Profesor: Borja Rey Seoane

ANEXO IV

REGEX

Módulo: **Desenvolvemento Web en Contorno Cliente**

Ciclo: **DAW**

Introdución

As **expresións regulares** (REGEX[1] ou REGEXP), en ocasións tamén chamadas **expresións racionais** son un mecanismo de xeración de patróns de busca e comparación baseado en secuencias de carácteres. Orixinalmente naxeron coma ferramenta matemática, tal e coma as concebiu o seu principal creador, Stephen Cole Kleene, en 1951.

As REGEX permiten definir patróns complexos de coincidencia de texto, o que facilita tarefas como a validación de formatos, a substitución de fragmentos dun texto, ou a extracción de información específica.

Na actualidade a súa difusión é tal que son soportadas pola grande maioría de linguaxes de programación, sistemas operativos e por unha vasta estensión de *softwares* de todo tipo.

[1] Do inglés *REGular EXpression*.

Conceptos básicos

Podemos estruturar esta guía de introdución ás REGEX en torno a seis conceptos básicos:

- **Letras e números.**
- **Meta-carácteres.**
- **Cuantificadores.**
- **Grupos e alternativas.**
- **Clases de carácteres.**
- **Ancoras.**

Letras e números

As **letras** e **números** son os carácteres máis básicos que podemos empregar nas expresións regulares. Cando son incluídos directamente nunha REGEX, vaise a esixir a coincidencia literal do mesmo carácter dentro do texto.

Vexamos algúns exemplos:

- A expresión regular `a` coincidirá con calquera aparición da letra "`a`" nunha cadea:
  - Cadea: "`pato`"
  - Coincidencia: "`a`"
- A expresión `123` coincidirá coa secuencia exacta de números "`123`".
  - Cadea: "`a túa puntuación foi 123 puntos`"
  - Coincidencia: "`123`"
- A expresión `abc123` coincidirá coa secuencia exacta "`abc123`".
  - Cadea: "`A clave é abc123`"
  - Coincidencia: "`abc123`"
- A expresión `[a-zA-Z]` coincidirá con calquera letra maiúscula ou minúscula do alfabeto inglés.
  - Cadea: "`Galicia`"
  - Coincidencias: "`G`", "`a`", "`l`", "`i`", "`c`", "`i`", "`a`"

Meta-carácteres

Os **meta-carácteres** (en ocasións chamados simplemente comodíns) son aqueles carácteres que teñen un significado especial nas expresións regulares. Non coinciden co seu valor literal, senón que representan patróns máis complexos.

Velaquí algúns exemplos:

- A expresión `.` (punto) coincidirá con calquera carácter individual (ou ausencia del), agás co salto de liña.
    - Regex: `a.c`
    - Cadea: "`abc ac acd`"
    - Coincidencias: "`abc`", "`ac`"
- A expresión `\d` coincidirá con calquera díxito do 0 ao 9.
    - Regex: `\d\d\d`
    - Cadea: "`O número é 123`"
    - Coincidencia: "`123`"
- A expresión `\w` coincidirá con calquera carácter da palabra (letras, números ou o guión baixo)
    - Regex: `\w+`
    - Cadea: "`A miña variable é var_1`"
    - Coincidencias: "`A`", "`miña`", "`variable`", "`é`", "`var_1`"
- A expresión `\s` coincidirá con calquera espazo en branco (incluíndo espazos, tabulacións ou saltos de liña).
    - Regex: `\s`
    - Cadea: "`Ola Mundo`"
    - Coincidencia: " " (só o espazo entre as palabras)

Cuantificadores

Os **cuantificadores** permiten definir cantas veces debe aparecer un carácter ou grupo de carácteres nunha coincidencia.

