

Andmed ja otsing

Massiiv

Tavalises muutujas saame hoida ühte väärtust. Kui aga lehel on sarnaseid andmeid palju, siis igaüht omaette muutujas hoida on tülikas. Neid tuleb palju ning sobiva leidmine sealt ikkagi keerukas. Sarnaseid andmeid aitab koos hoida massiiv. Eesnimede loetelu saab näiteks teha nõnda:

```
var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
```

Selle peale luuakse loetelu, kus Juku poole saab pöörduda kujul `eesnimed[0]` ning Madise poole `eesnimed[3]`. Ehk siis lugemist alustatakse nullist ning viimase järjekorranumbriks on elementide arvust ühe jagu väiksem arv. Katsetamiseks juurde ka töötav koodinäide.

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
      function algus(){
        document.getElementById("vastus").innerHTML=
          "Loetelu alguses on "+eesnimed[0]+" ja "+eesnimed[1];
      }
    </script>
  </head>
  <body onload="algus()">
    <div id="vastus">

      </div>
  </body>
</html>
```

Näite tulemuseks:

Loetelu alguses on Juku ja Kati

Elementide arv

Javaskript annab õnneks mugavasti kätte massiivi elementide arvu - selleks omadus `.length` massiivimuutuja nime küljes. Sealtkaudu võimalik siis kohe arvutuse teel kätte saada massiivi viimane element. Eesnimede massiivis siis

```
eesnimed[eesnimed.length-1]
```

Näide tervikuna

```
<!doctype html>
<html>
```

```

<head>
  <title>Otsing</title>
  <meta charset="utf-8" />
  <script>
    var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
    function algus(){
      document.getElementById("vastus").innerHTML=
        "Eesnimesid kokku on "+eesnimed.length+". Viimane neist "+
          eesnimed[eesnimed.length-1]+".";
    }
  </script>
</head>
<body onload="algus()" >
  <div id="vastus">

    </div>
</body>
</html>

```

Ja tulemus

Eesnimesid kokku on 4. Viimane neist Madis.

Ülesandeid

- * Koosta massiiv inimeste pikkustega. Trüki välja esimene ja viimane pikkus.
- * Trüki välja teine ja eelviimane pikkus.

Järjekorranumbri järgi küsimine massiivist

Nagu kõiksugu muid väärtusi, nii ka massiivi elementide järjekorranumbreid saab küsida kasutaja käest. Siin siis võetakse eesnime number sisestuskastist ning kuvatakse vastav eesnimi.

```

<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
      function otsi(){
        var nimenr=parseInt(document.getElementById("kast1").value);
        document.getElementById("vastus").innerHTML=
          "Eesnimi nr. "+nimenr+" on "+eesnimed[nimenr];
      }
    </script>
  </head>
  <body>
    Mitmes eesnimi:
    <input type="text" id="kast1" />
    <input type="button" value="OK" onclick="otsi()" />
    <div id="vastus">

  </div>
</body>
</html>

```

Mitmes eesnimi:

Eesnimi nr. 2 on Katrin

Juhusliku järjekorranumbriga element

Ka juhuarv toimib massiivi elemendi järjekorranumbri leidmise juures. Lihtsalt on kasulik hoolitseda, et järjekorranumbriks tuleks täisarv - parseInt pakub sellise võimaluse.

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
      function otsi(){
        var juhuarv=Math.random(); //0 .. 0.99
        var nimenr=parseInt(eesnimed.length*juhuarv);
        document.getElementById("vastus").innerHTML=
          "Eesnimi nr. "+nimenr+" on "+eesnimed[nimenr]+
            ".<br /> Juhuarv oli "+juhuarv+";
      }
    </script>
  </head>
  <body>
    <input type="button" value="Otsi juhuslik" onclick="otsi()" />
    <div id="vastus">

    </div>
  </body>
</html>
```

Eesnimi nr. 3 on Madis.

Juhuarv oli 0.7789426770122856.

Ülesandeid

- * Koosta massiiv ütlustega. Lehe avanemisel ilmub ekraanile neist üks juhuslik ütlus.
- * Erinevalt eelmisest teatatakse kasutajale, mitme ütluse vahel tal on valida. Kasutaja sisestab soovitud ütluse järjekorranumbri ning

näeb vastaval kohal olevat ütlust.

