

## Tehtäväsetti 3.

Joonas Kangaskoski, N3303

### Tehtävä 1.

```
.js T0301.js > ...
1  function montaTervehdysta(lkm) {
2      return console.log('Terve maailma! '.repeat(lkm));
3  }
4  montaTervehdysta(10);
5  |
```

### Tehtävä 2.

```
.js T0302.js > ...
1  function montaTervehdysta(lkm, tulostettava) {
2      tulostettava =
3      |   typeof tulostettava !== 'undefined' ? tulostettava : 'Terve maailma!' + ' ';
4      return console.log(tulostettava.repeat(lkm));
5  }
6  montaTervehdysta(3);
7  |
```

### Tehtävä 3.

```
.js T0303.js > ...
1  function onkoJaollinenKolmella(luku) {
2      if (luku % 3 === 0) {
3          return true;
4      } else {
5          return false;
6      }
7  }
8
9  let tulostus = onkoJaollinenKolmella(3);
10 console.log(tulostus);
11 console.log(onkoJaollinenKolmella(4));
12 |
```

## Tehtävä 4.

```
.js T0304.js > ...
1  function onkoKarkausVuosi(vuosi) {
2      if ((vuosi % 4 == 0 && vuosi % 100 != 0) || vuosi % 400 == 0) {
3          return true;
4      } else {
5          return false;
6      }
7  }
8  console.log(onkoKarkausVuosi(1998));
9  console.log(onkoKarkausVuosi(2012));
10 console.log(onkoKarkausVuosi(2001));
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0304.js
false
true
false
```

## tehtävä 5.

```
.js T0305.js > pyydaLuku
1  const readlineSync = require('readline-sync');
2
3  function pyydaLuku() {
4      const minLuku = 0;
5      const maxLuku = 31;
6      let syote;
7      const teksti = 'Anna luku väliltä 1-30, niin näytän sen sinulle!';
8      console.log(teksti);
9      do {
10         syote = parseInt(readlineSync.question('Anna luku: '));
11
12         if (syote <= minLuku || syote >= maxLuku) {
13             console.log('luvun piti olla väliltä 1-30');
14         } else {
15             return console.log(`Syöttämäsi luku oli ${syote}`);
16         }
17     } while (syote < 1 || syote > 30);
18 }
19 pyydaLuku();
20
```

## Tehtävä 6.

```
T0306.js > ...
1  function kertoTaulu(x, y) {
2      let taulu = [];
3      for (let i = x; i <= x; i++) {
4          for (let j = 1; j <= y; j++) {
5              taulu.push(i * j);
6          }
7      }
8      console.log(taulu);
9  }
10 kertoTaulu(3, 5);
11
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0306.js
[ 3, 6, 9, 12, 15 ]
```

## Tehtävä 7.

```
T0307.js > laskeMontaYhteenTaulukko
1  let luvut = [10, 20, 30];
2  function laskeMontaYhteenTaulukko(a) {
3      // tarkistetaan että parametri on annettu kutsussa, ellei niin 0
4      a = typeof a !== 'undefined' ? a : 0;
5      // tarkistetaan että parametri on taulukko/object js:ssä, ellei niin 0
6      a = typeof a === 'object' ? a : 0;
7      // lasketaan taulukon arvot yhteen
8      let sum = 0;
9      for (let i = 0; i < a.length; i++) {
10         sum += a[i];
11     }
12     // palautetaan summa.
13     return sum;
14 }
15 console.log(laskeMontaYhteenTaulukko(luvut));
16
```

## Tehtävä 8.

```
JS T0308.js > laskeMontaYhteen
1  function laskeMontaYhteen(...nums) {
2      let sum = 0;
3
4      // |...nums tekee yksittäisistä parametreista taulukon.
5      // foreachilla käydään taulukko läpi
6      // ja lisätään jokainen annettu numero summaan
7      nums.forEach(function (x) {
8          sum += x;
9      });
10     return sum;
11 }
12 console.log(laskeMontaYhteen(10, 20, 30, 40));
13
```

## Tehtävä 9.

```
JS T0309.js > ...
1  const foo = function (x, cb) {
2      const tulos = [];
3      for (let i = 1; i <= x; i++) {
4          tulos.push(i);
5      }
6      if (typeof cb === 'function') {
7          cb(tulos);
8      }
9  };
10
11 foo(5, function (foonTulos) {
12     console.log(foonTulos);
13 });
14
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0309.js
[ 1, 2, 3, 4, 5 ]
```