De seguido algúns exemplos:

- A expresión * coincidirá cero ou máis veces coa expresión asociada.
    - Regex: `ab*`
    - Cadea: "`a, ab, abb`"
    - Coincidencias: "`a`", "`ab`", "`abb`"
- A expresión + coincidirá unha ou máis veces coa expresión asociada.
    - Regex: `ab+`
    - Cadea: "`a, ab, abb`"
    - Coincidencias: "`ab`", "`abb`" (non coincide con "`a`" porque non ten "`b`")
- A expresión ? coincidirá unha vez ou ningunha coa expresión asociada.
    - Regex: `colou?r`
    - Cadea: "`color, colour, colr`"
    - Coincidencias: "`color`", "`colour`"

- A expresión {n} coincidirá exactamente n veces coa expresión asociada.
    - Regex: `\d{4}`
    - Cadea: "`O ano 2024 foi intenso`"
    - Coincidencia: "`2024`" (catro díxitos)
- A expresión `{n,}` coincidirá polo menos n veces coa expresión asociada.
    - Regex: `\d{2,}`
    - Cadea: "`10, 123, 12345`"
    - Coincidencias: "`10`", "`123`", "`12345`"
- A expresión `{n,m}` coincidirá entre n e m veces coa expresión asociada.
    - Regex: `\d{2,4}`
    - Cadea: "`10, 123, 12345`"
    - Coincidencias: "`10`", "`123`" (pero non "`12345`", porque ten 5 díxitos)

Grupos e alternativas

Os **grupos** permiten agrupar partes dunha expresión regular entre si, namentres que as **alternativas** permiten definir opcións entre diferentes patróns.

Algúns exemplos do anterior:

- A expresión de grupo `()` permitirá agrupar unha parte da REGEX.
    - Regex: `(abc)+`
    - Cadea: "`abc abc abc`"
    - Coincidencia: "`abc abc abc`" (todo o grupo coincide repetidamente)
- A expresión de alternativa `|` permitirá escoller entre varias opcións.
    - Regex: `a|b`
    - Cadea: "`ac bd`"
    - Coincidencias: "`a`", "`b`"
- As expresións tamén poden combinar grupos con cuantificadores.
    - Regex: `(ab)+`
    - Cadea: "`ab ab abcd`"
    - Coincidencias: "`ab ab`"

Clases de carácteres

As **clases de carácteres** definen un conxunto de carácteres empregando os corchetes para acoutar. Calquera carácter dentro dos corchetes implicará unha coincidencia.

Algúns exemplos disto poden ser:

- **Clases básicas:** `[abc]` coincidirá con "`a`", "`b`" ou "`c`".
    - Regex: `[abc]`
    - Cadea: "`defabc`"
    - Coincidencias: "`a`", "`b`", "`c`"
- **Rangos de carácteres:** `[a-m]` coincidirá con calquera letra minúscula do alfabeto inglés que estea entre a "`a`" e a "`m`".
    - Regex: `[a-z]`
    - Cadea: "`gato`"
    - Coincidencias: "`g`", "`a`", "`t`", "`o`"
- **Clases negativa:** `[^abc]` coincidirá con calquera carácter que non sexa "`a`", "`b`" ou "`c`".
    - Regex: `[^abc]`
    - Cadea: "`abcdxyz`"
    - Coincidencias: "`d`", "`x`", "`y`", "`z`"
- **Conxunto de caracteres específicos:** `[aeiou]` coincide con calquera vogal minúscula.
    - Regex: `[aeiou]`
    - Cadea: "`exemplo`"
    - Coincidencias: "`e`", "`e`", "`o`"
- **Díxitos e letras:** `[0-9A-Fa-f]` coincide con calquera díxito ou letra hexadecimal (`0-9` e `A-F` -en maiúsculas ou en minúsculas-).
    - Regex: `[0-9A-Fa-f]`
    - Cadea: "`123ABCdef`"
    - Coincidencias: "`1`", "`2`", "`3`", "`A`", "`B`", "`C`", "`d`", "`e`", "`f`"

Algunhas clases que compre salientar son aquelas que teñen nome propio, como son:

- `[:alpha:]` é a clase correspondente aos carácteres alfabéticos (maiúsculos e minúsculos).
- `[:digit:]` é a clase correspondente ás cifras da numeración árabe.
- `[:alnum:]` é a clase correspondente á unión das dúas anteriores (carácteres alfanuméricos).
- `[:punct:]` é a clase correspondente aos carácteres de puntuación (sendo estes **`! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~ .`**)
- `[:graph:]` é a clase correspondente á unión das dúas anteriores, isto é aos carácteres típicos de escritura (alfanuméricos máis os de puntuación).
- `[:blank:]` é a clase correspondente ao espazo e a tabulación.
- `[:cntrl:]` é a clase correspondente aos carácteres de control (en ASCII son aqueles cuxos códigos octais van do 037 ao 177).

- `[:lower:]` é a clase correspondente aos carácteres alfabéticos minúsculos.
- `[:upper:]` é a clase correspondente aos carácteres alfabéticos maiúsculos.
- `[:print:]` é a clase correspondente aos carácteres imprimibles, é dicir, os alfanuméricos, os de puntuación e o espazo.
- `[:space:]` é a clase correspondente aos carácteres de espazamento (tabulación, nova liña, tabulación vertical, sangría, retorno de carro e espazo).
- `[:xdigit:]` é a clase correspondente aos díxitos hexadecimais (aqueles que son letras poden ir tanto en maiúsculos coma en minúsculos).

Áncoras

As **áncoras** son símbolos que permiten establecer a posición na que debe coincidir o patrón dentro da cadea de texto.

Exemplos de uso das áncoras serían:

- A expresión `^` indicará o inicio dunha cadea (é dicir, cadeas que inicien con esa expresión que figure a continuación do circumflexo).
  - Regex: `^Ola`
  - Cadea: "`Ola Mundo`"
  - Coincidencia: "`Ola`" (só ao inicio da cadea)
- A expresión `$` indicará o remate dunha cadea (é dicir, cadeas que rematen coa expresión que figure xusto antes do símbolo do dólar).
  - Regex: `mundo$`
  - Cadea: "`Ola mundo`"
  - Coincidencia: "`mundo`" (só ao final da cadea)
- A expresión `\b` coincidirá co límite dunha palabra (xa sexa no comezo ou no remate).
  - Regex: `\bcarro` (palabras que comecen por "`carro`")
  - Cadea: "`O carro está aquí, pero non a carroza.`"
  - Coincidencia: "`carro`", "`carro`" (a parte coincidente de "`carroza`")
- A expresión `\B` coincidirá só cando non haxa límite de palabra.
  - Regex: `\Bola`
  - Cadea: "`ola a todos, pasádeme a bola`"
  - Coincidencia: "`ola`" (a que fai parte de "`bola`")

Bibliografía

- Eloquent JavaScript. (2018). Regular Expressions. [online] Dispoñible en: https://eloquentjavascript.net/09_regexp.html [Accedido o 10 de outubro de 2024].
- Friedl, J.E.F. (2006). Mastering Regular Expressions (3ª ed.). Sebastopol: O'Reilly Media.
- JavaScript.info. (2020). Expresións regulares. [online] Dispoñible en: https://javascript.info/regular-expressions [Accedido o 10 de outubro de 2024].
- MDN Web Docs. (2024). Regular expressions - JavaScript | MDN. [online] Dispoñible en: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions [Accedido o 10 de outubro de 2024].
- Regex101. (2024). Online regex tester and debugger. [online] Dispoñible en: https://regex101.com/ [Accedido o 10 de outubro de 2024].
- W3Schools. (2024). JavaScript Regular Expressions. [online] Dispoñible en: https://www.w3schools.com/js/js_regexp.asp [Accedido o 10 de outubro de 2024].

ÍNDICE

INTRODUCIÓN **1**

CONCEPTOS BÁSICOS **2**

LETRAS E NÚMEROS 2

META-CARÁCTERES 2

CUANTIFICADORES 3

GRUPOS E ALTERNATIVAS 4

CLASES DE CARÁCTERES 4

ÁNCORAS 6

BIBLIOGRAFÍA **7**