* Koosta üks massiiv poiste eesnimedega ja teine tüdrukute eesnimedega. Lehe avanemisel valitakse juhuslik poiss ja juhuslik tüdruk ning teatatakse, et nemad lähevad esimestena tantsima.

Kordused

Massiivi andmete otsimiseks, muutmiseks ja töötlemiseks on kasulikuks abiliseks kordused. Kõigepealt väike näide korduse töötamise kohta

```
for(var i=0; i<3; i++){
    document.getElementById("vastus").innerHTML+=i+" ";
}
```

Tuumikuks siis for-tsükkel, kus iga sammuga liidetakse vastuse sisule (innerHTML) juurde tsükli loenduri i väärtus ning sinna lisatakse tühik, et arvud üksteisega kokku ei kasvaks. Tehtemärk += tähendab olemasolevale väärtusele millegi juurde lisamist, ilma, et peaks seda kohta korduvalt välja trükkima.

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      function algus(){
        for(var i=0; i<3; i++){
          document.getElementById("vastus").innerHTML+=i+" ";
        }
      }
    </script>
  </head>
  <body onload="algus()">
    <div id="vastus">

      </div>
  </body>
</html>
```

Väljund

0 1 2

Ülesandeid

- * Tutvu tsükli näitega. Trüki arvude ruudud ühest kahekümneni (suurimaks siis 20*20 ehk 400).
- * Luba kasutajal sisestada, mitmeni arvude ruute soovitakse

Tsükkel ja massiiv

Sarnaselt tsükli abil saab välja trükkida või muul moel ette võtta ka kõik samas massiivis olevad eesnimed. Lihtsalt väljatrüki või muul moel kasutuse koha peal pole enam tsükliringi loendur, vaid küsitakse massiivist vastava järjekorranumbriga element. Praegusel juhul siis `eesnimed[i]`.

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
      function algus(){
        for(var i=0; i<eesnimed.length; i++){
          document.getElementById("vastus").innerHTML+=eesnimed[i]+" ";
        }
      }
    </script>
  </head>
  <body onload="algus()">
    <div id="vastus">

    </div>
  </body>
</html>
```

Väljundiks:

Juku Kati Katrin Madis

Eelmises näites olid eesnimede vahel tühikud. Nõnda tekstina saab aga ka kõiki HTMLi korraldusi anda ja nendega funktsiooni töö tulemust mõjutada. Kui eesnimede vahele kirjutada reavahetusmärgend `
`, siis kasutaja näeb iga nime omaette real.

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
      function algus(){
        for(var i=0; i<eesnimed.length; i++){
          document.getElementById("vastus").innerHTML+=
            eesnimed[i]+"<br />";
        }
      }
    </script>
  </head>
  <body onload="algus()">
    <div id="vastus">

    </div>
  </body>
```



```

        t+="

```

- Juku
- Kati
- Katrin
- Madis

Ülesandeid

- * Tutvu massiivi andmete trükkimise näidetega, tee näited läbi.
- * Kirjuta iga eesnime välja kaks korda
- * Trüki eesnimed ekraanile tagurpidises järjekorras
- * Asenda `` ``'ga ning vaata, kuidas mummud nimede ees asenduvad järjekorranumbritega

Teksti otsimise käsk

Javaskripti levinud tööks veebilehel on andmete seast sobivate välja otsimine - kui nimekiri on pikk, siis aitab programmikood sisestatud lõigu järgi sobivad välja valida. Selle jaoks leidub tekstifunktsioon nimega `indexOf` - mis teatab, kas ja kust otsitav tekst leiti. Tekstis on programmeerimiskeelte tarvis tähed nummerdatud, kusjuures Javaskripti puhul hakkab nummerdamine nullist. Ehk siis eesnimes "Mart" on täht "M" järjekorranumbriga 0 ja täht "a" järjekorranumbriga 1. Ning kui küsin et `"Mart".indexOf("a")`, siis saan vastuseks 1. Kui aga küsin mõnd olematut tähte, näiteks `"Mart".indexOf("u")`, siis saan vastuseks -1, sest tähte "u" Mardi nimes pole. Järgnevalt väike demo, kuidas seda käsklust veebilehel kasutada.

Ühte tekstikasti sisestab kasutaja uuritava lause, teise kasti otsitava teksti. Käsklus

```
var koht=lause.indexOf(otsitav);
```

püüab muutujasse nimega `koht` `indexOf`-käskluse tulemust, kus otsiti lausest otsitavat. Kui lõik leiti, nädatakse tähe järjekorranumbrit mitmendast lõik hakkab. Kui mitte, siis vastavaks koodiks olevat arvu -1.

