

rajapinnat: kaikki tehtävät

1. Kopioi alla oleva koodi tiedostoon `callback_example.js` ja suorita se.

```
setTimeout(doSomething,2000);
```

```
function doSomething(){  
    console.log("Demonstrating the callbacks");  
}  
  
console.log("The application is started");
```

2. Tee ohjelma käyttäen anonyymiä funktiota (anonymous function)
3. Tee ohjelma käyttäen nuolifunktiota (arrow function)

```
JS callback > ...
1  setTimeout(doSomething,2000);
2
3  function doSomething(){
4      console.log("Demonstrating the callbacks");
5  }
6  console.log("The application is started");
7
8  setTimeout(function(){
9      console.log("anonymous callbacks");
10 },3000);
11
12
13 console.log("The application is started");
14 setTimeout(()=>{
15     console.log("Arrow callbaks");
16 },4000);
17
18 |
19 console.log("The application is started");

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

The application is started
Demonstrating the callbacks
PS C:\rajapinnat> node callback
The application is started
Demonstrating the callbacks
PS C:\rajapinnat> node callback
The application is started
The application is started
The application is started
Demonstrating the callbacks
anonymous callbacks
Arrow callbaks
PS C:\rajapinnat> 
```

Assosiatiivinen taulukko

Luo assosiatiivinen array nimeltään bookArray, jossa on samat kentät kuin book-
taulussa kts. <https://peatutor.com/databases/library.php>

Tulosta konsolille mikä luomasi taulun "tietotyyppi" on (käytä JavaScriptin typeof-
funktioita)

Tulosta konsolille koko bookArrayn sisältö

Tulosta konsolille bookArrayn ensimmäinen rivi

Tulosta konsolille ensimmäisen kirjan nimi

Tulosta konsolille arrayn rivien määrä

Tulosta konsolille kaikkien kirjojen nimet

```

const bookArray = {
  1: { id_book: 1, name: "To Kill a Mockingbird", author: "Harper Lee", isbn: "111-1111111111" },
  2: { id_book: 2, name: "1984", author: "George Orwell", isbn: "222-2222222222" },
  3: { id_book: 3, name: "Pride and Prejudice", author: "Jane Austen", isbn: "333-3333333333" },
  4: { id_book: 4, name: "The Great Gatsby", author: "F. Scott Fitzgerald", isbn: "444-4444444444" },
  5: { id_book: 5, name: "Moby Dick", author: "Herman Melville", isbn: "555-5555555555" },
  6: { id_book: 6, name: "War and Peace", author: "Leo Tolstoy", isbn: "666-6666666666" },
  7: { id_book: 7, name: "The Catcher in the Rye", author: "J.D. Salinger", isbn: "777-7777777777" },
  8: { id_book: 8, name: "The Hobbit", author: "J.R.R. Tolkien", isbn: "888-8888888888" },
  9: { id_book: 9, name: "Crime and Punishment", author: "Fyodor Dostoevsky", isbn: "999-9999999999" },
  10: { id_book: 10, name: "The Odyssey", author: "Homer", isbn: "101-1010101010" },
  11: { id_book: 11, name: "The Brothers Karamazov", author: "Fyodor Dostoevsky", isbn: "111-1111111112" },
  12: { id_book: 12, name: "Brave New World", author: "Aldous Huxley", isbn: "121-1212121212" },
  13: { id_book: 13, name: "Jane Eyre", author: "Charlotte Brontë", isbn: "131-1313131313" },
  14: { id_book: 14, name: "Wuthering Heights", author: "Emily Brontë", isbn: "141-1414141414" },
  15: { id_book: 15, name: "The Divine Comedy", author: "Dante Alighieri", isbn: "151-1515151515" }
};

console.log(typeof bookArray);

console.log(bookArray);

console.log(bookArray[1]);

console.log(bookArray[1].name);
🔗
console.log([Object.keys(bookArray).length]);

for (const key in bookArray) {
  if (bookArray.hasOwnProperty(key)) {
    console.log(bookArray[key].name);
  }
}

