

정규표현식(1)

2014년 05월 21일 이훈구



- 목 차 -

*Regular
expressions*

1. 정규표현식이란?

2. Javascript

3. 문법(3~21)

22. Example



1. 정규표현식

(1) 개요

정규표현식(Regular Expression)은 문자열을 처리하는 방법 중의 하나로 특정한 조건의 문자를 '검색'하거나 '치환'하는 과정을 매우 간편하게 처리 할 수 있도록 하는 수단

`^(?:#([\w-]+)|(\w+)|\.([\w-]+))$`

(2) 참고 Site

- 교육문서 :

<http://zvon.org/comp/r/tut-Regexp.html#Pages~Contents>

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions

- 정규표현식 테스트 URL : <http://www.regexr.com/>
- 정규표현식 도식화 URL: <http://www.regexper.com/>

2. Javascript Syntax

*Regular
expressions*

- Javascript Method

Method	Description
exec	정규식으로 문자열을 전달인자로 받아 배열로 만들어 반환, 없으면 null 로 반환
test	문자열을 정규식으로 추출시 성공하면 true 실패하면 false 반환
match	이 메서드는 정규 표현식을 유일한 전달인자로 받고 매치된 결과를 배열로 만들어 반환하고 매칭되는 부분 문자열이 없다면 null 을 반환
search	이 메서드는 정규 표현식을 전달인자로 받아서 가장 처음 매칭되는 부분 문자열의 위치를 반환 하고, 매칭되는 부분 문자열이 없다면 -1 을 반환 * 정규표현식에서 g 플래그가 있으면 무시
replace	이 메서드는 찾아서 바꾸기 작업을 수행 * 정규 표현식에 g 플래그가 설정되어 있으면 문자열내에서 패턴에 매치되는 모든 부분 문자열을 교체할 문자열로 변경 * 정규 표현식에서 괄호로 묶인 부분 표현식은 왼쪽에서 오른쪽으로 번호가 매겨지고, 각 부분 표현식과 매치된 텍스트를 기억, 만약 교체할 문자열에 \$ 가 나오고 뒤따라 숫자가 나타나면 replace() 메서드는 \$ 와 숫자를 부분 문자열에 매치된 텍스트로 변경
split	문자열을 정규식에 정의된 구분으로 쪼개서 배열로 반환

2. Javascript Syntax

*Regular
expressions*

- Javascript Method

Flag	Description
i	대소문자를 구별하지 않고 매칭
g	전역 매칭 수행. 즉, 첫 번째 매치에서 끝내지 않고 매치되는 모든 것을 찾음
m	여러 줄 상태 ^는 줄의 시작이나 문자열의 시작에 매치되고, \$는 줄의 끝이나 문자열의 끝에 매치



2. Javascript Syntax

*Regular
expressions*

- Javascript 정규표현식 사용

```
var str = "#id";
```

```
var re = /^(?:#([\w-]+)|(\w+)|\.[\w-]+)$/g;
```

```
var re = new RegExp("^(?:#[\w-]+)|(\w+)|\.[\w-]+)$", "g");
```

```
re.exec(str);
```

```
str.match(re);
```

```
re.test(str);
```

```
var str = "abc de";
```

```
str.split(/\s+/);
```



2.문자

Regular expressions

- 문자를 입력 시 선택
- 대소문자 구분함

Source

Hello, world!

Case 1

Regular expression: Hello

First match: Hello, world!

All match: Hello, world!

Case 2

Regular expression: hello

First match: Hello, world!

All match: Hello, world!



2.문자

Regular expressions

- 문자를 입력 시 선택됨
- 공백, Tab, Newline 도 구분함

Source

Hello, world!

Case 1

Regular expression: Hello, world

First match: Hello, world!

All match: Hello, world!

Case 2

Regular expression: hello, world

First match: Hello, world!

All match: Hello, world!