```

<!doctype html>
<html>
  <head>

```

```

<title>Otsing</title>
  <meta charset="utf-8" />
  <script>
    function otsi(){
      var lause=document.getElementById("kast1").value;
      var otsitav=document.getElementById("kast2").value;
      var koht=lause.indexOf(otsitav);
      document.getElementById("vastus").innerHTML=
        "Otsitav kohal "+koht;
    }
  </script>
</head>
<body>
  Uuritav lause:
  <input type="text" id="kast1" value="Muna ja kana" />
  Otsitav tekst:
  <input type="text" id="kast2" value="ja" />
  <input type="button" value="Otsi" onclick="otsi()" />
  <div id="vastus">

  </div>
</body>
</html>

```

Kui lauseks "Muna ja kana" ning tekstiks "ja", siis vastuseks 5, sest tähe "j" järjekorranumber on 5 ("M"-tähe number on 0).

Uuritav lause: Otsitav tekst:

Otsitav kohal 5

Samas lauses aga kukk puudub ning vastuseks on miinus üks.

Uuritav lause: Otsitav tekst:

Otsitav kohal -1

Saadud arve on võimalik kasutajale koos selgitustega näidata. Tingimuslause abil tasub kindlaks teha, et kui leitud koha väärtus on -1, siis võib kasutajale selgesõnaliselt teada anda, et lauses puudub otsitav lõik. Muul juhul aga öeldakse, kust otsitav leiti.

```

<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      function otsi(){
        var lause=document.getElementById("kast1").value;
        var otsitav=document.getElementById("kast2").value;
        var koht=lause.indexOf(otsitav);
        if(koht===-1){
          document.getElementById("vastus").innerHTML=
            "Lauses puudub "+otsitav
        } else {
          document.getElementById("vastus").innerHTML=
            "Otsitav kohal "+koht;
        }
      }
    </script>
  </head>
  <body>
    Uuritav lause:
    <input type="text" id="kast1" value="Muna ja kana" />
    Otsitav tekst:
    <input type="text" id="kast2" value="kukk" />
    <input type="button" value="Otsi" onclick="otsi()" />
    <div id="vastus">

    </div>
  </body>
</html>

```



```

    }
  }
</script>
</head>
<body>
  Uuritav lause:
  <input type="text" id="kast1" value="Muna ja kana" />
  Otsitav tekst:
  <input type="text" id="kast2" value="ja" />
  <input type="button" value="Otsi" onclick="otsi()" />
  <div id="vastus">

  </div>
</body>
</html>

```

Samade lausete puhul siis saab nüüd koos selgitustega vasted.

Uuritav lause: Otsitav tekst:

Otsitav kohal 5

Uuritav lause: Otsitav tekst:

Lauses puudub kukk

Otsing massiivist

Edasi piisab lõigu otsimise näide kokku panna varasema massiivi elementide näitamise näitega. Iga eesnime puhul kontrollitakse, et kas otsitav lõik selles eesnimes leidub. Kui jah, siis lisatakse vastav eesnimi näidatavate loetellu. Kui ei leidu, siis ei lisata ning seda ekraanil ei näe. Vastav lõik eraldi välja tooduna:

```

if(eesnimed[i].indexOf(otsitav)!=-1){
  t+="- " + eesnimed[i] + "</li>";
}

```

Kui muidu koostati loetelu nõnda, et pandi kõik massiivis olevad eesnimed loetellu

```

var t="<ul>";
for(var i=0; i<eesnimed.length; i++){
  t+="- " + eesnimed[i] + "</li>";
}
t+="

```

siis nüüd lisandus for-tsükli sisse if-tingimuslause, mille abil vaid tingimusele vastavad eesnimed näidatakse:

```

var t="<ul>";
for(var i=0; i<eesnimed.length; i++){
  if(eesnimed[i].indexOf(otsitav)!=-1){
    t+="- " + eesnimed[i] + "</li>";
  }
}

```

