

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

по лабораторной работе № 3

Дисциплина: Языки интернет программирования

Москва, 2022

Написать Javascript-код для вывода дерева элементов страницы, с которой этот код запущен. Отступы для отображения формировать как символ

В процессе выполнения работы реализовать следующие пункты:

- 1.Сформировать страницу с произвольным кодом разметки, но обеспечить уровень вложенности внутри элемента <body> не менее 3.
- 2.добавить внутри элемента <body> секцию <div>, предназначенную для вывода результата обхода дерева элементов страницы.
- 3.Выбрать способ активации рекурсивной программы обхода дерева элементов, реализовать и подключить эту программу.
- 4.При проходе по узлам разметки обеспечить отладочный вывод в консоль. Привести в отчете содержимое консоли.
- 5.Реализовать вывод на странице.

В отчете привести код страницы с программой обхода, отладочный вывод в консоль и примеры обхода дерева элементов страницы.

Код программы:

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>JS TEST</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<div id="Alexandre_Dumas" class="text center">
<ul>
<li class="purple-text">FRANÇOIS I., King of France.</li>
<div class="container">
<li> HENRI II., his successor.</li>
<div class="container">
<li> CATHERINE DE MÉDICIS, Queen to Henri II.</li>
<li> THE DAUPHIN, afterwards François II.</li>
<div class="container">
<li> HENRI, his brother, afterwards Henri III.</li>
</div>
</div>
<li> MARY STUART, married to the Dauphin.</li>
<li> MARY, Queen of England.</li>
<li> DUC D'ORLÉANS, afterwards Charles IX.</li>
<div class="container">
<li> MARGUERITE DE FRANCE, sister of Henri II.</li>
```

```

<li> MARGUERITE DE VALOIS, daughter of Henri II.</li>
</div>
<li> PRINCESS ÉLISABETH.</li>
<li> FRANÇOIS, Duc d'Alençon.</li>
<li> DUC DE GUISE, Lieutenant-General of France.</li>
<li> MONSEIGNEUR LE CARDINAL DE LORRAINE, his brother.</li>
</div>
<li class="purple-text">DUC D'AUMALE, brother of Duc de Guise.</li>
</ul>
</div>
<div id="test" class="text center">
<h2>Header 2</h2>
<div>Div first
<div>Div second</div>
</div>
<h3>Head 3
<p>Text paragraph</p>
<div>Simple Div</div>
</h3>
</div>
<span class="gradline"></span>

<section class="interface">
<h1 class="purple-text center">Section for Results</h1>
<div class="output center">
<a onclick="out(this)">View</a>
<a onclick="print(get_all(document.getElementById('test')))">View 2</a>
<a onclick="print(get_all(document.getElementById('Alexandre_Dumas')))">View 3</a>
<a onclick="print2(get_all(document.getElementById('Alexandre_Dumas')))">View 3</a>
</div>
</section>
<div id="output" class="text">
</div>

</body>
<script>
function out(el) {
var output = document.getElementById('output');

var par = document.createElement('div');
var text = document.createTextNode(
"<" + (el.nodeName) + "> "
+ el.textContent +
" <" + el.nodeName + ">");

par.appendChild(text);
output.appendChild(par);
}

//Requires
function print2(el) {
let mas_el = el.split("\n");
mas_el = mas_el.filter(element => element.trim());

```

```

let output = document.getElementById('output');
mas_el.forEach(element => {
while(element.includes(' ')){
element = element.replace(' ', '\u00A0');
}
let par = document.createElement('p');
let text = document.createTextNode(element)

console.log(element);
par.appendChild(text);
output.appendChild(par);
});
}
function print(el) {
let output = document.getElementById('output');

let par = document.createElement('div');
let text = document.createTextNode(el)

console.log(el);
par.appendChild(text);
output.appendChild(par);
}
function get_all(el) {
if (el.childNodes.length)
{
let result = "<" + el.nodeName + ">";
for (let index = 0; index < el.childNodes.length; index++){
result += ' ' + get_all(el.childNodes[index]);
console.log(result + '\n\n');
}
el.childNodes.length === 1? result+= " ": result += "\n"; //if 1 kid close string on 1 line, else add \n
result += "</" + el.nodeName + ">";

return result;
} else
return el.textContent;
}
</script>
</html>

```

Style.css:

```

body {
background: #000;
}
.container {
padding-left: 40px;
margin-bottom: 15px;
}
.center {

```

```
align-items: center;
display: flex;
justify-content: center;
}
.text {
font-size: 17px;
color: #cecece;
font-family: Arial, Helvetica, sans-serif;
font-weight: 600;
letter-spacing: -0.022em;
}
.purple-text {
background-image: linear-gradient(180deg,#ffb6ff,#b344ff 10%,#ae38ff 33%,
#ffb6ff 45%,#ffe3ff 50%,#ffb6ff 66%,#b344ff);
animation: gradAnimation 8s infinite;
background-position: 50% 0;
background-size: 100% 300%;
margin: 6px 0;
padding-top: 0.5px;

color: transparent;
-webkit-background-clip: text;
}
span.gradline {
display: block;
overflow: hidden;

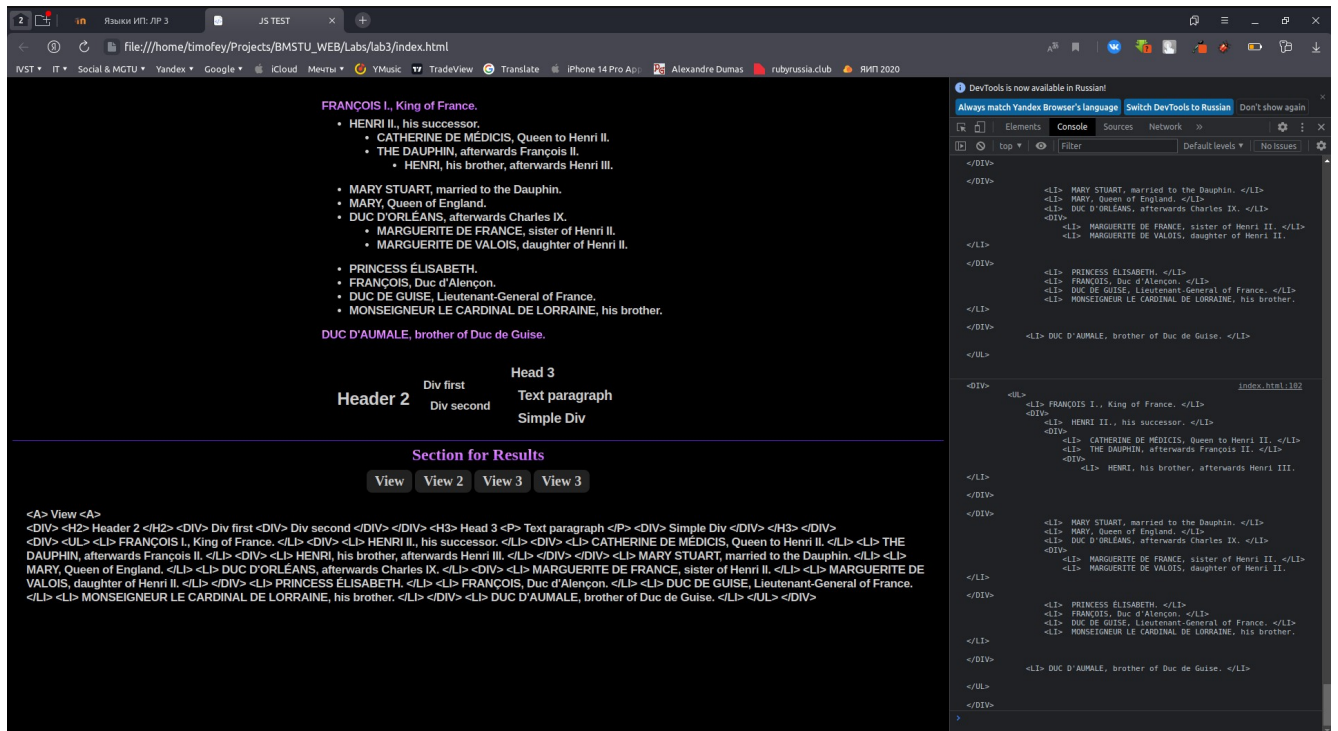
width: 100%;
height: 1px;
background-color: rgb(91, 26, 211);
animation: slidein 3s ease-in-out both;
}

