

php 기본함수 및 정규표현식

작성일: 2020.03.05

작성자: 도원진

01. ^ 과 \$

- ^는 라인의 시작을 뜻함
- \$는 라인의 끝 을 뜻함

Case 1

Regular Expression:	^who
First match:	who is who
All matches:	who is who

Case 2

Regular Expression:	who\$
First match:	who is who
All matches:	who is who

심화

\A : 앞의 경계

\Z: 뒤의 경계

Source

Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.

Case 1

Regular Expression:	\A...
First match:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.
All matches:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.

Case 2

Regular Expression:	...\Z
First match:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago .
All matches:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago .

^, \$ 와 \A, \Z 의 차이는?

- multiLine 모드일 때
 - ^, \$ 는 모든 Line에 대해 여러개를 선택함.
 - \A, \Z는 모든 Line 통털어 처음과 끝만 선택함.

02. 특수문자가 타겟문자열에 있을 때

Source

\$12\$ -\ \$25\$

Case 1

Regular Expression:	^\$ (라인의 처음과 끝을 뜻함)	비고
First match:	\$12\$ -\ \$25\$	검색불가
All matches:	\$12\$ -\ \$25\$	검색불가

Case 2

\ (백슬래시)를 통해 escape를 시키면 정규표현식이 아닌 문자로써 받아드림.

Regular Expression:	<code>\\$</code>
First match:	<code>\$12\$ -\ \$25\$</code>
All matches:	<code>\$12\$ -\ \$25\$</code>

Case 3

라인 시작부분에 \$

Regular Expression:	<code>^\\$</code>
First match:	<code>\$12\$ -\ \$25\$</code>
All matches:	<code>\$12\$ -\ \$25\$</code>

Case 4

라인의 끝에 \$가 위치해야함

Regular Expression:	<code>\\$\\$</code>
First match:	<code>\$12\$ -\ \$25\$</code>
All matches:	<code>\$12\$ -\ \$25\$</code>

Case 5

`\\` 로 `\` 를 검색

Regular Expression:	<code>\\</code>
First match:	<code>\$12\$ \- \$25\$</code>
All matches:	<code>\$12\$ \- \$25\$</code>

03. '. '은 모든 Character를 대표

단, [] 안에서 쓰이는 . 은 ([.]) 그냥 문자 . 을 의미함.

Source

Regular expressions are powerful!!!

Case 1

Regular Expression:	<code>.</code>
First match:	<code>Regular expressions are powerful!!!</code>
All matches:	<code>Regular expressions are powerful!!!</code>

Case 2

Regular Expression:
First match:	Regular expressions are powerful!!!
All matches:	Regular expressions are powerful!!!

Source

O.K.

Case 1

AnyCharacter 검색

Regular Expression:	.
First match:	O.K.
All matches:	O.K.

Case 2

점 만 검색

Regular Expression:	\.
First match:	O.K.
All matches:	O.K.

Case 3

점과 점 사이 anyChracter

Regular Expression:	\. \.
First match:	O.K.
All matches:	O.K.

04. ' . '와 [범위] 는 문자 하나다

[] 안에서 쓰이는 . 은 그냥 점 . 이다.

[] 밖에서 혼자 쓰이는 . 은 모든 문자다.

Source

How do you do?

Case 1

[abc] 에서 [] 안에 있는 a나 b나 c 인 문자1개를 검색

Regular Expression:	[oyu]
First match:	How do you do?
All matches:	How do you do?

Case 2

[] 문자 하나 (d or H)

. 문자 하나(any)

Regular Expression:	[dH].
First match:	How do you do?
All matches:	How do you do?

Case 3

[] 문자하나 (o or w or y)

[] 문자하나 (y or o or w)

Regular Expression:	[owy][yow]
First match:	How do you do?
All matches:	How do you do?

