

## Expresiones regulares: Ejercicios

**1. Usando el texto de la Declaración de Derechos Humanos, que está en el archivo `derechos_humanos.txt` en los materiales del tutorial:**

- 1.1. ¿Cuántas palabras terminan en "tad"?
- 1.2. ¿Cuál es el sufijo nominal más frecuente entre los siguientes: *-tad*, *-dad*, *-ión*, o *-iento*?
- 1.3.
  - a) ¿Cuántas veces aparece la preposición "de"?
  - b) ¿Haciendo una búsqueda sensible a mayúsculas obtendríamos el mismo resultado?
- 1.4. Dar una expresión que encuentre palabras relacionadas con la noción de "libertad" y solamente tales palabras (p. ej. el nombre "libertad", el adjetivo "libre", el verbo "liberar" etc.)  
  
¿Deberíamos cambiar la expresión si las palabras "libro", "librería", o el nombre propio "Delibes" fueran parte del texto?

**2. Dar una expresión regular que encuentre todas las palabras que empiezan con "b" (mayúscula o minúscula) en la oración siguiente, y solo esas palabras:**

La boa que vive en el baobab de Bob cerca de la casa de Rob

**¿Qué expresión encontraría las palabras que terminan con "b" (independientemente de mayúscula/minúscula)?**

**3. Dada la expresión regular `^[^^]`**

- a) proporcionar una cadena de caracteres que no coincida con la expresión
- b) proporcionar una cadena de caracteres que coincida con la expresión

**4. Dada la expresión regular `^[^$]+`**

- a) proporcionar una cadena de caracteres que no coincida con la expresión
- b) proporcionar una cadena de caracteres que coincida con la expresión

**5. Transforma "Adoro las expresiones regulares" en "las expresiones regulares, las adoro ", sin usar símbolos literales en el contexto de la expresión.**

(El *contexto* es la expresión de búsqueda (casilla *find*), donde indicamos los grupos dentro del patrón a buscar, si es necesario. Y la sustitución es la expresión que usamos en la casilla *replace*, en la cual podemos hacer referencia a los grupos con la notación \1, \2, ..., \9).

**6. Doblar *nuestros ingresos*. Dicho de otro modo, dada la cadena *nuestros ingresos*, proporcionar una expresión regular que escribe *nuestros ingresos nuestros ingresos*.**

**7. Dar dos expresiones regulares que empiecen y terminen con la letra *a*, y que se comporten de la forma descrita a continuación para la oración *abra el sistema*:**

- a) la primera expresión debe corresponder solo a la primera palabra, *abra*
- b) la segunda expresión debe corresponder a la oración completa, *abra el sistema*

**8. Transformar una *boa* en un *baobab* sin usar caracteres literales en el contexto de la regex. En otras palabras, dar una regex que reescribe la palabra *boa* como *baobab* sin usar símbolos literales en el contexto.**

**9. Dar una expresión que corresponda a todas las palabras que contengan “o” o bien “ó” en el poema del archivo *yurkievich\_gloton.txt* que se encuentra en los materiales del tutorial, y que proviene del libro *Rimbomba* publicado por Saúl Yurkiévich en 1978.**

**10. La idea en este ejercicio es tomar un archivo parcialmente marcado en TEI (un estándar ampliamente usado en humanidades, en formato XML) y completar el marcado. El archivo con marcado incompleto es *tei\_incomplete\_example.xml* en los materiales del curso.**

**El objetivo es añadir etiquetas <l> a los versos y etiquetas <lg> a las estrofas:**

- Al principio de cada verso poner <l> y al final de cada verso poner </l>
- Al principio de cada estrofa poner <lg> y al final poner </lg>

**Puedes usar varias expresiones regulares.**

11. En la columna Input tenemos marcado HTML incompleto. Está incompleto porque las filas (<tr>) no tienen celdas. La tarea consiste en añadir celdas (<td>) *del color necesario*, usando expresiones regulares. El color se expresa en cada caso como el valor de un atributo *class* añadido a cada celda, como se ve en la columna *Salida 1*.

## Entrada

```
<html>
<head>
<style>
  body {
    font-family:sans,sans-serif;
  }
  .red{
    color:red;
    font-size:100px;
  }
  .blue{
    color:blue;
    font-size:100px;
  }
</style></head>
<body>
<table>
<tr>red blue</tr>
<tr>blue red</tr>
<tr>red blue</tr>
<tr>blue red</tr>
</table>
</body>
```

## Salida 1

```
<html>
<head>
<style>
  body {
    font-family:sans,sans-serif;
  }
  .red{
    color:red;
    font-size:100px;
  }
  .blue{
    color:blue;
    font-size:100px;
  }
</style>
</head>
<body>
<table>
<tr><td class="red">red</td> <td class="blue">blue</td></tr>
<tr><td class="blue">blue</td> <td class="red">red</td></tr>
<tr><td class="red">red</td> <td class="blue">blue</td></tr>
<tr><td class="blue">blue</td> <td class="red">red</td></tr>
</table>
</body>
```

**Salida 1** se ve como sigue en un navegador (como Firefox, Chrome u otros):

red blue  
blue red  
red blue  
blue red

Output 1

¿Qué expresiones regulares serían necesarias para modificar el HTML en Salida 1 para que se vea como en la imagen siguiente (que llamaremos Salida 2)?:

rougebleu  
bleu rouge  
rougebleu  
bleu rouge

Output 2

¿Qué expresiones regulares usarías para cambiar el HTML de Salida 2 para que se vea como la imagen Salida 3 abajo? (y que crea un [efecto Stroop](#)).

bleu rouge  
rougebleu  
bleu rouge  
rougebleu

Salida 3