

DESARROLLO WEB EN ENTORNO CLIENTE

JAVASCRIPT
EXPRESIONES REGULARES

Coincidencias Básicas

- . - Cualquier Caracter, excepto nueva línea
- \d - Cualquier Dígitos (0-9)
- \D - No es un Dígito (0-9)
- \w - Caracter de Palabra (a-z, A-Z, 0-9, _)
- \W - No es un Caracter de Palabra.
- \s - Espacios de cualquier tipo. (espacio, tab, nueva línea)
- \S - No es un Espacio, Tab o nueva línea.

Límites

- \b - Límite de Palabra
- \B - No es un Límite de Palabra
- ^ - Inicio de una cadena de texto
- \$ - Final de una cadena de texto

Cuantificadores:

- * - 0 o Más
- + - 1 o Más
- ? - 0 o Uno
- {3} - Número Exacto
- {3,4} - Rango de Números (Mínimo, Máximo)

Conjuntos de Caracteres

- [] - Caracteres dentro de los brackets
- [^] - Caracteres que NO ESTAN dentro de los brackets

Grupos

- () - Grupo
- | - Uno u otro

EJERCICIO 1

Crea un pequeño formulario con un cuadro para introducir texto y un botón para testear las expresiones regulares siguientes:

1. Encuentra cualquier número de tres dígitos en una cadena:
2. Encuentra todas las palabras que comienzan con "g" en una cadena:
3. Encuentra todas las ocurrencias de letras seguidas por un número:
4. Encuentra todas las ocurrencias de espacios en blanco seguidos por una palabra:
5. Encuentra todas las ocurrencias de palabras que no terminan en "s":
6. Encuentra todas las líneas que comienzan con "Inicio":
7. Encuentra todas las líneas que terminan con "Fin":
8. Encuentra todas las ocurrencias de palabras que contienen al menos una vocal:
9. Encuentra todas las ocurrencias de números de teléfono en el formato (123) 456-7890:
10. Encuentra todas las ocurrencias de palabras que tienen al menos tres letras:
11. Encuentra todas las ocurrencias de palabras que no contienen la letra "e":
12. Encuentra todas las ocurrencias de palabras que comienzan con "a" o "b":
13. Encuentra todas las ocurrencias de palabras que terminan con "o" o "a":
14. Encuentra todas las ocurrencias de palabras que tienen exactamente cinco letras:
15. Encuentra todas las ocurrencias de palabras que comienzan con "c" y terminan con "o":

EJERCICIO 2

Interpretar y explicar las siguientes expresiones regulares:

1. `/^\d{3}-\d{2}-\d{4}$/` → La cadena empieza con 3 dígitos, después un guión, 2 dígitos, otro guión y termina con 4 dígitos. Ejemplo: 123-45-6789
2. `/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/` → La cadena empieza con un conjunto de 1 o más caracteres de los cuales los permitidos son: mayúsculas, minúsculas, números y los caracteres que están dentro. Seguido de esto debe ir un @ y otra cadena como la anterior. Al final de esta cadena debe haber un punto y seguido de este, mínimo dos caracteres sean minúsculas o mayúsculas indicando el final del campo. Ejemplo: adAD@gmail.es
3. `/^([01]?[0-9])?([20-3]):[0-5][0-9]$/` → El campo puede o no empezar con 01 y seguido de este, o bien, un número del 0 al 9 o dos números del 0 al 3, y, obligatorio después de esto debe haber dos puntos y tras el, un número del 0 al 5 y que la cadena termine con un número del 0 al 9. Ejemplo: 12:58
4. `/^(http|https):\/\/[a-zA-Z0-9]+\.[a-zA-Z]{2,}$/` → Que la cadena empiece con *http* o bien con *https* y seguido de este, obligatoriamente dos puntos con dos barras / Seguido de esto debe haber un conjunto de 1 o más caracteres. Seguido de esto, un punto y que la cadena termine con como mínimo una cadena de dos caracteres. Ejemplo: https://marca.es
5. `/^[^0-9]+$/` → La cadena debe empezar y terminar con uno o más caracteres de los cuales no sea un número. Ejemplo: rubenesfeo
6. `/(\b\w+\b).*/` → La cadena debe empezar con 1 o más caracteres que pueden ser tanto minúsculas como mayúsculas como números o como una barra baja y debe ir seguido de cualquier carácter desde longitud 0 hasta la que sea. Ejemplo: adri5

7. `/^(\+|-)?\d+$/` → La cadena debe empezar con el signo + o con el signo - y puede o no estar. Esta cadena termina con 1 o mas caracteres que sean numeros. Ejemplo: +89
8. `/^[aeiou]([a-z]|[0-9])+$/i` → La cadena empieza por una vocal y seguido puede ser, o bien una letra o bien un numero. Puede haber uno o mas de estos y debe ser la parte final de la cadena. Acepta tanto minusculas como mayusculas. Ejemplo: a45
9. `/^\d{1,3}(\,\d{3})*(\.\d+)?$/` → La cadena debe empezar como minimo 1 numero y como maximo 3. Seguido de una coma agrupa un numero de tres digitos que puede repetirse desde 0 hasta n veces y puede o no terminar la cadena con un punto y uno o mas digitos. Ejemplo: 23,456.45
10. `/^\s*(.*?)\s*$/` → La cadena empieza con 0 o mas espacios en blanco, puede o no haber desde 0 a N cantidad de caracteres y debe terminar con 0 o mas espacios en blanco. Ejemplo: (espacio)23(espacio)
11. `/^([01]?[0-9]|2[0-3]):[0-5][0-9]\s(am|pm)$/i` → La cadena debe empezar pudiendo o no ser 01, seguido de esto, o bien, un numero o dos numeros del 0 al 3. Obligatoriamente despues, dos puntos, un numero del 0 al 5 y un numero del 0 al 9. Seguido de esto, un espacio y, o bien, la cadena *am* o bien la cadena *pm* y puede ser tanto en mayusculas como en minusculas y debe ser el final de la cadena. Ejemplo: 23:45 PM
12. `/^\d{1,2}\.\d{1,2}\.\d{4}$/` → La cadena debe empezar siendo minimo un digito y como maximo dos y seguido de este una barra. A continuacion, como minimo un digito y como maximo dos y le sigue otra barra. Al final de la cadena un digito de 4 numeros. Ejemplo: 23/12/2004