3. “^” & “\$”

- “^” 라인의 첫번째를 표현
- “\$” 라인의 마지막을 표현

Source

who is who

Case 1

Regular expression: ^who

First match: **who** is who

All match: **who** is who

Case 2

Regular expression: who\$

First match: who is **who**

All match: who is **who**



4. “\” (Backslash)

- “\” 는 명령어를 문자로 인식

Source

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

Case 1

Regular expression: ^\$

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

All match: \$12\$ \- \$25\$

Case 2

Regular expression: \\$

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

All match: \$12\$ \- \$25\$



4. “\” (Backslash)

Source

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

Case 3

Regular expression: `^\$`

First match: `12 \- 25`

All match: `12 \- 25`

Case 4

Regular expression: `\$`

First match: `12 \- 25`

All match: `12 \- 25`

Case 5

Regular expression: `\\`

First match: `12 \- 25`

All match: `12 \- 25`



5. “.” (Point)

- “.” 모든 문자를 표현

Source

Regular expressions are powerful!!! O.K.

Case 1

Regular expression: .

First match: Regular expressions are powerful!!! O.K.

All match: **Regular expressions are powerful!!! O.K.**

Case 2

Regular expression:

First match: Regular expressions are powerful!!! O.K.

All match: **Regular expressions are powerful!!! O.K.**



5. “.” (Point)

*Regular
expressions*

Source

Regular expressions are powerful!!! O.K.

Case 3

Regular expression: \.

First match: Regular expressions are powerful!!! O.K.

All match: Regular expressions are powerful!!! O.K.

Case 4

Regular expression: \.\.

First match: Regular expressions are powerful!!! O.K.

All match: Regular expressions are powerful!!! O.K.



6. “[]” (Square brackets) *Regular expressions*

- “[]” 대괄호 안에 있는 문자를 검색

Source

How do you do?

Case 1

Regular expression: [oyu]

First match: How do you do?

All match: How do you do?

Case 2

Regular expression: [dH].

First match: How do you do?

All match: How do you do?



6. “[]” (Square brackets) *Regular expressions*

Source

How do you do?

Case 3

Regular expression: [owy][yow]

First match: H**ow** do you do?

All match: H**ow** do **y**ou do?



7. “[-]” (Square brackets) *Regular expressions*

- “[-]” 문자를 나열하지 않고 Form To로 지정 가능

Source

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

Case 1

Regular expression: [C-K] = [CDEFGHIJK]

First match:

AB**C**DEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

All match:

AB**CDEFGHIJK**LMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

7. “[-]” (Square brackets) *Regular expressions*

Source

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

Case 2

Regular expression: [C-Ka-d2-6]

First match:

AB**C**DEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

All match:

AB**CDEFGHIJK**LMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 01**23456**789



8. “[^]”

- “[^]” 대괄호 안의 “^”는 첫번째를 표현하는 것이 아니라 부정을 표현

Source

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

Case 1

Regular expression: [^CDW-Zghi45]

First match:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

All match:

ABCD**EFGHIJKLMNOPQRSTU**VWXYZ
abcdef**ghijklmnopqrstuvw**xyz **0123**45**6789**

9. “|” (Separate)

- “(|)” 소괄호 안의 “|”는 또는이라는 의미

Source

Monday Tuesday Friday

Case 1

Regular expression: (on|ues|rida)

First match: Monday Tuesday Friday

All match: Monday Tuesday Friday

Case 2

Regular expression: (Mon|Tues|Fri)day

First match: Monday Tuesday Friday

All match: Monday Tuesday Friday



9. “|” (Separate)

Regular expressions

Source

Monday Tuesday Friday

Case 3

Regular expression: `..(id|esd|nd)ay`

First match: **Monday** Tuesday Friday

All match: **Monday Tuesday Friday**



10. “*” & “+” & “?”

Regular expressions

- “*” 앞문자의 개수를 의미 0~n개를 표현
- “+” 앞문자의 개수를 의미 1+n개를 표현
- “?” 앞문자의 개수를 의미 0+1개를 표현

Source

aabc abc bc

Case 1

Regular expression: **a*b**

First match: **aabc** abc bc

All match: **aabc abc bc**

Case 2

Regular expression: **a+b**

First match: **aabc** abc bc

All match: **aabc abc bc**



10. “*” & “+” & “?”

Regular expressions

Source

aabc abc bc

Case 3

Regular expression: a?b

First match: a**ab**c abc bc

All match: a**ab**c **ab**c **b**c



10. “*” & “+” & “?”

