

Web technológiák 1

Dokumentáció

Készítette: Palkó Patrik Dávid

Neptunkód: ZW7DOR

Egy autókkal foglalkozó oldalt készítettem amiben hozzá lehet adni a táblázathoz őket és adataikat.

Ilyen például a jármű típusa, gyártója, súly, hengerűrtartalom és lóerő.

Ha képekre rá húzzuk az egeret akkor az megnagyobbodik ha elhúzzuk onnan akkor pedig vissza megy az eredeti méretre.

A képekre rá is lehet kattintani, az egyik egy külső oldalra irányít minket a másik pedig egy lokális html linkre.

Kattintható a kontrakt és a cím is azok is külön lokális html linkre irányítanak.

Az hogy a motor mondjuk egy másfél literes át váltja nekünk az oldal ha új itemet adunk hozzá.

Lehet szűrni is a bevitt adatok közül minden attributumra.

Kép felnagyítása :

imgZoom.js

.hover funkciónak adjuk meg a transition style-t vagy vesszük el

```
$(document).ready(function(){
    $('[name='imgDiv']").hover(function() {
        $(this).addClass('transition');
    }, function() {
        $(this).removeClass('transition');
    });
});
```

azt adja meg hogy mekkora legyen az eredeti mérethez képest

```
.transition {
    transform: scale(1.2);
}
```

Ez határozza meg a nagyítási időt

```
div.zoom {  
  transition: transform .3s;  
  width: 40%;  
}
```

Motor köbcenti slider :

cmSlider.js

```
$(document).ready(function(){  
  $("#cmSlider").on('input', function(){  
    var normal = $(this).val();  
    var Liters = ($(this).val() * 1000).toFixed(0);  
    $('#cmTextNormal').val(normal);  
    $('#cmTextliters').val(Liters);  
  });  
  $("#cmTextNormal").on('input', function(){  
    var normal = $(this).val();  
    var Liters = ($(this).val() * 1000).toFixed(0);  
    $('#cmTextliters').val(Liters);  
    $('#cmSlider').val(normal)  
  })  
  $("#cmTextliters").on('input', function(){  
    var Liters = $(this).val();  
    var normal = ($(this).val() / 1000).toFixed(0);  
    $('#cmTextNormal').val(normal);  
    $('#cmSlider').val(normal)  
  })  
})
```

#cmSlider rész a behúzott értéket egyszer megjegyzi aztán egy másik változóban megszorozza 1000-el.

#cmTextNormal rész a beírt értéket egyszer megjegyzi aztán egy másik változóban pedig megszorozza 1000-el.

#cmTextliters rész a cm³-ben megadott értéket alakítja át a literre, úgy, hogy az eredeti beírt értéket megjegyzi aztán elosztja a 1000-el.

```

<label for="cmSlider" style="color :Blue" class="form-label">Liters: </label>
<input id="cmTextNormal" type="number" class="form-control" value="0" min=0 max=5 required>
<label for="cmSlider" style="color :Blue" class="form-label">Cm3: </label>
<input id="cmTextliters" type="number" class="form-control" value="0" min=1 max=5000 required>

```

Maximum és minimum van megadva a motor tulajdonságokhoz

Nem lehet több mint 5 literes/5000cm³

Motor:

Liters: Cm3:



Nevek, Email, Születési év, hozzáadása :

```

$(document).ready(function(){
    peopleList = new Array()
})

function enterName(){
    var name = $("#Name").val();
    var email = $("#email").val();
    var date = $("#Bdate").val();
    var alert = document.getElementById('alertId');
    if(!(name === "" || email === "" || date === "" ))
    {
        checkList(name);
        alert.classList.remove("alert-danger");
        alert.classList.add("alert-success");
        alert.innerHTML = ("<strong>Success!</strong> Adding was succesfull!");
        alert.classList.remove('hidden');
    }
    else{
        alert.classList.remove("alert-success");
        alert.classList.add("alert-danger");
        alert.innerHTML = ("<strong>Error!</strong> Adding was unsuccesfull! Fill all field!");
        alert.classList.remove('hidden');
    }
}

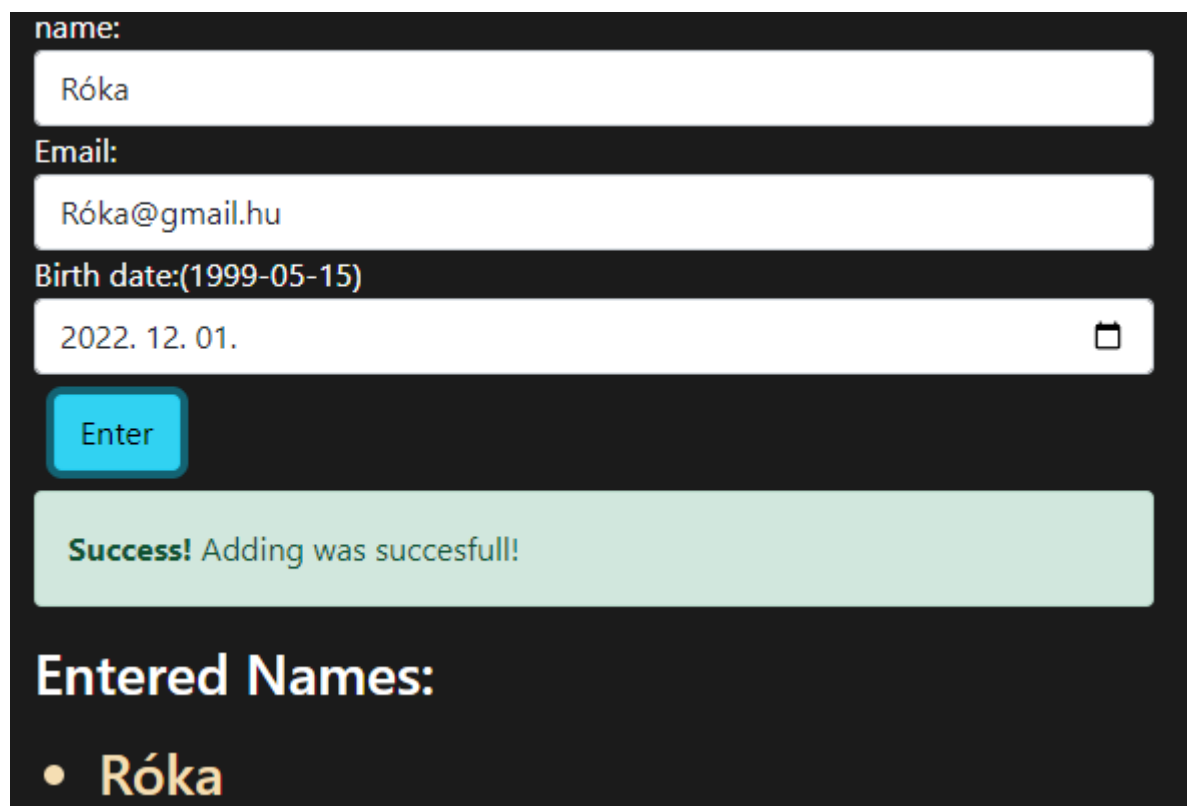
function checkList(name){
    var checkList = (element) => element == name;
    var isIn = peopleList.findIndex(checkList);
    console.log(peopleList)
    console.log(isIn)
    if(isIn == -1){
        var list = document.getElementById("enteredNames");
        var entry = document.createElement('li');
        entry.appendChild(document.createTextNode(name));
        list.appendChild(entry);
        peopleList.push(name)
    }
}

```

Létrehozunk egy peoplelist nevű listát amiben majd eltároljuk a nevet, emailt, születési évet.

Ellenőrizzük, hogy van-e olyan mező ami üres, ha van akkor hibát dobunk és nem adodik hozzá a listához, ha nincs akkor hozzá adjuk egy listához


checkListben megnézzük van-e már olyan név, ha van akkor nem adjuk hozzá ha nincs akkor igen



The screenshot shows a web form on a dark background. It has three input fields: 'name:' with the value 'Róka', 'Email:' with the value 'Róka@gmail.hu', and 'Birth date:(1999-05-15)' with the value '2022. 12. 01.' and a calendar icon. Below the fields is a blue 'Enter' button. A green success message bar reads 'Success! Adding was succesfull!'. Below this is a section titled 'Entered Names:' which contains a single bullet point '• Róka'.

name:
Róka

Email:
Róka@gmail.hu

Birth date:(1999-05-15)
2022. 12. 01. 

Enter


Success! Adding was succesfull!

Entered Names:

- Róka

name:

Email:

Birth date:(1999-05-15)


Error! Adding was unsuccesfull! Fill all field!

Entered Names:

- Róka

Table Filter :

```
$(document).ready(function(){
    $("#textFilterInput").on("input", function() {
        var value = $(this).val().toLowerCase();
        $("#tableBody tr").filter(function() {
            $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)
        });
    });
});
```

Itt azokat az értékeket adja vissza amik megegyeznek a táblában és beírt között, van kis és nagy betű érzékelés, ezt úgy érzük el, hogy igazából mindent kicsire állít a ui mögött és úgy keres.

