

Javascript 정규식

2014.1.17

Step 1

정규표현식은 대소문자를 구별한다.

Source

Hello, world!

Case 1

Regular Expression: **Hello**

First match: **Hello**, world!

All matches: **Hello**, world!

Case 2

Regular Expression: **hello**

First match: Hello, world!

All matches: Hello, world!

Step 2

눈에 보이지 않는 space, tab, new line
기호도 정확히 일치해야한다.

Source

Hello, world!

Case 1

Regular Expression: **Hello, world**

First match: **Hello, world!**

All matches: **Hello, world!**

Case 2

Regular Expression: **Hello, world**

First match: Hello, world!

All matches: Hello, world!

Step 3

어떤 문자는 특별한 의미를 가지고 있다.

[^] 는 매칭할 문자열의 시작을 의미한다.

^{\$} 는 매칭할 문자열의 끝을 의미한다.

Source

who is who

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**

Step 4

만일 특별한 의미를 가지고 있는 문자열의 **literal** 값이 필요하다면 역슬래시(\)를 앞에 붙여 줘야한다.

Source

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

Case 1

Regular Expression: ^\$

First match: \$12\$ \- \ \$25\$

All matches: \$12\$ \- \ \$25\$

Case 2

Regular Expression: \\$

First match: \$12\$ \- \ \$25\$

All matches: \$12\$ \- \ \$25\$

Case 3

Regular Expression: ^\\$

First match: \$12\$ \- \ \$25\$

All matches: \$12\$ \- \ \$25\$

Case 4

Regular Expression: \\$\\$

First match: \$12\$ \- \ \$25\$

All matches: \$12\$ \- \ \$25\$

Case 5

Regular Expression: \\

First match: \$12\$ \- \ \$25\$

All matches: \$12\$ \- \ \$25\$

Step 5

- 점은 모든 문자를 의미한다.

Source

Regular expressions are powerful!!!

Case 1

Regular Expression:

.

First match: Regular expressions are powerful!!!

All matches: Regular expressions are powerful!!!

Case 2

Regular Expression:

.....

First match: Regular expressions are powerful!!!

All matches: Regular expressions are powerful!!!

Step 6

. 점의 literal 값이 필요하다면
역시 역슬래시(\) 가 필요하다.

Source

O.K.

Case 1

Regular Expression: .

First match: O.K.

All matches: O.K.

Case 2

Regular Expression: \.

First match: O.K.

All matches: O.K.

Case 3

Regular Expression: \.\.

First match: O.K.

All matches: O.K.

Step 7

[] 대괄호 안에는 매칭될수 있는 문자의 목록을 넣는다.
목록의 순서는 중요하지 않다.

Case 1

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

Case 2

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

Case 3

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

Step 8

문자의 범위는 [-] 문법으로 나타낼수 있다.
여러가지 범위도 Case 5 처럼 하나의 표현식으로 쓸수있다.

Source

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

Case 1

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

Case 2

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

Case 3

Regular Expression: [a-d]
First match: ABCDEFGHIJKLMNOPQRSTUVWXYZ
All matches: ABCDEFGHIJKLMNOPQRSTUVWXYZ
matches: abcdefghijklmnopqrstuvwxyz 0123456789

Case 4

Regular Expression: [2-6]
First match: ABCDEFGHIJKLMNOPQRSTUVWXYZ
All matches: ABCDEFGHIJKLMNOPQRSTUVWXYZ
matches: abcdefghijklmnopqrstuvwxyz 0123456789

Case 5

Regular Expression: [C-Ka-d2-6]
First match: ABCDEFGHIJKLMNOPQRSTUVWXYZ
All matches: ABCDEFGHIJKLMNOPQRSTUVWXYZ
matches: abcdefghijklmnopqrstuvwxyz 0123456789

Case 9

[] => 문자 클래스

문자 클래스 안에 [^abc] 처럼 첫문자로 ^ 가 있다면
a,b,c 각각은 매칭하지 않을 문자 목록이 된다.

Source

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

Case 1

Regular Expression: [^CDghi45]

First match: ABCDEFGHIJKLMNOPQRSTUVWXYZ
All matches: abcdefghijklmnopqrstuvwxyz 0123456789

Case 2

Regular Expression: [^W-Z]

First match: ABCDEFGHIJKLMNOPQRSTUVWXYZ
All matches: abcdefghijklmnopqrstuvwxyz 0123456789

Step 10 ????

문자열을 교차 매칭 시키려면 소괄호 안에 | 로 구분해서
문자열을 나열하면된다.
(문자열1 | 문자열2 | 문자열3)

Source

Monday Tuesday Friday

Case 2

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

Regular Expression: ..(id|esd|nd)ay

First match: Monday Tuesday Friday

All matches: Monday Tuesday Friday

Step 11

수량자(Quantifiers) : ***** , **+** , **?**

수량자는 문자가 몇번 올수 있는지 정의한다.

***** : 0번이상 (없어도 되고 여러개가 있어도 된다)

+ : 1번이상 (반드시 있어야 하며 여러개가 있어도 된다)

? : 0 번 또는 1번 (없어도 되고 있다면 오직 한개만 허용)

Source

aabc abc bc

Case 1

Regular Expression: **a*b**

First match: **aabc** abc bc

All matches: **aabc** **abc** **bc**

Case 2

Regular Expression: **a+b**

First match: **aabc** abc bc

All matches: **aabc** **abc** bc

Case 3

Regular Expression: **a?b**

First match: **aabc** abc bc

All matches: **aabc** **abc** **bc**

Step 12

수량자 * 의 사용예

0 번 이상

Source

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

Case 1

Regular Expression:

.*

First match:

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

All matches:


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

Case 2



Regular Expression:

-A*-

First match:

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

All matches:


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

Case 3





Regular Expression:

[-@]*

First match:

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

All matches:

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

Step 13

수량자 **+** 의 사용예

1번이상 (없으면 안됨)

Source

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

Case 1

Regular Expression:

*+

First match: -@@@- * * * - - "*" - - * * * -@@@-

All matches: -@@@- * * * - - "*" - - * * * -@@@-

Case 2

Regular Expression:

-@+ -

First match: -@@@- * * * - - "*" - - * * * -@@@-

All matches: -@@@- * * * - - "*" - - * * * -@@@-

Case 3

Regular Expression:

[^]+

First match: -@@@- * * * - - "*" - - * * * -@@@-

All matches: -@@@- * * * - - "*" - - * * * -@@@-

Step 14

수량자 ? 사용예

0 번 또는 1번 (있어도 한번만 가능)

Source

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

Case 1

Regular

-X?XX?X

Expression:

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-@@-@@-

Step 15

{ } 는 정확한 문자의 반복횟수를 정의한다.

{ m } 는 m번 반복 (Case 1)

{ m,n } 는 최소 m 번 최대 n번 반복

{ m, } 는 최소 m번 반복

Source

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

Case 1

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

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

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**

Step 16

- * 는 { 0, } 와 같다.
- + 는 { 1, } 와 같다.
- ? 는 { 0, 1 } 와 같다.

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