Regular expressions

Source

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

Case 4

Regular expression: .*

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

All match: -@- *** -- "*" -- *** -@-

Case 5

Regular expression: -A*-

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

All match: -@- *** -- "*" -- *** -@-

Case 6

Regular expression: [-@]*

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

All match: -@- *** -- "*" -- *** -@-



10. “*” & “+” & “?”

Regular expressions

Source

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

Case 7

Regular expression: *+

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

All match: -@@@- * ** - - "*" -- * ** -@@@-

Case 8

Regular expression: -@+-

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

All match: -@@@- * ** - - "*" -- * ** -@@@-

Case 9

Regular expression: [^]+

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

All match: -@@@- * ** - - "*" -- * ** -@@@-

10. “*” & “+” & “?”

Regular expressions

Source

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

Case 10

Regular expression: -X?XX?X

First match: --XX-@-XX-@@-XX-@@@@-XX-@@@@-XX-@@-@@-

All match: --XX-@-XX-@@-XX-@@@@-XX-@@@@-XX-@@-@@-

Case 11

Regular expression: -@?@?@?-

First match: --XX-@-XX-@@-XX-@@@@-XX-@@@@-XX-@@-@@-

All match: --XX-@-XX-@@-XX-@@@@-XX-@@@@-XX-@@-@@-

Case 12

Regular expression: [^@]@?@

First match: --XX-@-XX-@@-XX-@@@@-XX-@@@@-XX-@@-@@-

All match: --XX-@-XX-@@-XX-@@@@-XX-@@@@-XX-@@-@@-

11. “{}”

- “{}” 소괄호 안의 숫자는 앞문자의 개수를 표현

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 match: **One ring to bring them all and in the darkness bind them**

Case 2

Regular expression: `[els]{1,3}`

First match: One**e** ring to bring them all and in the darkness bind them

All match: One**e** ring to bring the**m** all and in the**e** darkne**ss** bind the**m**

11. “{}”

Regular expressions

Source

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 match: One **ring** to **bring them all and** in **the darkness bind them**



11. “{}”

Regular expressions

Source

AA ABA ABBA ABBBA

Case 4

Regular expression: $AB\{0,\}A == AB^*A$

First match: AA ABA ABBA ABBBA

All match: AA ABA ABBA ABBBA

Case 5

Regular expression: $AB\{1,\}A == AB^+A$

First match: AA ABA ABBA ABBBA

All match: AA ABA ABBA ABBBA

Case 6

Regular expression: $AB\{0,1\}A == AB^?A$

First match: AA ABA ABBA ABBBA

All match: AA ABA ABBA ABBBA



12. “*?” & “+?” & “??”

*Regular
expressions*

- “*?” “*”은 0~n 개라는 의미를 가지고 있는데 “?”가 뒤에 있으면 “*” 가 가지는 0~n 개중 0이라는 개수만 가지게됨
- “+?” “+”은 1~n 개라는 의미를 가지고 있는데 “?”가 뒤에 있으면 “+” 가 가지는 1~n 개중 1이라는 개수만 가지게됨
- “??” “?”은 0~1 개라는 의미를 가지고 있는데 “?”가 뒤에 있으면 “?” 가 가지는 0~1 개중 0이라는 개수만 가지게됨



12. “*?” & “+?” & “??”

*Regular
expressions*

Source

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

Case 1

Regular expression: r.*

First match: One **ring to bring them all and in the darkness bind them**

All match: One **ring to bring them all and in the darkness bind them**

Case 2

Regular expression: r.*?

First match: One **r**ing to bring them all and in the darkness bind them

All match: One **r**ing to **b**ring them all and in the da**r**kness bind them



12. “*?” & “+?” & “??”

*Regular
expressions*

Source

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 match: One **ring to bring them all and in the darkness bind them**

Case 4

Regular expression: r.+?

First match: One **ri**ng to bring them all and in the darkness bind them

All match: One **ri**ng to **br**ing them all and in the **dar**kness bind them



12. “*?” & “+?” & “??”

Regular expressions

Source

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

Case 5

Regular expression: r.?

First match: One **r**ing to bring them all and in the darkness bind them

All match: One **r**ing to **b**ring them all and in the **dar**kness bind them

Case 6

Regular expression: r.??

First match: One **r**ing to bring them all and in the darkness bind them

All match: One **r**ing to **b**ring them all and in the **dar**kness bind them



12. “*?” & “+?” & “??”