## Tehtävä 10.

```
.js T0310.js > keskiarvoMerkkijono
1  const keskiarvoMerkkijono = (y, ...numerot) => {
2    let summa = 0;
3    // lasketaan taulukon numeroiden summa
4    numerot.forEach((x) => {
5      summa += x;
6    });
7    // lasketaan keskiarvo summa jaettuna taulukopituudella
8    let ka = summa / numerot.length;
9    // palautetaan tuloste
10   return console.log(`Keskiarvo (${y}) on ${ka}`);
11 };
12 keskiarvoMerkkijono('ikä', 10, 20, 30, 40);
13
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0310.js
Keskiarvo (ikä) on 25
```

## Tehtävä 11.

```
T0311.js > ...
1 // haetietojostain materiaalin functiot muutettu arrow funktioiksi, tässä ensimmäinen
2 const haeTietoaJostain = (haku, cb) => {
3   console.log('Palveluun saatiin yhteys ja suoritetaan komento: ' + haku);
4   let tulos = [];
5
6   // tässä toinen arrow
7   setTimeout(() => {
8     if (haku === 'Hae kaikki') {
9       tulos = [
10        {
11          nimi: 'Pekka',
12          kengannumero: 43,
13        },
14        {
15          nimi: 'Anneli',
16          kengannumero: 37,
17        },
18      ];
19    }
20
21    if (typeof cb === 'function') {
22      cb(tulos);
23    }
24  }, 3000);
25 };
26 // täällä viimeinen ja siihen lisätty muutos, että järjestää
27 // tuloksen kengännumeron mukaan nousevaksi.
28 haeTietoaJostain('Hae kaikki', (tulos) => {
29   console.log(tulos.sort((a, b) => a.kengannumero - b.kengannumero));
30 });
31
32 console.log('Suoritus on täällä');
```

## Tehtävä 12.

```
T0312.js > ...
1 const luvut = [50, -20, 100, 10, -100, 10, 30];
2
3 // suodatellaan luvut taulukkoa
4 const suodatetut = luvut.filter((luku) => luku < 0 || (luku > 24 && luku < 51));
5 console.log(suodatetut);
6
```

### Tehtävä 13.

```
T0313.js > ...
1  const nimet = ['Pekka Ruuskanen', 'Anneli Tehhunen', 'Ilari Itäpuro'];
2
3  // käydään taulukon alkiot läpi, otetaan jokaisesta ensimmäinen kirjain,
4  // lisätään piste ja etsitään välilyönnin jälkeinen kirjain
5  const nimikirjaimet = nimet.map(
6    (nimi) => nimi.charAt(0) + '.' + nimi.charAt(nimi.indexOf(' ') + 1)
7  );
8  console.log(nimikirjaimet);
9
```

### Tehtävä 14.

```
T0314.js > [e] eiLapaissyt
1  const opiskelijat = [
2    { nimi: 'Pekka', pisteet: 24, lapi: true },
3    { nimi: 'Anneli', pisteet: 30, lapi: true },
4    { nimi: 'Einari', pisteet: 13, lapi: false },
5    { nimi: 'Inkeri', pisteet: 26, lapi: true },
6    { nimi: 'Seppo', pisteet: 9, lapi: false },
7    { nimi: 'Aulikki', pisteet: 4, lapi: false },
8  ];
9
10 const lapaisseet3 = opiskelijat;
11 // luodaan muuttuja
12 let eiLapaissyt = opiskelijat
13   // filteröidään vanhasta taulukosta opiskelijat jotka eivät läpäisseet
14   // | ja pisteet alle 10
15   .filter((op) => op.lapi == false && op.pisteet < 10)
16   // järjestetään laskevaan järjestykseen
17   .sort((a, b) => {
18     return b.pisteet - a.pisteet;
19   });
20 // logataan ulos uusi muuttuja eli uusi taulukko
21 console.log(eiLapaissyt);
22 // vanha taulukko edelleen entisensä
23 console.log(lapaisseet3);
24
```

### Tehtävä 15.

```
JS T0315.js > ...
1  const ostokset = [
2    { tuote: 'Maitoa', maara: '5 litraa' },
3    { tuote: 'Kahvi', maara: '2 pakettia' },
4    { tuote: 'Banaaneita', maara: '1 kilo' },
5    { tuote: 'Leipä', maara: '1 kpl' },
6  ];
7  // käydään taulukko läpi for of lauseella ja tulostetaan
8  // tuotteiden arvot
9  for (let a of ostokset) {
10   console.log(`${a.tuote}`);
11 }
12
```