[] 안에 범위를 정해 편리성을 높이자

Source

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789

Case 1

Regular Expression:	[C-K]
First match:	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789
All matches:	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789

Case 2

Regular Expression:	[CDEFGHIJK]
First match:	ABCDEF GHIJK LMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789
All matches:	ABC DEFGHIJK LMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789

Case 3

Regular Expression:	[a-d]
First match:	ABCDEF GHIJK LMNOPQRSTUVWXYZ a bcdefghijklmnopqrstuvwxyz 0123456789
All matches:	ABCDEF GHIJK LMNOPQRSTUVWXYZ abcd efghijklmnopqrstuvwxyz 0123456789

Case 4

Regular Expression:	[2-6]
First match:	ABCDEF GHIJK LMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 01 2345 6789
All matches:	ABCDEF GHIJK LMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 01 234567 89

Case 5

Regular Expression:	[C-Ka-d2-6]
First match:	ABCDEF GHIJK LMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789
All matches:	ABC DEFGHIJK LMNOPQRSTUVWXYZ abcd efghijklmnopqrstuvwxyz 01 234567 89

05. ^ 의 또 다른 의미 : [] 안에서 사용되면 ^ 은 not 의 의미를 갖는다.

Source

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789

Case 1

Regular Expression:	[^CDghi45]
First match:	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789
All matches:	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789

Case 2

Regular Expression:	[^W-Z]
First match:	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789
All matches:	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789

06. SubPattern

Source

Monday Tuesday Friday

Case 2

- [] 와 헛갈리지 말것. [] 는 대괄호안에 있는 조건과 매칭되는 문자하나 검색
- (|) 는 괄호안에 있는 문자열들과 매칭되는 문자열 검색

Regular Expression:	(on ues rida)
First match:	Monday Tuesday Friday
All matches:	Monday Tuesday Friday

Case 2

Regular Expression:	(Mon Tues Fri)day
First match:	Monday Tuesday Friday
All matches:	Monday Tuesday Friday

Case 3

- ay로 끝나는 단어
- any Character 2개 (..)

Regular Expression:	..(id esd nd)ay
First match:	Monday Tuesday Friday
All matches:	Monday Tuesday Friday

07. 수량자

Source

aabc abc bc

Case 1

* 앞에 있는 문자가 0개 이상일 때 즉, 있을 수도 있고 없을 수도 있고

Regular Expression:	a*b
First match:	aabc abc bc
All matches:	aabc abc bc

Case 2

+ 앞에 있는 문자가 1개 이상일 때 즉, 반드시 하나가 여러개 있는 경우

Regular Expression:	a+b
First match:	aabc abc bc
All matches:	aabc abc bc

Case 3

없거나(0개) or 하나(1개)인 경우

Regular Expression:	a?b
First match:	a abc abc bc
All matches:	a abc abc bc

수량자를 다른 특수기호와 쓰는 경우

Source

-@- *** -- "*" -- *** -@-

Case 1

* 는 0~多 개

Regular Expression:	.*
First match:	-@- *** -- "*" -- * -@-**
All matches:	-@- *** -- "*" -- * -@-**

Case 2

- - 는 표현식아니고 그냥 문자임.
- A는 있거나 없거나

Regular Expression:	-A*-
First match:	-@- *** -- "*" -- *** -@-
All matches:	-@- *** -- "*" -- *** -@-

Case 3

Regular Expression:	[-@]*
First match:	-@- *** -- "*" -- *** -@-
All matches:	-@- *** -- "*" -- *** -@-

Source

-@@@- * ** - - "*" -- * ** -@@@-

Case 1

Regular Expression:	*+
First match:	-@@@- * * * - - "*" -- * * * -@@@-
All matches:	-@@@- * * * - - "*" -- * * * -@@@-

Case 2

Regular Expression:	-@+-
First match:	-@@@- * * - - "'" -- * * * -@@@-
All matches:	-@@@- * * - - "'" -- * * * -@@@-

Case 3

공백이 아닌 것이 하나이상일 경우

Regular Expression:	[^]+
First match:	-@@@- * * - - "'" -- * * * -@@@-
All matches:	-@@@- * * * - - "*" -- * * * -@@@-

Source

--XX-@-XX-@@-XX-@@@-XX-@@@@-XX-@@-@@-

Case 1

?: 없거나 하나

- 반드시 있어야 하는 것: 양쪽의 대시(-)

Regular Expression:	-X?XX?X
First match:	--XX-@-XX-@@-XX-@@@-XX-@@@@-XX-@@-@@-
All matches:	--XX-@-XX-@@-XX-@@@-XX-@@@@-XX-@@-@@-

Case 2

Regular Expression:	-@?@?@?-
First match:	--XX-@-XX-@@-XX-@@@-XX-@@@@-XX-@@-@@-
All matches:	--XX-@-XX-@@-XX-@@@-XX-@@@@-XX-@@-@@-

