Лабораторийн ажил 8. CV генератор

Даалгавар

Энэ удаагийн лабораторийн ажлаар бид өмнөх лаборторийн ажил дээр бүтээсэн оюутны бүртгэлийн ФОРМ/маягт-г ашиглана. Тухайн формын элементийн утгыг JavaScript ашиглан авах, Хичээл сонголт хэсэгт таг хэвлэх аргуудыг хэрэглэн НЭМЭХ-товч хийж нийт маягтийг бөглөөд YYСГЭХ-BUTTON товч дарахад лабораторийн ажил 2 дээр хийсэн CV загварын дагуу маягтад бөглөсөн утгуудыг ашиглан CV дэлгэцэнд хэвлэнэ.

ДаалгаварN№1

Форм дээрх элментүүдийн утгыг авч товч дарахад хуудасны доод хэсэг шинэ DIV Дээр CV – ЛАБ№2 дээрх шиг харагдахаар үүсгэнэ.

- Форм бүхий лаб-н файлыг нээж ЛАБ8 болгож дахин нэрлэж хадгалах
- Формын элементүүд руу document.getElementById(id), document.getElementsByTagName(name) ашиглаж хандаж утгуудыг авна
- Хуудасны доод хэсэгт шинэ <DIV name='generatedCV'> үүсгэ
- Формоос авсан утгуудыг загварчиж/элементүүд нэмж загварчилах –хэвшүүлэх, хүнэгт/ CV үүсгэ

ДаалгаварN№2

Хичээл сонгол хэсэгт Хичээл сонгоод НЭМЭХ-BUTTON товч дарахад доорх жагсаалтанд нэмэгддэг боломж нэмэх

- Select элемент –д хичээлийн мэдээллийг нэмэх
- Сонгосон утгыг НЭМЭХ-BUTTON товч дарахад доорх хүснэгтэд мөр болгон нэмэх

ДаалгаварN№3

Форм дээрх элментүүдийн утгыг onchange ивент ашиглаж зөв буруу орууж байгаа эсхийг шалгах.

Үнэлгээ/Ашиглах

- document.getElementById(id),
- document.getElementsByTagName(name)
- Давталт /for,while …/
- Нөхцөл шалгах
- document.createElement(element)
  document.removeChild(*element*)
  document.appendChild(*element*)
  document.replaceChild(*new, old*)
  document.write(*text*)
- Event /onclick , ondblclick, onchange /

Нэмэлт үнэлгээ

Лаб№2 лаб дээр хийсэн хичээлий хуваарийн жагсаалтыг үүсгэж, түүнийг CV хэсэгт нэмж үүсгэж шалгуулвал +1 оноо

Нэмэлт материал

Дэлгэрэнгүйг:
https://developer.mozilla.org/en-US/docs/Web/API/Node

Event handler

Example

```
<html>
<head>
  <title>||Working with elements event||</title>
</head>
<body>
<button id='btn1'>Change color</button>
<script>

function random(number) {
  return Math.floor(Math.random() * (number+1));
}
const btn = document.getElementById('btn1').onclick =
function() {
  const rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' +
random(255) + ')';
  document.body.style.backgroundColor = rndCol;
}
</script>
<body>
```

Event Listenener

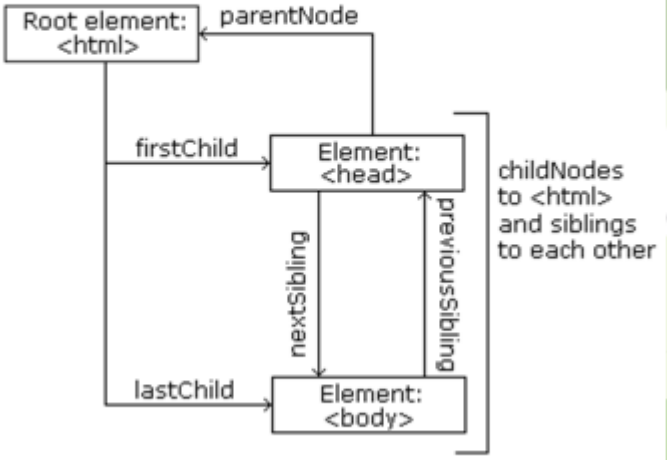
Add a simple listener
This example demonstrates how to use addEventListener()
to watch for mouse clicks on an element.

```
<head>
  <title>||Working with elements eventlistener||</title>
</head>
<body>
<table id="outside"border=1>
  <tr><td id="t1">one</td></tr>
  <tr><td id="t2">two</td></tr>
</table><html>
<script>


// Function to change the content of t2
function modifyText() {
  const t2 = document.getElementById("t2");
  if (t2.firstChild.nodeValue == "three") {
    t2.firstChild.nodeValue = "two";
  } else {
    t2.firstChild.nodeValue = "three";
  }
}

// Add event listener to table
const el = document.getElementById("outside");
el.addEventListener("click", modifyText, false);
</script>
<body>
```

Node Relationship



Navigating Between Nodes

- parentNode
- childNodes[*nodenumber*]
- firstChild
- lastChild
- nextSibling
- previousSibling

- nodeValue
- nodeName
- nodeType

22

firstChild

```
<p id="para-01">
  <span>First span</span>
</p>

<script>
  const p01 = document.getElementById("para-01");
  console.log(p01.firstChild.nodeName);
</script>
```

24

lastChild

```
const tr = document.getElementById("row1");
const corner_td = tr.lastChild;
```