### Tehtävä 16.

```
JS T0316.js > ...
1  const ostokset = [
2    { tuote: 'Maitoa', maara: '5 litraa' },
3    { tuote: 'Kahvi', maara: '2 pakettia' },
4    { tuote: 'Banaaneita', maara: '1 kilo' },
5    { tuote: 'Leipä', maara: '1 kpl' },
6  ];
7  // luodaan uusi muuttuja, otetaan vain tuotteiden nimet talteen
8  const tuotteet = ostokset.map((a) => {
9    return a.tuote;
10 });
11 // logataan uusi taulukko
12 console.log(tuotteet);
13
```



### Tehtävä 17.

- Funktio järjestää taulukon pienemmistä isoimpaan sekä tulostaa vain 2 ensimmäistä taulukon alkia.
- Funktio mutatoi annetun taulukon ja sen voi tarkistaa esim. `console.log(luvut)`, mutatoonnin välttämiseksi tulee luoda uusi muuttuja funktion sisään ja kopioida taulukko spread metodilla.

```
T0317.js >  pienimmat2 >  x
1  function pienimmat2(taulukko, lkm) {
2      // luodaan uusi muuttuja funktion käyttöön, jolloin aiemmin
3      // annettu taulukko ei mutatoitu
4      let x = [...taulukko];
5      return x.sort((a, b) => a - b).splice(0, lkm);
6  }
7
8  const luvut = [5, 10, 1, 8];
9  const uusi = pienimmat2(luvut, 2);
10 console.log(uusi);
11 // tarkistetaan että vanha on ennallaan.
12 console.log(luvut);
13
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0317.js
[ 1, 5 ]
[ 5, 10, 1, 8 ]
```

## Tehtävä 18.

```
T0318.js > jaljelleJaaneet
1 function pienimmat3(taulukko, lkm) {
2   // ei mutatoida alkuperäistä taulukkoa
3   const taulukko2 = [...taulukko];
4   // luodaan uusi muuttuja
5   const obj = {
6     // leikataan aiemmalla komennolla 2 ensimmäistä indexia ja
7     // muihin sitten lopputaulukko
8     pienimmat: taulukko2.sort((a, b) => a - b).splice(0, lkm),
9     muut: taulukko2,
10  };
11  return obj;
12 }
13 // laitoin testimielessä vähän lisää lukuja.
14 const luvut = [5, 10, 1, 8, 12, 11];
15 const uusi = pienimmat3(luvut, 2);
16 console.log(uusi);
17 //const jaljelleJaaneet = uusi['muut'];
18 const [...jaljelleJaaneet] = uusi['muut'];
19 console.log(jaljelleJaaneet);
20
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0318.js
{ pienimmat: [ 1, 5 ], muut: [ 8, 10, 11, 12 ] }
[ 8, 10, 11, 12 ]
```

## Tehtävä 19.

```
T0319.js > ...
1 const mina = {
2   etunimi: 'Joonas',
3   sukunimi: 'Kangaskoski',
4   kengannumero: 44,
5   harrastukset: ['Lenkkeily', 'Kuntosali', 'Koodailu'],
6   kotipaikka: { kunta: 'Jyväskylä', maakunta: 'Keski-suomi' },
7 };
8 console.log(
9   `Minä olen ${mina.etunimi} ja minulla on kengät numeroltaan ${mina.kengannumero}.
10   Kotipaikkani on ${mina.kotipaikka.kunta}
11   ja minulla on ${mina.harrastukset.length} harrastusta`
12 );
13
```

## Tehtävä 20.

```
T0320.js > esittely > pitka
1  const mina = {
2    etunimi: 'Joonas',
3    sukunimi: 'Kangaskoski',
4    kengannumero: 44,
5    harrastukset: ['Lenkkeily', 'Kuntosali', 'Koodailu'],
6    kotipaikka: { kunta: 'Jyväskylä', maakunta: 'Keski-suomi' },
7  };
8
9  function esittely(hlo) {
10   // taas kopioidaan
11   const x = { ...hlo };
12   // palautetaan objekti jossa lyhyt, keskipitka, pitka
13   return {
14     lyhyt: `Henkilö ${x.etunimi.charAt(0)}${x.sukunimi}, kotipaikka ${
15       hlo.kotipaikka.kunta
16     }`,
17     keskipitka: `Henkilö ${x.etunimi} ${x.sukunimi}, jonka kotipaikka on ${x.kotipaikka.kunta} maakunnassa`,
18     pitka: `Henkilö ${x.etunimi} ${x.sukunimi}, jonka kotipaikka on ${x.kotipaikka.kunta} maakunnassa`,
19   };
20 }
21 // Luodaan mie
22 const mie = esittely(mina);
23 // testataan tulostukset eri versioin, bueno!
24 console.log(mie.keskipitka);
25
```

Pitkät on litaniat niin ei oikein järkevää kuvaa saa... tulostuksista näkyy.