```
t+="</ul>";
```

Töötav näide tervikuna:

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var eesnimed=new Array("Juku", "Kati", "Katrin", "Madis");
      function otsi(){
        var otsitav=document.getElementById("kast2").value;
        var t="<ul>";
        for(var i=0; i<eesnimed.length; i++){
          if(eesnimed[i].indexOf(otsitav)!=-1){
            t+="<li>"+eesnimed[i]+"</li>";
          }
        }
        t+="</ul>";
        document.getElementById("vastus").innerHTML=t;
      }
    </script>
  </head>
  <body>
    Otsitav tekst:
    <input type="text" id="kast2" />
    <input type="button" value="Otsi" onclick="otsi()" />
    <div id="vastus"></div>
  </body>
</html>
```

Otsitav tekst:

- Kati
- Katrin

Suur- ja väiketähed

Javaskript eristab suur- ja väiketähti. Ehk siis K ning k on tema jaoks sama erinevad tähed kui a ja b. Kasutaja aga otsimise ajal ei pruugi jälgida ega eristada, kumma suurusega tähe järgi ta parajasti otsida soovib. Kasutajat häirivate üllatuste vältimiseks saab otsingulehe panna käituma nõnda, et sobivad eesnimed leitakse nii suurte kui väiksete tähtede järgi otsides. Üheks mooduseks on nii kasutaja sisestatud tekst kui ka võrreldav nimi muuta võrdlemise ajaks väiketähtedeks. Sellisel juhul pole vahet, milline oli tähekõrgus enne - võrreldakse ikkagi lõpuks ühesuguseid sümboleid. Kasutaja sisestatud tekst tehakse väikseks kohe andmete sisselugemisel.

```
var otsitav=
  document.getElementById("kast2").value.toLowerCase();
```

Massiivist eesnimesid võttes aga tehakse nad väikseks ükshaaval iga võrdlemise tarbeks.

```
if(eesnimed[i].toLowerCase().indexOf(otsitav)!=-1){
    t+="- "+eesnimed[i]+"</li>";
}

```

Õigemini massiivis olev nimi ise jääb endiselt nõnda nagu ta on, aga võrdlemiseks tehakse koopia mille toLowerCase() väljastab.

Otsingurakendus tekstina:

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var eesnimed=new Array("Juku", "Kati", "Katriin", "Madis");
      function otsi(){
        var otsitav=
          document.getElementById("kast2").value.toLowerCase();
        var t="<ul>";
        for(var i=0; i<eesnimed.length; i++){
          if(eesnimed[i].toLowerCase().indexOf(otsitav)!=-1){
            t+="- "+eesnimed[i]+"</li>";
          }
        }
        t+="

```

Otsitav tekst:

- Juku
- Kati
- Katriin
- Madis

Otsitav tekst:

- Juku
- Kati
- Katriin

Ülesanded

- * Tutvu kasutaja otsitavate nimede näitamisega
- * Näita vaid neid nimesid, mis algavad otsitava tekstiga (indexOf(otsitav)==0)
- * Näita kõik leitud nimed välja väiketähtedena

Otsing kirjetest

Päriselus ette tulevate andmete juures kipub sageli olema rohkem kui üks tunnus. Raamatul on pealkiri ja autor ja ilmumisaeg. Pildil mõõtmed, tegemisaeg, koht ja vahel kirjeldus. Ning mõnikord neid tunnuseid ehk välju võib andmetel päris palju olla. Kui otsitakse korvpallureid, siis pigem kahesaja ringis olev vastav väärtus pikkuse juures võib sobiva vihje anda. Kui aga kehakaal samale arvule läheneb, siis pole vastaval tegelasel ehk suuremate võistluste juures kuigi palju peale hakata. Õnneks võimaldab ka Javaskript tunnuseid nõnda eraldi jagada ja pärast vaadata, et millist neist just parajasti kätte saada tahetakse. Järgnevalt näide, kus laste andmeteks on nende eesnimed ja sünniajad. Otsitakse eesnime järgi, aga näidatakse mõlemat tunnust. Eelnevalt väike seletus andmestiku kirjapaneku kohta.

Luuakse uus massiiv nimega lapsed. Tühi massiiv näeks välja

```
var lapsed=[];
```

Siia aga on võimalik kohe väärtused ka sisse panna. Ühe lapse andmed kannatab kirja panna kujul

```
{"eesnimi":"Juku", "synniaeg": "13.06"}
```

Igal lapsel siis kaks tunnust: eesnimi ja synniaeg. Teispool koolonit nende väärtused konkreetse lapse puhul. Kui laste andmeid mitu järjest ja komadega eraldatult on, siis tulebki tulemuseks laste massiiv.

