에 대괄호 자체는 어떤 문자와도 일치하지 않는다. 문자 a는 a와 일치하고 마침표(.)는 모든 문자와 일치한다. \.는 마침표(.)와 일치하고 xls는 문자 그대로 xls와 일치한다. 이 패턴을 쓰면 찾으려던 세 파일명만 찾을 수 있다.

**Note** 실제로 이런 상황에서 [ns]a.\.xls도 정답은 아니다. 만약 파일명 가운데 usa1.xls가 있다면, 해당 파일명 역시 정규 표현식과 일치하기 때문이다. 이 문제를 해결하려면 위치도 지정해서 찾아야 하는데, 이와 관련된 내용은 6장 「위치 찾기」에서 다룬다.

**Tip** 앞서 보았듯이, 정규 표현식이 제대로 동작하는지 알아보는 일은 까다로울 수 있다. 얻고자 하는 결과와 패턴이 일치하는지 확인하는 일은 매우 쉽지만, 진짜 도전 과제는 얻고 싶지 않는 결과도 일치하는지 확인하는 일이다.

문자 집합은 흔히 대소문자를 구별하지 않고 검색하거나 검색할 부분의 특정 영역만 대소문자를 구별하지 않을 때 사용한다. 다음 예를 보자.

예문
The phrase "regular expression" is often abbreviated as RegEx or regex.
정규 표현식
[Rr]eg[Ee]x
결과
The phrase "regular expression" is often abbreviated as <b>RegEx</b> or <b>regex</b> .
분석

여기서 사용한 패턴에는 문자 집합이 두 개 있다. [Rr]은 R이나 r과 일치하고, [Ee]는 E나 e와 일치한다. 이런 식으로 RegEx와 regex 둘 다 일치시킬 수 있다. 하지만 REGEX와는 일치하지 않는다.

**Tip** 대소문자를 구별하지 않고 모든 문자를 찾아도 된다면, 이런 기술은 필요 없다. 이런 식의 검색은 부분적으로 대소문자를 구별하지 않을 때만 사용한다.

### 문자 집합 범위 사용하기

다시 20쪽에서 설명한 예제로 돌아가 보자. 마지막으로 사용한 패턴 [ns]a.\.xls에는 또 다른 문제가 있다. 만약 sam.xls라는 파일이 있다면 어떻게 할 것인가? 마침표(.)는 숫자뿐만 아니라 모든 문자와 일치하기 때문에 sam.xls라는 파일명과도 일치한다.

문자 집합을 사용해 이 문제를 해결할 수 있다. 다음을 보자.

예문
sales1.xls orders3.xls sales2.xls sales3.xls apac1.xls europe2.xls sam.xls na1.xls na2.xls sa1.xls ca1.xls
정규 표현식
[ns]a[0123456789]\.xls
결과
sales1.xls orders3.xls sales2.xls sales3.xls apac1.xls europe2.xls sam.xls na1.xls na2.xls sa1.xls ca1.xls
분석

이번에 수정한 정규 표현식에서는 첫 글자가 n이나 s와 일치하고, 두 번째 글자는 a, 세 번째 글자는 숫자([0123456789]에 속한 숫자)와 일치한다. sam.xls는 일치하지 않았다는 점에 주의하자. m은 숫자 열 개 중 무엇과도 일치하지 않기 때문이다.

정규 표현식을 사용할 때, 여러분은 0부터 9, A부터 Z 같이 문자들의 범위를 지정하는 일이 자주 있음을 알게 될 것이다. 문자 범위를 단순하게 만들 때 정규 표현식에서는 특별한 메타 문자인 하이픈(-)을 제공한다.

다음은 위와 같은 예제이지만 패턴에 범위를 지정했다.

예문
sales1.xls orders3.xls sales2.xls sales3.xls apac1.xls europe2.xls sam.xls na1.xls na2.xls sa1.xls ca1.xls
정규 표현
[ns]a[0-9]\.xls
결과
sales1.xls orders3.xls sales2.xls sales3.xls apac1.xls europe2.xls sam.xls na1.xls na2.xls sa1.xls ca1.xls
분석

[0-9] 패턴은 [0123456789]와 같다. 따라서 결과도 앞서 본 예제와 같게 나온다.

범위를 단지 숫자에만 쓸 수 있는 것은 아니다. 다음과 같은 범위에도 모두 사용할 수 있다.

- A-Z는 A부터 Z 사이에 있는 모든 대문자와 일치한다.
- a-z는 a부터 z 사이에 있는 모든 소문자와 일치한다.
- A-F는 A부터 F 사이에 있는 대문자와 일치한다.
- A-z는 아스키(ASCII)문자 A와 아스키문자 z 사이에 있는 모든 문자와 일치한다(이 패턴은 사용하지 않도록 하자. Z와 a 사이에 속하는 아스키 문

자 중에는 여는 대괄호([)와 캐럿(^) 같은 문자도 포함되기 때문이다).

아스키 문자 가운데 아무거나 두 개를 선택해 한 범위의 시작과 마지막을 표시할 수 있지만, 실제 사용할 때는 숫자 전체를 범위로 정하기도 하고, 일부만 정하기도 한다. 마찬가지로 영문자(alphabetic)도 일부만 범위로 지정하기도 하고 전체를 지정하기도 한다.

**Tip** 범위를 지정할 때는 두 값 중 더 큰 값이 앞에 나오면 안 된다. 예를 들어 [3-1] 처럼 말이다. 이렇게 하면 동작하지 않는다. 또한 패턴 자체가 동작하지 않을 수도 있다.

**Note** 하이픈(-)은 대괄호([) 안에서만 메타 문자인 특수한 메타 문자다. 집합 밖에서 하이픈(-)은 단순히 문자 그대로 하이픈(-)과 일치한다. 그래서 집합 밖에서는 하이픈(-) 문자에 굳이 역슬래시(\)를 붙일 필요가 없다.