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset\node T0320.js
Henkilö J.Kangaskoski, kotipaikka Jyväskylä
Henkilö Joonas Kangaskoski, jonka kotipaikka on Jyväskylä maakunnassa Keski-suomi. Hänellä on kengät numeroltaan 44
Henkilö Joonas Kangaskoski, jonka kotipaikka on Jyväskylä maakunnassa Keski-suomi. Hänellä on kengät numeroltaan 44 ja 3 harrastusta: Lenkkeily,Kuntosali,Koodailu
```

## Tehtävä 21.

```
T0321.js > mina > juttele
1  const mina = {
2    etunimi: 'Joonas',
3    sukunimi: 'Kangaskoski',
4    kengannumero: 44,
5    harrastukset: ['Lenkkeily', 'Kuntosali', 'Koodailu'],
6    kotipaikka: { kunta: 'Jyväskylä', maakunta: 'Keski-suomi' },
7    nuku: () => {
8      console.log('ZZZZzzzzzz!');
9    },
10   juttele: (lause) => {
11     console.log(lause);
12   },
13 };
14 mina.nuku();
15 mina.juttele('Onko kaikki hyvin?');
```

## Tehtävä 22.

```
T0322.js > mina
1  const mina = {
2    etunimi: 'Joonas',
3    sukunimi: 'Kangaskoski',
4    kengannumero: 44,
5    harrastukset: ['Lenkkeily', 'Kuntosali', 'Koodailu'],
6    kotipaikka: { kunta: 'Jyväskylä', maakunta: 'Keski-suomi' },
7    nuku: () => {
8      console.log('ZZZZzzzzzz!');
9    },
10   juttele: (lause) => {
11     console.log(lause);
12   },
13   // luodaan esittele metodi
14   esittele: function (lyhyt) {
15     if (lyhyt === true) {
16       console.log(`Olen ${this.etunimi} ${this.sukunimi}`);
17     } else {
18       console.log(
19         `Olen ${this.etunimi} ${this.sukunimi}, kotipaikkani on ${this.kotipaikka.kunta} ja kengännumeroni on ${this.kengannumero}`
20       );
21     }
22   },
23 };
24 mina.esittele(true);
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0322.js
Olen Joonas Kangaskoski
```

## Tehtävä 23.

```
T0323.js > ...
1  // luodaan auto-konstruktori
2  function Auto(merkki) {
3    this.merkki = merkki;
4  }
5  // luodaan prototyypit
6  Auto.prototype.tulostaMerkki = function () {
7    return this.merkki;
8  };
9  Auto.prototype.aja = function () {
10   setInterval(() => {
11     console.log(this.merkki + ' ajaa eteenpäin ja kuuluu vain wroom wroom');
12   }, 1000);
13 };
14 // luodaan uudet autot ja kokeillaan niiden toiminnat
15 let rr = new Auto('Rolls Royce');
16 console.log(rr.tulostaMerkki());
17 rr.aja();
18 let lada = new Auto('Lada');
19 console.log(lada.tulostaMerkki());
20 lada.aja();
```

```

D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0323.js
Rolls Royce
Lada
Rolls Royce ajaa eteenpäin ja kuuluu vain wroom wroom
Lada ajaa eteenpäin ja kuuluu vain wroom wroom
Rolls Royce ajaa eteenpäin ja kuuluu vain wroom wroom
Lada ajaa eteenpäin ja kuuluu vain wroom wroom
Rolls Royce ajaa eteenpäin ja kuuluu vain wroom wroom

```

## Tehtävä 24.