.output a {
position: relative;
text-align: center;
border-radius: 10px;
padding: 5px 12px;
font-size: 20px;
font-weight: 800;
color: #cecece;
transition: 0.5s;
border: 2px solid #000;
background: #222;
}
.output a:hover {
border: 2px solid #cecece;
cursor: pointer;
}
#test *{
margin: 10px;
}
```

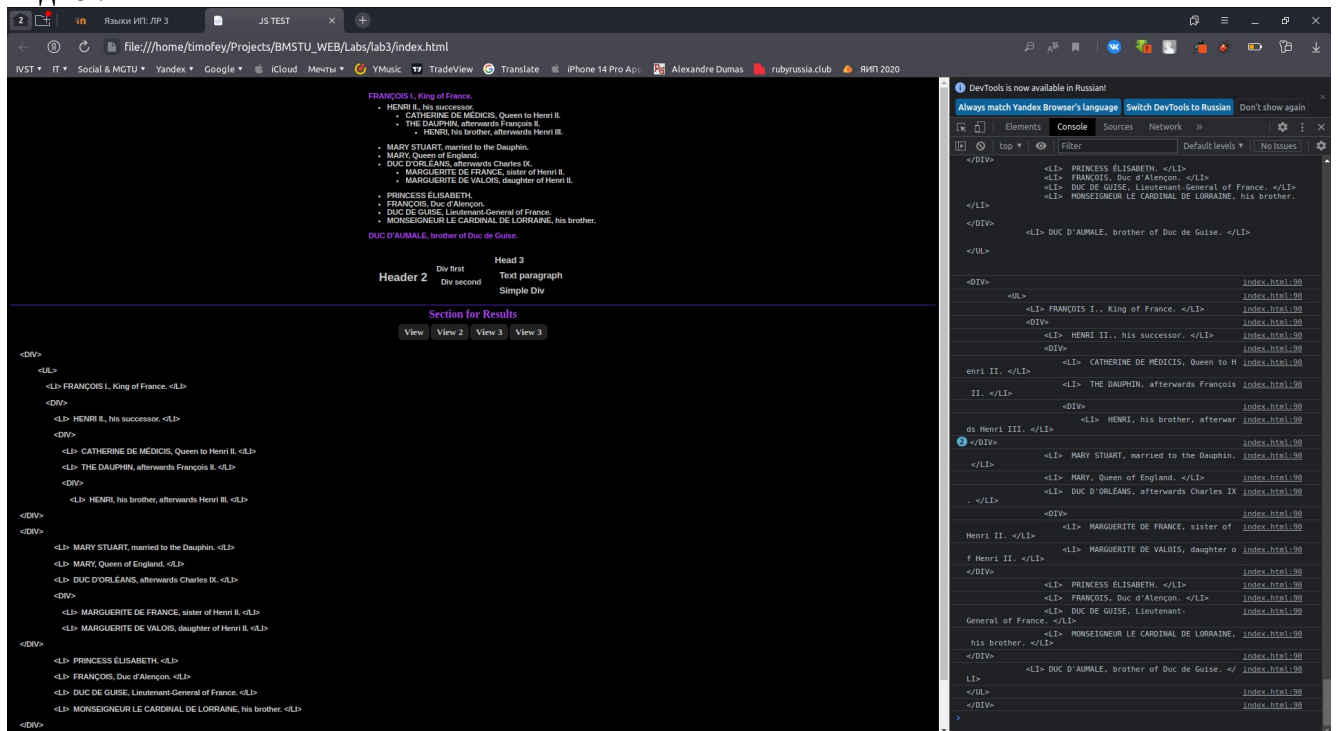
```
#output {  
margin: 20px;  
}
```

```
@keyframes gradAnimation {  
0% {  
background-position: 50% 0;  
}  
25% {  
background-position: 50% 100%;  
}  
50% {  
background-position: 50% 100%;  
}  
75% {  
background-position: 50% 0;  
}  
to {  
background-position: 50% 0;  
}  
}  
@keyframes slidein {  
0% {  
transform: ScaleX(0);  
}  
to {  
transform: ScaleX(100%);  
}  
}
```

Активацию программы выбрал на нажатие кнопок View, View 2, View 3. На скриншоте вывод нажатых первых трех:



Нажатие последней реализовал с корректным переносом, она показывает иерархию, которую видно :



Вывод: Научился добавлять динамическое поведение с использованием Javascript.