25

nextSibiling

```
<div id="div-1">Here is div-1</div>
<div id="div-2">Here is div-2</div>
<br />
<output><em>Not calculated.</em></output>
<script>
let el = document.getElementById("div-1").nextSibling;
let i = 1;
let result = "Siblings of div-1:<br/>";

while (el) {
  result += `${i}. ${el.nodeName}<br/>`;
  el = el.nextSibling;
  i++;
}
const output = document.querySelector("output");
output.innerHTML = result;
</script>
```

Here is div-1
Here is div-2

Siblings of div-1:
1. #text
2. DIV
3. #text
4. BR
5. #text
6. OUTPUT
7. #text
8. SCRIPT

26

previousSibiling

```
<img id="b0" /><img id="b1" /><img id="b2" />

<SCRIPT>
document.getElementById("b1").previousSibling; // <img id="b0">
document.getElementById("b2").previousSibling.id; // "b1"
</SCRIPT>
```

27

InsertBefore

```
<div id="parentElement">
  <span id="childElement">foo bar</span>
</div>
<script>
  // Create the new node to insert
  const newNode = document.createElement("span");
  // Get a reference to the parent node
  const parentDiv = document.getElementById("childElement").parentNode;

  // Begin test case [ 1 ] : Existing childElement (all works correctly)
  let sp2 = document.getElementById("childElement");
  parentDiv.insertBefore(newNode, sp2);
  // End test case [ 1 ]
  // Begin test case [ 2 ] : childElement is of Type undefined
  sp2 = undefined; // Non-existent node of id "childElement"
  parentDiv.insertBefore(newNode, sp2); // Implicit dynamic cast to type Node
  // End test case [ 2 ]
  // Begin test case [ 3 ] : childElement is of Type "undefined" (string)
  sp2 = "undefined"; // Non-existent node of id "childElement"
  parentDiv.insertBefore(newNode, sp2); // Generates "Type Error: Invalid Argument"
  // End test case [ 3 ]
</script>
```

28

Note: There is no insertAfter() method. It can be emulated by combining the insertBefore method with Node.nextSibling.

```
parentDiv.insertBefore(sp1, sp2.nextSibling);
```

29

Element.before - insert element

```
let container = document.createElement("div");
let p = document.createElement("p");
container.appendChild(p);
let span = document.createElement("span");

p.before(span);

console.log(container.outerHTML);
// "<div><span></span><p></p></div>"
```

10

Element.before -text

```
let container = document.createElement("div");
let p = document.createElement("p");
container.appendChild(p);

p.before("Text");

console.log(container.outerHTML);
// "<div>Text<p></p></div>"
```

31

Element.before -element,text

```
let container = document.createElement("div");
let p = document.createElement("p");
container.appendChild(p);
let span = document.createElement("span");

p.before(span, "Text");

console.log(container.outerHTML);
// "<div><span></span>Text<p></p></div>"
```

32

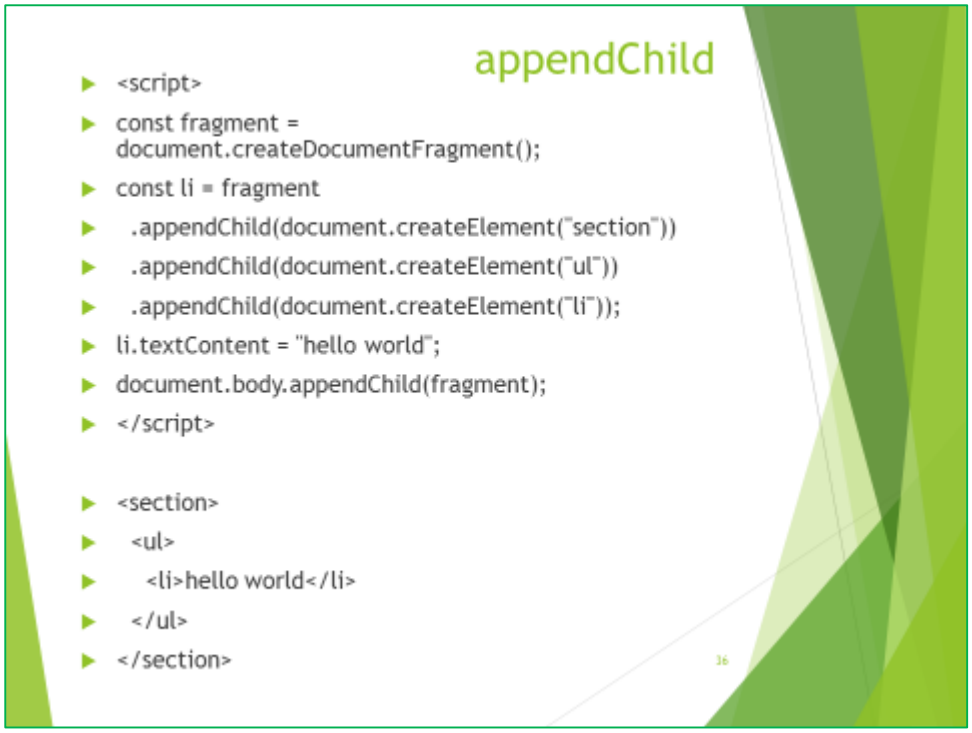
Element.after-element,text

```
let container = document.createElement("div");
let p = document.createElement("p");
container.appendChild(p);
let span = document.createElement("span");

p.after(span, "Text");

console.log(container.outerHTML);
// "<div><p></p><span></span>Text</div>"
```

35


appendChild
<script>
const fragment = document.createDocumentFragment();
const li = fragment
.appendChild(document.createElement("section"))
.appendChild(document.createElement("ul"))
.appendChild(document.createElement("li"));
li.textContent = "hello world";
document.body.appendChild(fragment);
</script>
<section>

hello world

</section>
36