범위 여러 개를 집합 하나에 합칠 수 있다. 예를 들어 다음 패턴은 대소문자를 포함해서 모든 영숫자 문자와 일치하며, 숫자와 문자가 아닌 경우는 일치하지 않는다.

```
[A-Za-z0-9]
```

이 패턴을 펼치면 다음과 같다.

```
[ABCDEFGHJKLMN0PQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789]
```

보다시피 범위를 지정하면 정규 표현식이 더 깔끔해진다.

한 가지 예를 더 들 텐데, 이번에는 RGB 값(빨간색, 초록색, 파란색 정도에 따라 16진수로 색을 표시하는 방법)을 찾아볼 것이다. 웹 페이지에서 RGB 값은 #000000(검정), #FFFFFF(하양), #FF0000(빨강) 등으로 나타낸다. RGB 값은 대문자와 소문자를 가리지 않는다. 따라서 #FF00ff(자홍)도 맞는 표기다.

다음 예제를 보자.

```
예문
<BODY BGCOLOR="#336633" TEXT="#FFFFFF"
      MARGINWIDTH="0" MARGINHEIGHT="0" TOPMARGIN="0"
      LEFTMARGIN="0">
```

```
정규 표현식
#[0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f]
```

```
결과
<BODY BGCOLOR="#336633" TEXT="#FFFFFF"
      MARGINWIDTH="0" MARGINHEIGHT="0" TOPMARGIN="0"
      LEFTMARGIN="0">
```

**분석**  
여기서 사용한 패턴에서는 샵(#) 문자가 문자 그대로 일치하고 [0-9A-Fa-f] 문자 집합이 여섯 번 반복된다. 이 패턴은 샵(#)문자와 그 뒤로 이어지는 문자 집합 여섯 개가 예문과 일치하는데, 이 문자들은 숫자나 A와 F 사이의 문자이며, 대소문자는 구별하지 않는다.

### 제외하고 찾기

문자 집합은 일반적으로 찾고 싶은 문자의 목록을 정하는 데 쓰지만, 반대로 찾을 때 제외하고 싶은 문자 목록을 정할 때도 쓸 수 있다. 다른 말로, ‘여기서 지정한 목록을 제외하기’라고도 한다.

몇 개만 빼고 대부분의 문자를 포함해야 할 경우에는 패턴이 너무 길어지므로 원하는 문자를 모두 나열하기보다는 캐럿(^) 문자를 써 제외할 문자 집합을 지정한다. 다음 예제를 보자.

```
예문
sales1.xls
orders3.xls
sales2.xls
sales3.xls
apac1.xls
```

```
europe2.xls
sam.xls
na1.xls
na2.xls
sa1.xls
ca1.xls
```

정규 표현식

```
[ns]a[0-9]\.xls
```

결과

```
sales1.xls
orders3.xls
sales2.xls
sales3.xls
apac1.xls
europe2.xls
sam.xls
na1.xls
na2.xls
sa1.xls
ca1.xls
```

분석

이 예제에서는 바로 앞에 썼던 것과 정반대 패턴을 사용했다. [0-9]는 모든 숫자와 일치한다(또 오직 숫자하고만 일치한다). [^0-9]는 정해진 범위 내에 있는 숫자와 일치하지 않는다. 즉, [ns]a[0-9]\.xls 는 na1.xls, na2.xls, sa1.xls와는 일치하지 않고, sam.xls와 일치한다.

**Note** 캐럿(^) 문자는 이 문자 바로 뒤에 있는 문자나 범위뿐만 아니라 집합 안에 있는 문자나 범위를 모두 제외한다.

### 정리해 보자

메타 문자인 여는 대괄호()와 닫는 대괄호()는 문자 집합을 정의하는데, 그 집합 구성원 중에 한 문자라도 일치해야 한다(AND가 아닌 OR다). 문자 집합은 문자를 일일이 열거하거나 하이픈(-) 문자를 사용해 범위를 설정할 수 있다. 또 캐럿(^) 문자는 지정한 문자들을 제외한 어떤 것들과 일치시킨다.

# 4장

R e g u l a r E x p r e s s i o n s

## 메타 문자 사용하기

2장 「문자 하나 찾기」에서 이미 메타 문자를 살펴보았다. 4장에서는 특별한 문자나 문자 형(type)과 일치하는 메타 문자를 더 알아볼 것이다.

### 이스케이프 다시 살펴보기

메타 문자의 세계로 더 깊이 들어가기 전에 이스케이프(escape)에 대해 이해해야 한다.

메타 문자는 정규 표현식 안에서 특별한 의미가 있다. 마침표(.)는 메타 문자로, 어떤 문자든 문자 하나와 일치한다(2장에서 설명했다). 이와 유사하게, 여는 대괄호()도 집합의 시작을 나타내는 메타 문자다(3장 「문자 집합으로 찾기」에서 설명했다).

메타 문자들은 정규 표현식에서 특별한 의미가 있기 때문에 자기 자신을 문자 그대로 표현할 수 없다. 여는 대괄호()가 여는 대괄호()와 일치하지 않고, 마침표(.)가 마침표(.)와 일치하지 않는다는 뜻이다.

다음 예제를 보자. 이 정규 표현식에서는 여는 대괄호()와 닫는 대괄호()를 포함하는 자바스크립트 배열을 찾으려 한다.