



Automata Formal Languages & Logic

Preet Kanwal

Department of Computer Science & Engineering

Automata Formal Languages & Logic

Unit 2

Preet Kanwal

Department of Computer Science & Engineering

Regular Expression in Practice

Automata Formal Languages and Logic

Unit 2 - Regular Expression in Practice



Special characters

1. .

Special characters

1. .

(dot)

matches any single character except newline

Automata Formal Languages and Logic

Unit 2 - Regular Expression in Practice



Special characters

1. .
2. *

Special characters

1. .

2. *

(Star)

0 or more repetitions of preceding regex

Special characters

1. .
2. *

Example :

a^*

Strings matched :

0 or more no. of a's

Example : $\lambda, a, aa, aaa, aaaa, aaaaa, \dots$

Automata Formal Languages and Logic

Unit 2 - Regular Expression in Practice



Special characters

1. .
2. *
3. +

Special characters

1. .
2. *
3. +

Example :

a+

Strings matched :

1 or more no. of a's

Example : a,aa,aaa,aaaa,aaaaa....

Special characters

1. .
2. *
3. +
4. ?

Special characters

1. .
2. *
3. +
4. ?

Example:

ab?c

matches abc or ac

Special characters

1. .
2. *
3. +
4. ?
5. []

Character Class

Matches any single character

Special characters

1. .
2. *
3. +
4. ?
5. []

Example

[cat]

String matched :

c

or a

or t

Special characters

1. .
2. *
3. +
4. ?
5. []
6. ^

^ (caret) :

Example:

Example: ^abc matches "a" at the start of the string.

[^abc]: This pattern matches any character except a or b or c.

Special characters

1. .
2. *
3. +
4. ?
5. []
6. ^
7. \$

\$ (dollar) :

Example :

- > `abc$` matches "c" at the end of a line.
- > `^$` matches the empty string.

Special characters

1. .
2. *
3. +
4. ?
5. []
6. ^
7. \$
8. { } (Syntax - $R\{m,n\}$)

Example of { }

$a\{0, \}$ - same as a^*

$a\{1, \}$ - same as a^+

$a\{2,3\}$ matches "aa" or "aaa".

$a\{3\}$ matches aaa only

Special characters

1. .
2. *
3. +
4. ?
5. []
6. ^
7. \$
8. { } (Syntax - $R\{m,n\}$)
9. \d - matches any digit between [0-9]
 - a. \d{9} matches any 9 digits
 - b. \d{2,3} matches 2 or 3 digits

We will discuss how to construct regex for validating the following :

- PAN Card
- Adhaar Card
- Mobile Number
- Date
- Email Address

- **PAN CARD:**

The valid PAN Card number must satisfy the following conditions:

1. It should be 10 characters long.
2. The first five characters should be any upper case alphabets.
3. The next four-characters should be any number from 0 to 9.
4. The last(tenth) character should be any upper case alphabet.
5. It should not contain any white spaces.

PAN CARD Regex - $^[A-Z]\{5\}[0-9]\{4\}[A-Z]\$$

Input: str = "BNZAA2318J"

Output: true

Explanation:

The given string satisfies all the above mentioned conditions.

Input: str = "23ZAABN18J"

Output: false

Explanation:

The given string does not start with upper case alphabets, therefore it is not a valid PAN Card number.

- **AADHAR NUMBER:**

The valid Aadhar number must satisfy the following conditions:

1. It should have 12 digits.
2. It should not start with 0 and 1.
3. It should not contains any alphabet and special characters.
4. It should have white space after every 4 digits.

AADHAR Number Regex - `^[2-9]{1}\d{3}\s{4}[" "]\s{4}$`

Input: str = "3675 9834 6012"

Output: true

Explanation:

The given string satisfies all the above mentioned conditions. Therefore it is a valid Aadhar number.

Input: str = "3675 9834 6012 8"

Output: false

Explanation:

The given string contains 13 digits. Therefore it is not a valid Aadhar number.

- **INDIAN MOBILE NUMBER:**

The valid Mobile number must satisfy the following conditions:

- It is a 10 digits number
- The first digit should contain number between 6 to 9.
- The rest 9 digit can contain any number between 0 to 9.
- The mobile number can have 11 digits also by including 0 at the starting.
- The mobile number can be of 13 digits also by including +91 at the starting

Automata Formal Languages and Logic

Unit 2 - Regular Expression in Practice



MOBILE Number Regex - `^(0|"+91")?[6-9]\d{9}$`

Dates in the year 2020

- It is a leap year
- Let the format be DD-MM-YY

Months with 31 days are : Jan, Mar, May, July, Aug, Oct, Dec
: [1, 3, 5, 7, 8, 10, 12]

Months with 30 days are : Apr, June, Sep, Nov
: [4, 6, 9, 11]

Month with 29 days : Feb
: [2]

- Year : 2020

Dates in the year 2020

Regex:

```
(  
(0[0-9])|[10-31]"-(0[13578]|1[0|2])  
|  
(0[0-9])|[10-30]"-(0[469]|11)  
|  
(0[0-9])|[10-29]"-"02"  
)"-""2020"
```

Email Address:

An email is a string (a subset of ASCII characters) separated into two parts by @ symbol. a "personal_info" and a domain, that is personal_info@domain.

The length of the personal_info part may be up to 64 characters long and domain name may be up to 253 characters.

The personal_info part contains the following ASCII characters.

- Uppercase (A-Z) and lowercase (a-z) English letters.
- Digits (0-9).
- Characters ! # \$ % & ' * + - / = ? ^ _ ` { | } ~
- Character . (period, dot or fullstop) provided that it is not the first or last character and it will not come one after the other.

The domain name [for example com, org, net, in, us, info] part contains letters, digits, hyphens, and dots.

Automata Formal Languages and Logic

Unit 2 - Regular Expression in Practice



Email Address:

Example of valid email id

- mysite@ouearth.com
- my.ownsite@ouearth.org
- mysite@you.net

Example of invalid email id

- mysite.ouearth.com [@ is not present]
- mysite@.com.my [tld (Top Level domain) can not start with dot "."]
- @you.me.net [No character before @]

**$$^[_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)^*@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)^*\.$$
$$(((0-9)\{1,3\})|([a-zA-Z]\{2,3\})|(com|edu|info|museum|name))\$$$**

Regular Expression in practice: Examples

Example 2: Simple URL Validator

Regular Expression in practice: Examples

Example 2: Simple URL Validator

$$^((ht|f)tp(s?))\:\/\/([0-9a-zA-Z\-.]+[a-zA-Z]{2,6}(\:[0-9]+)?(\/\S*)?)\$$$

Regular Expression in practice: Examples

Example 5: -mail addresses Validation

Regular Expression in practice: Examples

Example 5: -mail addresses Validation

$$^[_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)^*@[a-zA-Z0-9-]+(\. [a-zA-Z0-9-]+)^*\.
(([0-9]{1,3}) | ([a-zA-Z]{2,3}) | (com | edu | info | museum | name))\$$$

Regular Expression in practice: Examples

Example 6: Ip Address validation:

Regular Expression in practice: Examples

Example 6: Ip Address validation:

```
^(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9])\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[0-9])$
```



THANK YOU

Preet Kanwal

Department of Computer Science & Engineering

preetkanwal@pes.edu

+91 80 6666 3333 Extn 724