Regular expressions

Source

<div>test1</div> <div>test2</div> <div></div> <div></div>

Case 7

Regular expression: <div>.*</div>

First match: <div>test1</div> <div>test2</div>
<div></div> <div></div>

All match: <div>test1</div> <div>test2</div>
<div></div><div></div>

Case 8

Regular expression: <div>.*?</div>

First match: <div>test1</div> <div>test2</div>
<div></div><div></div>

All match: <div>test1</div> <div>test2</div>
<div></div> <div></div>

12. “*?” & “+?” & “??”

Regular expressions

Source

<div>test1</div> <div>test2</div> <div></div> <div></div>

Case 9

Regular expression: <div>.+</div>

First match: <div>test1</div> <div>test2</div>
<div></div> <div></div>

All match: <div>test1</div> <div>test2</div>
<div></div><div></div>

Case 10

Regular expression: <div>.+?</div>

First match: <div>test1</div> <div>test2</div>
<div></div><div></div>

All match: <div>test1</div> <div>test2</div>
<div></div> <div></div>

12. “*?” & “+?” & “??”

Regular expressions

Source

<div>test1</div> <div>test2</div> <div></div> <div></div>

Case 11

Regular expression: <div>.?</div>

First match: <div>test1</div> <div>test2</div>

<div></div> <div></div>

All match: <div>test1</div> <div>test2</div>

<div></div> <div></div>

Case 12

Regular expression: <div>.??</div>

First match: <div>test1</div> <div>test2</div>

<div></div> <div></div>

All match: <div>test1</div> <div>test2</div>

<div></div> <div></div>

13. “\w” (word)

*Regular
expressions*

- “\w” 는 “[A-z0-9_]” 과 동일한 표현

Source

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

Case 1

Regular expression: \w == [A-z0-9_]

First match: **A1** B2 c3 d_4 e:5 ffGG77--__--

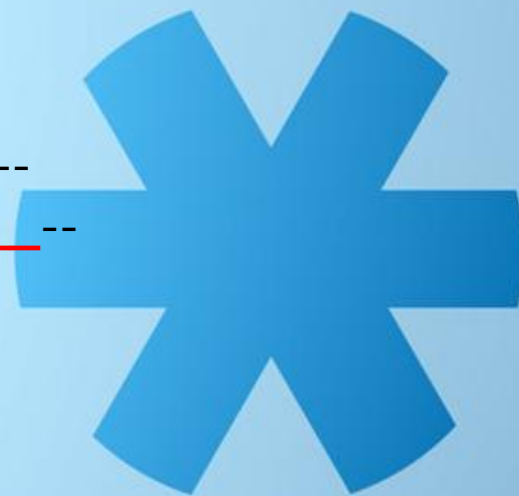
All match: **A1 B2 c3 d_4 e:5 ffGG77--__--**

Case 2

Regular expression: \w+

First match: **A1** B2 c3 d_4 e:5 ffGG77--__--

All match: **A1 B2 c3 d_4 e:5 ffGG77--__--**



13. “\w” (word)

*Regular
expressions*

Source

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

Case 4

Regular expression: \w{5}

First match: A1 B2 c3 d_4 e:5 **ffGG7**7--__--

All match: A1 B2 c3 d_4 e:5 **ffGG7**7--__--



14. “\W” (not word)

Regular expressions

- “\w” 는 “[^A-z0-9_]” 과 동일한 표현

Source

AS _34:AS11.23 @#\$ %12^*

Case 1

Regular expression: \W == [^A-z0-9_]

First match: AS _34:AS11.23 @#\$ %12^*

All match: AS _34:AS11.23 @#\$ %12^*



15. “\s” & “\S” (Space)

*Regular
expressions*

- “\s” Space 문자를 말함
- “\S” Space를 제외한 문자를 말함

Source

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

Case 1

Regular expression: \s

First match: One ring to bring them all and in the darkness bind them

All match: One ring to bring them all and in the darkness bind them

Case 2

Regular expression: \S

First match: One ring to bring them all and in the darkness bind them

All match: One ring to bring them all and in the darkness bind them

16. “\d” & “\D” (Decimal)