```

T0324.js > ...
1  const Auto = {
2    tulostaMerkki: function () {
3      console.log(this.merkki);
4    },
5    aja: function () {
6      setInterval(() => {
7          console.log(this.merkki + ' ajaa eteenpäin ja kuuluu vain wroom wroom');
8      }, 1000);
9    },
10 };
11 // Luodaan kysit object creatella
12 const rr = Object.create(Auto);
13 // merkki on asetettu rr:ään muttei "Autoon"
14 rr.merkki = 'Rolls Royce';
15
16 const lada = Object.create(Auto);
17 lada.merkki = 'Lada';
18 // ajellaan tulosteet läpi
19 lada.aja();
20 lada.tulostaMerkki();
21 rr.tulostaMerkki();
22 rr.aja();

```

```

D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0324.js
Lada
Rolls Royce
Lada ajaa eteenpäin ja kuuluu vain wroom wroom
Rolls Royce ajaa eteenpäin ja kuuluu vain wroom wroom
Lada ajaa eteenpäin ja kuuluu vain wroom wroom
Rolls Royce ajaa eteenpäin ja kuuluu vain wroom wroom

```

## Tehtävä 25.

```
const Henkilo = {
  esittaydy: function () {
    console.log(
      `Nimeni on ${this.sukunimi}, ${this.etunimi} ${this.sukunimi} `
    );
  },
  // kerroIka funktio
  kerroIka: function () {
    console.log(`Olen ${this.ika}-vuotias`);
  },
};

const ihmiset = [];
for (let i = 0; i < 5; i++) {
  // luodaan object.creatella uusia tyyppejä, lisänä myös ika
  const tyypit = Object.create(Henkilo);
  (tyypit.etunimi = etunimet[arvoLuku(0, etunimet.length - 1)]),
  (tyypit.sukunimi = sukunimet[arvoLuku(0, sukunimet.length - 1)]),
  (tyypit.ika = new Date().getFullYear() - arvoLuku(1970, 2000));
  ihmiset.push(tyypit);
}
console.log(ihmiset);
// kaikki esittäytyy
ihmiset.forEach((a) => console.log(a.esittaydy()));
//ensimmäisen ja viimeisen ikäesittely
ihmiset[0].kerroIka();
ihmiset[ihmiset.length - 1].kerroIka();
```

## Tehtävä 26.

```
JS T0326.js > kertotaulu
1 function kertotaulu(x, y) {
2   const kertsu = [];
3   // tarkistetaan onko syötteet ohjeiden mukaiset
4   try {
5     if (x < 0 || y < 0) {
6       // jos ei niin heitetään error
7       throw 'Argumenttien pitää olla positiivisia lukuja';
8       // jos on niin antaa laulaa vaan
9     } else {
10      for (let i = 1; i <= y; i++) {
11        kertsu.push(i * x);
12      }
13      return kertsu;
14    }
15    // kiinni otetun errorin tulostus
16  } catch (error) {
17    console.log('Error: ' + error);
18    // tarkistus info et laitoinko muka oikeasti väärän luvun
19    // tai mitkäs luvut laitoinkaan.
20  } finally {
21    console.log(`Syöttämäsi luvut oli ${x} ja ${y}`);
22  }
23 }
24
25 console.log(kertotaulu(-1, 6));
26
```

```
D:\syksy 2021\Ohjelmointi1\tehtavapalautukset>node T0326.js
Error: Argumenttien pitää olla positiivisia lukuja
Syöttämäsi luvut oli -1 ja 6
```

## Tehtävä 27.

```
T0305.js > pyydaLuku
2
3 function pyydaLuku() {
4   const minLuku = 0;
5   const maxLuku = 31;
6   let syote;
7   const teksti = 'Anna luku väliltä 1-30, niin näytän sen sinulle!';
8   console.log(teksti);
9   do {
10    syote = parseInt(readlineSync.question('Anna luku: '));
11
12    // tarkistetaan vielä että syöte on numero eikä esim. stringi, rullatus pyörii niin kauan
13    // kunnes syote on oikeanlainen eli numero väliltä 1-30
14    try {
15      if (syote <= minLuku || syote >= maxLuku) {
16        throw 'HOX!HOX! Syötteen tulee olla numero väliltä 1-30';
17      } else if (syote !== 'number' || isNaN(syote)) {
18        throw 'HOX!HOX! Syötteen tulee olla numero väliltä 1-30';
19      } else {
20        return console.log(`Syöttämäsi luku oli ${syote}`);
21      }
22    } catch (error) {
23      console.log('Error: ' + error);
24    }
25  } while (syote < 1 || syote > 30 || syote !== 'number');
26 }
27 pyydaLuku();
28
```