```
var lapsed=[
  {"eesnimi":"Juku", "synniaeg": "13.06"},
  {"eesnimi":"Kati", "synniaeg": "11.06"},
  {"eesnimi":"Mati", "synniaeg": "25.08"},
  {"eesnimi":"Madis", "synniaeg": "09.02"}
];
```

Sellist andmete kirjapanekut nimetatakse JSON-iks (JavaScript Object Notation). Sellisel kujul programmile etteantud andmeid saab mugavasti koodis kasutada. Muutuja `lapsed` on täiesti tavaline massiiv, `lapsed[0]` näiteks massiivi esimene laps. Kui aga soovitakse tema eesnime, siis tuleb küsida `lapsed[0].eesnimi`. Soovides tsükli ehk korduslause abil järgemööda kõikide laste poole pöörduda, tuleb kantsulgudes oleva arvu asemele panna tsüklimuutuja. Nõnda siis kontrollitakse `i`-nda lapse puhul, et kas tema väiketähtedena vaadatud eesnimes on otsitav tekst.

```
if(lapsed[i].eesnimi.toLowerCase().indexOf(otsitav)!=-1){}
```

Loetellu näitamiseks aga lisatakse eesnimi koos sünniajaga.

```
t+="<li>"+lapsed[i].eesnimi+ " "+
  lapsed[i].synniaeg+"</li>";
```

Edasi juba otsingu kood tervikuna.

```
<!doctype html>
<html>
  <head>
    <title>Otsing</title>
    <meta charset="utf-8" />
    <script>
      var lapsed=[
        {"eesnimi":"Juku", "synniaeg": "13.06"},
```

```

        {"eesnimi":"Kati", "synniaeg": "11.06"},
        {"eesnimi":"Mati", "synniaeg": "25.08"},
        {"eesnimi":"Madis", "synniaeg": "09.02"}
    ];
    function otsi(){
        var otsitav=
            document.getElementById("kast2").value.toLowerCase();
        var t="<ul>";
        for(var i=0; i<lapsed.length; i++){
            if(lapsed[i].eesnimi.toLowerCase().indexOf(otsitav)!=-1){
                t+="<li>"+lapsed[i].eesnimi+ " "+
                    lapsed[i].synniaeg+"</li>";
            }
        }
        t+="</ul>";
        document.getElementById("vastus").innerHTML=t;
    }
</script>
</head>
<body onload="otsi()">
    Otsitav tekst:
    <input type="text" id="kast2" onkeyup="otsi()" />
    <div id="vastus"></div>
</body>
</html>

```

Otsitav tekst:

- Juku 13.06
- Kati 11.06

Ülesandeid

- * Tutvu eesnimede ja sünnipäevade vaatamise näitega.
- * Kuva sünnipäev eespool, eesnimi tagapool
- * Koostage väike sõnaraamat, kus näidatakse sisestatud sõna vasteid.

Sõnaraamat.

Sõnu koos vastetega vähemasti ühe õpiku õppetüki jagu.
 Kasutaja sisestatud tähtede peale näidatakse välja sobivad sõnad ja vasted, leht on mõnevõrra kujundatud (näiteks vasted üle ühe erinevat värvi)
 Viisakas on näiteks kasutada mõnd olemasolevat mobiililehe malli (template)

Pikkuste joonis.

Massiivis on inimeste nimed ja pikkused.
 Joonisele kuvatakse vaid nende inimeste nimed, kes on etteantud pikkuste vahemikus. Nime kõrval näidatakse pikkus vastava kõrgusega tulbana.

Asukohad parklas.

Massiivis on kirjas autode numbrid ning nende x- ja y-koordinaadid parklas. Sisestades autonumbri või selle osa näeb otsingule vastavaid autosid parklas (koos õige asukohaga).