*Regular
expressions*

- “\d” 숫자를 말함
- “\D” 숫자를 제외한 모든문자(Space포함)를 말함

Source

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

Case 1

Regular expression: \d == [0-9]

First match: Page **1**23; published: 1234 id=12#24@112

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

Case 2

Regular expression: \D

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

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

17. “\b” (Boundary)

*Regular
expressions*

- “\b” Word Boundary는 문자의 식별을 할수 있는 기능

Source

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

Case 1

Regular expression: `\b\w` <- 단어의 첫번째 문자를 선택

First match: One ring to bring them all and in the darkness bind them

All match: One ring to bring them all and in the darkness bind them

Case 2

Regular expression: `\w\b` <- 단어의 마지막 문자를 선택

First match: One ring to bring them all and in the darkness bind them

All match: One ring to bring them all and in the darkness bind them

17. “\b” (Boundary)

*Regular
expressions*

Source

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

Case 3

Regular expression: `\b\w+` <- 단어를 선택

First match: **One** ring to bring them all and in the darkness bind them

All match: **One ring to bring them all and in the darkness bind them**



17. “\b” (Boundary)

Source

cat cats tomcat

Case 4

Regular expression: `\bcat` <- **cat**으로 시작하는 단어를 선택

First match: **cat** cats tomcat

All match: **cat cat**s tomcat

Case 5

Regular expression: `cat\b` <- **cat**으로 끝나는 단어를 선택

First match: **cat** cats tomcat

All match: **cat** cats tom**cat**



18. “\B” (Boundary)

*Regular
expressions*

- “\b” Word Boundary는 문자의 식별을 할수 있는 기능의 반대

Source

cat cats tomcat

Case 1

Regular expression: \B. <- 단어의 첫번째 문자를 제외한 선택

First match: cat cats tomcat

All match: cat cats tomcat

Case 2

Regular expression: \B.\B. <- 단어의 첫자 마지막 문자를 제외하여 선택

First match: cat cats tomcat

All match: cat cats tomcat



19. “\A” & “\Z”

*Regular
expressions*

- “\A”는 “^” 와 비슷하나 이것은 같은 행에서 처음을 가르키고
- “\A” 문장에 제일 처음을 가리키는 기능
- “\Z”는 “\$” 와 비슷하나 이것은 같은 행에서 마지막을 가르키고
- “\Z” 문장에 제일 마지막을 가리키는 기능

Source

Hello

Hello World

Case 1

Regular expression: \A...

First match:

Hello

Hello World

All match:

Hello

Hello World



19. “\A” & “\Z”

*Regular
expressions*

Source

Hello

Hello World

Case 2

Regular expression: ...`\Z`

First match:

Hello

Hello World

All match:

Hello

Hello World



20. “(?=<pattern>)”

- 검색대상에서는 존재하나 선택 시 제외시키는 기능

Source

AAAX---aaax---111

Case 1

Regular expression: `\w+(?=X)` <- 앞은 임의문자가 오고 나중에 “X”로 끝남

First match: **AAAX**---aaax---111

All match: **AAAX**---aaax---111

Case 2

Regular expression: `\w+(?=\w)` <-전부 문자이면서 끝은 제외

First match: **AAAX**---aaax---111

All match: **AAAX**---**aaa**x---**111**



21. “(?!<pattern>)”

- 단어 검색중 검색대상이 존재하게 되면 해당 건을 제외하고 선택

Source

AAAX---AAA

Case 1

Regular expression: **AAA(?!X)**

First match: AAAX---**AAA**

All match: AAAX--- **AAA**

Case 2

Regular expression: **AAA**

First match: **AAA**X---AAA

All match: **AAA** X--- **AAA**



22.Example

- URL 추출

`https?:\/\/[\w-]+\.(?:[\w-]+|)(?:(?:\.)[\w-]+(?:\.)[\w-]+|)/`

- jQuery (<http://code.jquery.com/jquery.js>)

`rinputs = /^(?:input|select|textarea|button)$/i`
input, select, textarea, button

`rheader = /^h\d$/i`
- h1, h2, h3

`rquickExpr = /^(?:#([\w-]+)|(\w+)|\.([\w-]+))$/`
- ID, TAG, CLASS 구분



Q & A

감사합니다.

Thank you