Case 3

Regular Expression:	[^@]@?@
First match:	--XX-@-XX-@@-XX-@@@-XX-@@@@-XX-@@-@@-
All matches:	--XX-@-XX-@@-XX-@@@-XX-@@@@-XX-@@**-@@**-

08. 개선된 수량자

Source

One ring to bring them all and in the darkness bind them

Case 1

이전 수량자의 한계

- * 또는 + 는 수량을 표현하기에 너무 general하다
- { n }을 통해 반복할 횟수를 표현할 수 있다.

Regular Expression:	.{5}
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

Case 2

{n, m} : n이상 m이하의 갯수반복

Regular Expression:	[els]{1,3}
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

Case 3

{n, } : n이상 반복

Regular Expression:	[a-z]{3,}
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

{ } 수량자로 + , * , ? 모두 대체할 수 있다.

Source

AA ABA ABBA ABBBA

Case 1

Regular Expression:	AB*A
First match:	AA ABA ABBA ABBBA
All matches:	AA ABA ABBA ABBBA

Case 2

Regular Expression:	AB{0,}A
First match:	AA ABA ABBA ABBBA
All matches:	AA ABA ABBA ABBBA

Case 3

Regular Expression:	AB+A
First match:	AA ABA ABBA ABBBA
All matches:	AA ABA ABBA ABBBA

Case 4

Regular Expression:	AB{1,}A
First match:	AA ABA ABBA ABBBA
All matches:	AA ABA ABBA ABBBA

Case 5

Regular Expression:	AB?A
First match:	AA ABA ABBA ABBBA
All matches:	AA ABA ABBA ABBBA

Case 6

Regular Expression:	AB{0,1}A
First match:	AA ABA ABBA ABBBA
All matches:	AA ABA ABBA ABBBA

수량자 뒤에 오는 ?는 수량자를 최소값으로 변경시킨다.

Source

One ring to bring them all and in the darkness bind them

Case 1

r 뒤에 어떤 문자열이 오든 상관없다

Regular Expression:	r.*
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

Case 2

수량자 다음에 오는 ?가 오면 수량자의 기존 의미를 달라지게 만든다.

- 0~多 의 의미를 갖는 * 의미가
- 0 개를 갖는 것으로 의미가 변경된다.

Regular Expression:	r.*?
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

Case 3

Regular Expression:	r.+
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

Case 4

- 1~多 의 의미를 갖는 + 의미가
- 1 개를 갖는 것으로 의미가 변경된다.

Regular Expression:	r.+?
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

Case 5

Regular Expression:	r.?
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

Case 6

- 0 or 1 의 의미를 갖는 ? 의미가
- 0 개를 갖는 것으로 의미가 변경된다.

Regular Expression:	r.??
First match:	One ring to bring them all and in the darkness bind them
All matches:	One ring to bring them all and in the darkness bind them

그럼 ? 가 어디에 쓰이나

Source

<div>test</div><div>test2</div>

Case 1

- 탐욕적인 수량자
 - </div>는 중간의 것이 아니라 source 맨 마지막에 있는 </div>를 선택함.

Regular Expression:	<div>.+</div>
First match:	<div>test</div><div>test2</div>
All matches:	<div>test</div><div>test2</div>

Case 2

- 게으른 수량자
 - ?를 넣어서

Regular Expression:	<div>.+?</div>
First match:	<div>test</div><div>test2</div>
All matches:	<div>test</div><div>test2</div>

09. Character Class

알파벳(a,b,c,,,A,B,C,,)+숫자(0,1,2,,,,,)+언더라인(_) = \w

\w에 " "공백은 포함되지 않음.

