

Expresiones regulares.

- EXPRESIONES REGULARES. 1
- 1. MOTIVACIÓN 2
- 2. EJERCICIO 4
 - 2.1. VALIDACIÓN DE FORMULARIOS CON EXPRESIONES REGULARES 4

1. Motivación

Las expresiones regulares son modelos que describen las combinaciones de caracteres en el texto. Se podrían definir como una serie de caracteres que forman un patrón, que representan a otro grupo de caracteres mayor, de tal forma, que podemos comparar el patrón con otros conjuntos de caracteres para ver las coincidencias.

Las expresiones regulares pueden utilizarse en múltiples lenguajes de programación incluido javascript, en concreto, en javascript son muy útiles para la validación de formularios.

La tabla siguiente contiene los caracteres especiales de las expresiones regulares:

Carácter	Texto buscado
^	Principio de entrada o línea
\$	Fin de entrada o línea
*	El carácter anterior cero o más veces
+	El carácter anterior una o más veces
?	El carácter anterior una vez como máximo (es decir, indica que el carácter anterior es opcional).
.	Cualquier carácter individual, salvo el de salto de línea.
x y	x o y
{n}	Exactamente n apariciones del carácter anterior.
{n, m}	Como mínimo n y como máximo m apariciones del carácter anterior.
[abc]	Cualquiera de los caracteres entre corchetes. Especifique un rango de caracteres con un guión (por ejemplo, [a-f] es equivalente a [abcdef]).
[^abc]	Cualquier carácter que no esté entre corchetes. Especifique un rango de caracteres con un guión (por ejemplo, [^a-f] es equivalente a [^abcdef]).
\b	Límite de palabra (como un espacio o un retorno de carro).
\B	Cualquiera que no sea un límite de palabra.
\d	Cualquier carácter de dígito. Equivalente a [0-9].
\D	Cualquier carácter que no sea de dígito. Equivalente a [^0-9].
\f	Salto de página

\n	Salto de línea.
\r	Retorno de carro
\s	Cualquier carácter separador (espacios, tabulaciones, saltos de página o saltos de línea).
\S	Cualquier carácter que no sea separador.
\t	Tabulación
\w	Cualquier carácter alfanumérico, incluido el de subrayado. Equivalente a [A-Za-z0-9_].
\W	Cualquier carácter que no sea alfanumérico ni subrayado. Equivalente a [^A-Za-z0-9_].

El patrón es una descripción del texto que se está buscando y JavaScript encontrará las subcadenas que concuerdan con ese patrón o definición. Las expresiones regulares se usan con el objeto `Regular Expression` y también dentro de los métodos `String.match`, `String.replace`, `String.search` y `String.split`. La tabla siguiente contiene algunos de los patrones más utilizados a la hora de validar formularios.

Patrón	Expresión regular
Cualquier letra en minúscula	[a-z]
Entero	^(?:\+ -)?\d+\$
Correo electrónico	/[\w-\.] {3,}@[([\w-] {2,} \.)* ([\w-] {2,} \.)* [\w-] {2,4} /
URL	^(ht f)tp(s?)\:\/\/[0-9a-zA-Z]([-.\w]*[0-9a-zA-Z])*(:(0-9)*)*(\/?)([a-zA-Z0-9\-\.\? \, '\\/\+\&%\$#_]*)?\$
Contraseña segura	(?!^[0-9]*\$)(?!^[a-zA-Z]*\$)^([a-zA-Z0-9]{8,10})\$ (Entre 8 y 10 caracteres, por lo menos un dígito y un alfanumérico, y no puede contener caracteres espaciales)
Fecha	^\d{1,2}\/\d{1,2}\/\d{2,4}\$ (Por ejemplo 01/01/2007)
Hora	^(0[1-9] 1\d 2[0-3]):([0-5]\d):([0-5]\d)\$ (Por ejemplo 10:45:23)
Número tarjeta de crédito	^((67\d{2}) (4\d{3}) (5[1-5]\d{2}) (6011))(-?\s?\d{4}){3}((3[4,7])\d{2}-?\s?\d{6}-?\s?\d{5})\$
Número teléfono	^[0-9]{2,3}-? ?[0-9]{6,7}\$
Código postal	^([1-9]{2} [0-9]{1-9} [1-9]{0-9})[0-9]{3}\$
NIF	^(X(- \.)?0?\d{7}(- \.)?[A-Z] [A-Z](- \.)?\d{7}(- \.)?[0-9A-Z] \d{8}(- \.)?[A-Z])\$

En el siguiente ejemplo, veremos una función javascript que valida la correcta introducción de un password con los siguientes requisitos:

- Longitud entre 8 y 10 caracteres.
- Al menos un número y una letra.
- No pueden haber caracteres especiales.

```
function validatePass (campo)
{
    var RegExPattern = /^(?!^[0-9]*$)(?!^[a-zA-Z]*$)^[a-zA-Z0-9]{8,10}$/;

    if ((campo.value.match(RegExPattern)) && (campo.value!=''))
        alert('Password Correcta');
    else
        alert('Password Incorrecta.');
```

En el ejemplo, la función recibe como parámetro el campo del formulario que estamos validando, la forma de llamar a la función sería la siguiente:

```
Var campo = document.getElementById("password");
Var passwordOK = validatePass(campo);
```

2. Ejercicio

2.1. Validación de formularios con expresiones regulares

Modificar las funciones de validación de formularios del ejercicio del tema anterior (Firebug), de forma que utilicen expresiones regulares para realizar las distintas validaciones que se piden.