

# «Волга-IT'XXI» Финал

На многих сайтах структура навигации передается на клиент в виде JSON. В этом задании вы попробуете создать навигационное меню согласно полученной из базы данных структуре.

## Задача

Ваша задача - написать функцию, получающую на вход данные в виде JSON и возвращающую DOM дерево.

JSON содержит два поля: `structure`, в котором описано дерево вложенности и типы компонентов, и `components` с описанием общих параметров для каждого типа компонентов.

Каждый элемент в дереве структуры может быть либо **строкой**, либо **объектом** одного из следующих типов:

- **node** - представляет из себя обычный `div`, всегда содержит поле `children` с дочерними элементами.
- **link** - представляет из себя тег `a`.
- **button** - это элемент `button`.

Каждый элемент типа **link** или **button** кроме поля `children` может содержать дополнительные параметры, например, `"href": "go/to/owen-lars"`, которые должны стать атрибутами DOM элемента. Эти параметры всегда имеют приоритет над "общими", которые указаны в `components`.

Пример входных данных:

```
{
  "structure": [
    {
      "type": "node",
      "children": [
        {
          "type": "link",
          "props": { "href": "go/to/owen-lars", "children": ["Owen Lars"] }
        },
        {
          "type": "link",
          "props": { "href": "go/to/obi-wan-kenobi", "children": ["Obi-Wan Kenobi"] }
        }
      ]
    },
    {
      "type": "node",
      "children": [
```

```

    {
      "type": "link",
      "props": { "href": "go/to/mon-mothma", "children": ["Mon Mothma"] }
    },
    { "type": "button", "props": { "children": ["Kit Fisto"] } }
  ]
},
{
  "type": "node",
  "children": [
    { "type": "button", "props": { "children": ["Palpatine"] } },
    {
      "type": "link",
      "props": { "href": "go/to/jango-fett", "children": ["Jango Fett"] }
    }
  ]
},
{
  "type": "node",
  "children": [
    {
      "type": "link",
      "props": { "href": "go/to/yarael-pooof", "children": ["Yarael Poof"] }
    },
    { "type": "button", "props": { "children": ["Bib Fortuna"] } },
    { "type": "button", "props": { "children": ["Nien Nunb"] } },
    { "type": "button", "props": { "children": ["Dooku"] } },
    { "type": "link", "props": { "href": "go/to/watto", "children": ["Watto"] } }
  ]
}
],
"components": {
  "link": {
    "style": "color:red;font-size:20px;"
  },
  "button": {
    "style": "display:block;width:100%;border:none;height:30px;"
  }
}
}

```

## Оформление

Файл с решением следует оформить по шаблону:

```

module.exports = function createMenu(navigationData) {
  return // DOM дерево
}

```

Отлично, теперь доработаем нашу навигацию, и позволим администратору сайта менять порядок ссылок в меню.

## Задача

Используя написанную функцию генерации DOM дерева и данные приведенные в качестве примера из первой задачи, создайте разметку для файла index.html. Используя [HTML Drag & Drop API](#) предоставьте пользователю возможность менять местами элементы **link** и **button** внутри одного div элемента. Добавьте кнопку "Сохранить" по клику на которую создайте и выведите в консоль JSON структуру навигации с учетом сделанных перестановок.

## Оформление

Файл с решением следует оформить по шаблону:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Admin</title>
    <meta charset="UTF-8" />
    <style>
      #app-menu {
        width: 300px;
      }
      #app-menu div {
        padding-left: 10px;
      }
      #app-menu a, #app-menu button {
        display: block;
        margin: 10px 0;
        text-align: center;
      }
    </style>
  </head>

  <body>
    <div id="app-menu">
      <!-- Меню созданное с помощью JSON структуры из первого задания -->
      <div>
        <button>Сохранить</button>
        <script>
          // Ваш код для Drag & Drop и "Сохранения"
        </script>
      </div>
    </div>
  </body>
</html>
```