Source

A1 B2 c3 d_4 e:5 ffGG77--__--

Case 1

Regular Expression:	\w
First match:	A1 B2 c3 d_4 e:5 ffGG77--__--
All matches:	A1 B2 c3 d_4 e:5 ffGG77--__--

Case 2

Regular Expression:	\w*
First match:	A1 B2 c3 d_4 e:5 ffGG77--__--
All matches:	A1 B2 c3 d_4 e:5 ffGG77--__--

Case 3

Regular Expression:	[a-z]\w*
First match:	A1 B2 c3 d_4 e:5 ffGG77--__--
All matches:	A1 B2 c3 d_4 e:5 ffGG77--__--

Case 4

Regular Expression:	\w{5}
First match:	A1 B2 c3 d_4 e:5 ffGG77 --__--
All matches:	A1 B2 c3 d_4 e:5 ffGG77 --__--

Case 5

Regular Expression:	[A-z0-9_]
First match:	A1 B2 c3 d_4 e:5 ffGG77--__--
All matches:	A**1** B**2** c**3** d**4** e:5 f**fGG77--_**--

\W 는 not \w 임.

Source

AS_34:AS11.23 @#\$ %12^*

Case 1

Regular Expression:	\W
First match:	AS_34:AS11.23 @\$ %12^*
All matches:	AS_34:AS11.23 @**#\$** %12^*

Case 2

Regular Expression:	\w
First match:	AS_34:AS11.23 @\$ %12^*
All matches:	A*S**_**34:AS11.23** @\$ %1**2**^

Case 3

Regular Expression:	[^A-z0-9_]
First match:	AS_34:AS11.23 @\$ %12^*
All matches:	AS_34:AS11.23 @**#\$** %12^*

\s는 whitespace(공백)를 뜻함

Source

Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.

Case 1

Regular Expression:	\s
First match:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.
All matches:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.

\S는 not \s를 뜻함.

Case 2

Regular Expression:	\s
First match:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.
All matches:	Ere iron was found or tree was hewn, When young was mountain under moon; Ere ring was made, or wrought was woe, It walked the forests long ago.

\d는 숫자를 뜻함

Source

Page 123; published: 1234 id=12#24@112

Case 1

Regular Expression:	\d
First match:	Page 123 ; published: 1234 id=12#24@112
All matches:	Page 123 ; published: 1234 id= 12#24@112

Case 2

Regular Expression:	\D
First match:	Page 123; published: 1234 id=12#24@112
All matches:	Page 123; published: 1234 id=12#24@112

Case 3

Regular Expression:	[0-9]
First match:	Page 123 ; published: 1234 id=12#24@112
All matches:	Page 123 ; published: 1234 id= 12#24@112

word boundary : \b

- 단어를 식별할 때
 - \b\w+\b

Case 1

Regular Expression:	\b\w
First match:	E re iron was found
All matches:	Ere i ron w as f ound

Case 2

Regular Expression:	\w\b
First match:	E r e iron was found
All matches:	E r e iron n was f ound

Case 3

Regular Expression:	\b\w+\b
First match:	Ere iron was found
All matches:	Ere iron was found

Case 4

Regular Expression:	\bcat	cat\b
All matches:	cat concat	cat concat

word boundary : \B (not \b)

경계가 아닌 쪽의 문자가 선택됨.

Case 1

Regular Expression:	\B.	.\B
First match:	cat concat	cat concat

10. "?=" 은 특수한 기호

- **?=**
 - 전방탐색
 - **Reg. Expression:**
`.(?=원)`
 - Text:**

1000원
2000원
3000원
5000원
10000원

- ?<=
 - 후방탐색
 - **Reg. Expression:**
(?<=)[0-9.]+
Text:
1: \$600.4
2: \$10.25
3: \$47.33
4: \$112.34

Source

AAAX---aaax---111

Case 1

- 앞에 문자가 1개 이상
- (?=X)
 - 탐색을 할 때는 X를 사용하지만
 - 결과적으로 X는 선택하지 않는다.

Regular Expression:	\w+(?=X)
First match:	AAAX---aaax---111
All matches:	AAAX---aaax---111

Case 2

Regular Expression:	\w+
First match:	AAAX---aaax---111
All matches:	AAAX---aaax---111

Case 3

Regular Expression:	\w+(?=\w)
First match:	AAAX---aaax---111
All matches:	AAAX---aaax---111

부정형 전방탐색 (?!)

Source

AAAX---AAA

Case 1

Regular Expression:	AAA(?!X)
First match:	AAAX--- AAA
All matches:	AAAX--- AAA

부정형 후방탐색(?<!)

Reg. Expression:

\b(?<!\\$)\d+

- 해석
 - 앞에 경계가 있고
 - \$가 앞에 없어야 하고
 - 숫자만 여러개

Text:

\$10 **5** \$6 **77** \$788