Homework. Task 2.

Implement a function(I've named it replaceElements, you can choose whatever name you want) that will have a possibility to change the tag of HTML element with a new tag, by providing css selector of the tag that should be changed and the name of new tag.

New tag should have all attributes, child nodes available from an old one. That function also should have a possibility for adding new attributes to the new tag element by accepting array of objects with 'key'(attribute name) and 'value'(attribute value) properties.

E.g.:

If I have such html fragment:

<ul class='background' data-color='gray'>

<!-- red -->

<li>item 1</li>

<!-- orange -->

<span> I am not a &lt li &gt tag !!!</span>

<li>item 2</li>

<!-- red -->

<!-- cyan -->

<li>item 3</li>

</ul>

And apply my function:

replaceElements('ul.background', 'ol', [

{ key: 'type', value: 'a' },

{ key: 'style', value: 'border: 1px dashed red; padding: 15px 35px;' }

]);

replaceElements('span', 'i', [ { key: 'style', value: 'text-align: center; display: block;' } ]);

I will get such result(html):

<ol class='background' data-color='gray' type='a' style='border: 1px dashed red; padding: 15px 35px;'>

<!-- red -->

<li>item 1</li>

<!-- orange -->

<i style='text-align: center; display: block;' > I am not a &lt li &gt tag !!!</i>

<li>item 2</li>

<!-- red -->

<!-- cyan -->

<li>item 3</li>

</ol>

Try to use next functions:

* \*.querySelector\*
* document.createElement
* \*.setAttribute
* \*.replaceChild
* \*.cloneNode