

# Python Regex Language

# Single Character Patterns

# Single Character Patterns

Basic building block is a single character pattern

Problem: Find all occurrences of letter 'a'

Pattern: a

Text: this is a great test

# Regex is case-sensitive

## Regex is case sensitive

Problem: Find all occurrences of letter 'a'

Pattern: A

Text: this is a great test

## Set case-insensitive comparison flag *i*

Pattern: (?i)A

Text: this is a great test

# String Literal

You can specify a literal string as pattern

Problem: Find all occurrences of string 'big'

Pattern: big

Text: this is the biggest test

# OR Condition

Pipe Character | is used for OR condition

Problem: Find all occurrences of letters 'b','i','g'

Pattern: b|i|g

Text: this is the biggest test

# OR Condition using Set

Set [...] is used for OR condition

Problem: Find all occurrences of letters 'b','i','g'

Pattern: [big]

Text: this is the biggest test

# Set Negation

Negate a set using ^ as first character in the set

Problem: Find all characters that are NOT vowels

Pattern: [^aeiou]

Flags: global (g)

Text: this ^ is a big test

For negation, ^ needs to be first character in the set; otherwise, it is literal ^

Pattern: [a^eiou]

Text: this is a \_ big test



# Range of characters

## Set notation supports range of characters

Problem: Find all occurrences of (a,b,c,d)

Pattern: [a-d]

Text: this is a big test

## Multiple Range of Characters

Problem: Find all occurrences of (a,b,c,d,x,y,z,0,1,2,3)

Pattern: [a-dx-z0-3]

Text: x-ray 3 won't work for this test

# Negate Range of Characters

Negate using ^

Pattern: `[^a-dx-z0-3]`

Text: x-ray 3 won't work for this test

# Wildcard

Dot (.) is a wildcard character and matches all characters except newline \n

Problem: Match all characters

Pattern: . (dot)

Text: This is a big test

Dot (.) can match newline when s flag is set

Pattern: (?s). (dot)

Text: this is a great test

Line 2

# Escape character

## Use \ to escape special characters

Problem: Find all occurrences of '.' (dot)

Pattern: \.

Text: This\_ Is\_ a test\_

## Use Set

Pattern: [.]

Text: This\_ Is\_ a test\_

# Control Characters

Control characters (tab, newline, carriage return and so forth)

Problem: Find all occurrences of newlines

Pattern: `\n`

Text:

This

Is

A

test

# Hexadecimal

Use hexadecimal code for characters (\x)

Problem: Find all occurrences of newlines

Pattern: \x0A

Text: this

is

a

Test

\x0A is hexadecimal code for newline character

# Unicode

Use Unicode for characters (\u)

Problem: Find all occurrences of newlines

Pattern: \u000A

Text: this

is

a

Test

\u000A is hexadecimal code for newline character

# Anchors

Word, String, and Line Boundary



# Word boundary

Use \b anchor to match at word boundaries

Problem: Find occurrences of word 'log'

Pattern: \blog\b

Text: catalog of log

# Start of string or line

Use ^ to check at beginning of string or line

Problem: Find occurrences of apple at start of string

Pattern: ^apple

Text: apple grows on apple tree

Set Multiline mode (m) flag to check at start of each line

Pattern: (?m)^apple

Text:

apple 1 grows on apple tree

apple 2 grows on apple tree

# End of string or line

Use \$ to check at end of string or line

Pattern: apple\$

Text: apple apple

Set Multiline mode (m) flag to check at end of each line

Pattern: (?m)apple\$

Text:

apple

apple

# Character Classes

# Digit character class

Match digits in all languages

Pattern: `\d`

Text: ABCD 1234567890 ௦௧௨௩௪௫௬௭௮௯

Negation

Match non-digits

Pattern: `\D`

Text: ABCD 1234567890\_ ௦௧௨௩௪௫௬௭௮௯

# Word character class

Match word characters, `_`, digits in all language

Pattern: `\w`

Text: Test `_` 1234. தமிழ்

## Negation

Pattern: `\W`

Text: Test `_` 1234. தமிழ்

# Form Validation

Specify regex pattern for HTML input controls using **pattern** attribute

User provided input must pass the regex check to be considered valid

The following input types support regex pattern: **text, tel, email, url, password, search**

Reference: <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/pattern>

# Pattern for Form Validation

Must be a valid JavaScript Pattern

```
let re = new RegExp("pattern", "u");
```

^, \$ anchors are implied (no need to specify them in the pattern)

Unicode flag is automatically set

No forward slashes around the pattern text ~~/pattern/~~

Reference: <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/pattern>



# USA Zip Code

Five-digit: 99999

Nine-digit: 99999-9999

Write a regex pattern to check for the above format

Specify a **pattern** attribute in the HTML input control



Chandra Lingam

57,000+ Students



For AWS self-paced video courses, visit:

<https://www.cloudwavetraining.com/>