Type
m4
M4
Supra MK4
Lada 2105

Type	Manufacturer
m4	BMW
M4	BMW

Érték hozzáadás a táblához :

```
//Type validation
$('#typeCheck').hide();
let typeError = false;
$('#type').keyup(function(){
  validateType();
});

function validateType(){
  let typeVal = $('#type').val();
  if(typeVal.length == ''){
    $('#typeCheck').show();
    typeError = false;
  }
  else {
    $('#typeCheck').hide();
    typeError = true;
  }
};
```

Ellenőrzi, hogy type kapott-e értéket, ha igen akkor typeError-t true-ra állítja, ha pedig nem akkor false-ra.

Ilyen fajta ellenőrzésből van 5 darab.

```

//Adding row to the table
$("#addItemBtn").click(function(){

    var isGood = typeError &&
                ManufacturerError &&
                weightError &&
                cmError &&
                HPErrors;

    if(isGood){
        insertTable()
        typeError = false;
        ManufacturerError = false;
        weightError = false;
        cmError = false;
        HPErrors = false;
    }else{
        if(!typeError){
            $('#typeCheck').show();
        }
        if(!ManufacturerError){
            $('#ManufacturerCheck').show();
        }
        if(!weightError){
            $('#weightCheck').show();
        }

        if(!cmError){
            $('#cmCheck').show();
        }
        if(!HPErrors){
            $('#HPCheck').show();
        }
    }
})

```

Itt ha a mindegyik változó ugyan azt az értéket foglalja magába akkor, elindítja az insertTable() metódust, ha egynél is más érték található akkor azt vissza vezeti a ui-ra

PL:

```

<div class="form-item">
<label for="type" style="color :Blue">Type: </label>
<span id="typeCheck" >**Type is missing </span>
<input id="type" type="text" class="form-control" required>
</div>

```



```

function insertTable(){
    var table = document.getElementById("tableBody");
    var row = table.insertRow(0);
    var typeRow = row.insertCell();
    var ManufacturerRow = row.insertCell();
    var weightRow = row.insertCell();
    var cmRowNormal = row.insertCell();
    var cmRowliters = row.insertCell();
    var HPRow = row.insertCell();
    typeRow.innerHTML = $("#type").val();
    ManufacturerRow.innerHTML = $("#Manufacturer").val();
    weightRow.innerHTML = $("#weight").val();
    cmRowNormal.innerHTML = $("#cmTextNormal").val();
    cmRowliters.innerHTML = $("#cmTextliters").val();
    HPRow.innerHTML = $("#HP").val();
    clearForm();
}

function clearForm(){
    $("#type").val("");
    $("#Manufacturer").val("");
    $("#weight").val("");
    $("#cmTextNormal").val(0);
    $("#cmTextliters").val(0);
    $("#cmSlider").val(0);
    $("#HP").val("");
}

```

Itt beillesztjük a táblázatba az értékeket.

Létrehoz egy táblát, azon belül egy sort, és ennek a sornak a celláiba illeszti be majd az adatokat, aminek szintén létrehozzuk a változóit.

Ezeket a változókat fel is töltjük olyan sorrendbe ahogy vannak a cellák.

Add a new item to the table



Type: **Type is missing

Manufacturer: **Manufacturer is missing

Weight: **Weight is missing

HP: **HP is missing

Motor: **Cm3 can't be zero

Liters:

Cm3:

Add Item

Close

Style :

```
.titleP{
  padding: 3%;
}
.fit {
  max-width: 99%;
  max-height: 99%;
}
.center {
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 50%;
  border-radius: 50%;
}
```

- .titleP = 3% térközt csinál a cím alá
- .fit = 99%-os nagyságot enged a képeknek
- .center = kör alakúra formázza a képet

```
footer{  
  height: auto;  
  padding: 20px;  
  text-align:center;  
  border-top: 2px solid #c0c0c0;  
}  
footer a{  
  text-decoration: none;  
  color:white;  
}  
footer p > a:hover{  
  color:#00ffcc;  
}  
footer p > a:visited{  
  color:#ff0000;  
}  
footer > p{  
  padding: 0px;  
  font-size:14px;  
}
```

footer = élőláb formázása középre és egy vonalat húz keresztbe
ami fehér színű és beállítja a betű nagyságot

footer p = a linkhez állítbe színt

```

#carTable{
  color: ■ white;
}

tbody tr:nth-child(odd){
  background-color: □ rgb(61, 61, 62);
}

tbody tr:nth-child(even){
  background-color: □ rgb(36,37,38);
}

img {
  padding: 0;
  display: block;
  margin: 0 auto;
  max-height: 100%;
  max-width: 100%;
}

html, body {
  height: 100%;
  margin: 0;
  padding: 0;
}

```

carTable = táblának állítja be betű színét

tbody (odd) = tábla hátterét állítja szürkére

tbody (even) = tábla hátterét állítja feketére

img = térközt nullára állítja és a margót is, a magasságot és a szélességet szabályozza

html = térközt nullára állítja és a margót is, és a kontent térnek a magasságát állítja be.