```

Olio

```

const bookArray = {
  1: { id_book: 1, name: "To Kill a Mockingbird", author: "Harper Lee", isbn: "111-111111111" },
  2: { id_book: 2, name: "1984", author: "George Orwell", isbn: "222-222222222" },
  3: { id_book: 3, name: "Pride and Prejudice", author: "Jane Austen", isbn: "333-333333333" },
  4: { id_book: 4, name: "The Great Gatsby", author: "F. Scott Fitzgerald", isbn: "444-444444444" },
  5: { id_book: 5, name: "Moby Dick", author: "Herman Melville", isbn: "555-555555555" },
  6: { id_book: 6, name: "War and Peace", author: "Leo Tolstoy", isbn: "666-666666666" },
  7: { id_book: 7, name: "The Catcher in the Rye", author: "J.D. Salinger", isbn: "777-777777777" },
  8: { id_book: 8, name: "The Hobbit", author: "J.R.R. Tolkien", isbn: "888-888888888" },
  9: { id_book: 9, name: "Crime and Punishment", author: "Fyodor Dostoevsky", isbn: "999-999999999" },
  10: { id_book: 10, name: "The Odyssey", author: "Homer", isbn: "101-101010101" },
  11: { id_book: 11, name: "The Brothers Karamazov", author: "Fyodor Dostoevsky", isbn: "111-111111112" },
  12: { id_book: 12, name: "Brave New World", author: "Aldous Huxley", isbn: "121-121212121" },
  13: { id_book: 13, name: "Jane Eyre", author: "Charlotte Brontë", isbn: "131-131313131" },
  14: { id_book: 14, name: "Wuthering Heights", author: "Emily Brontë", isbn: "141-141414141" },
  15: { id_book: 15, name: "The Divine Comedy", author: "Dante Alighieri", isbn: "151-151515151" }
};

function getAllBooks() {
  console.log(bookArray);
}

function getOneBook(id) {
  if (bookArray[id]) {
    console.log(bookArray[id]);
  } else {
    console.log(`Book with id ${id} not found.`);
  }
}

function addBook(id_book, name, author, isbn) {
  if (!bookArray[id_book]) {
    bookArray[id_book] = { id_book, name, author, isbn };
    console.log(`Book with id ${id_book} added.`);
  } else {
    console.log(`Book with id ${id_book} already exists.`);
  }
}

console.log(getAllBooks());

console.log(getOneBook(1));

console.log(addBook(16, "The Picture of Dorian Gray", "Oscar Wilde", "161-161616161"));

```

Javascript tehtävä

```

function whichNumberIsHigher()
{
    var number1 = 5;
    var number2 = 10;
    if (number1 > number2){
        console.log("Number 1 is higher than number 2");
    } else {
        console.log("Number 2 is higher than number 1");
    }
}

console.log(whichNumberIsHigher());

function isPalindrome(str) {
    let j = str.length - 1;
    for (let i = 0; i < str.length; i++){
        if(str[i] !== str[j]){
            return false;
        }
        j--;
    }
    return true;
}

let str1 = "saippuakivikauppias";
let str2 = "nitin";
let str3 = "moi";

console.log(isPalindrome(str1));
console.log(isPalindrome(str2));
console.log(isPalindrome(str3));

```

Tehtävä 19

```

20 app.get('/example',
21   function(request, response){
22     response.send('Olen esimerkki');
23     console.log('Olen esimerkki');
24   }
25 );
26
27 app.use(
28   function(req,res,seuraava){
29     console.log('Yleinen väliohjelmisto nimeltä');
30     seuraava();
31   }
32 );|
33
34 app.get('/example/:name',
35   function(request, response){
36     response.send('hello ' + request.params.name);
37     console.log('hello ' + request.params.name);
38   });
39 app.get('/example/:name/:lastname',
40   function(request, response){
41     response.send('hello ' + request.params.name+" "+ request.params.lastname);
42     console.log('hello ' + request.params.name.age+" "+ request.params.lastname);
43   });
44
45 app.post('/',
46   function(request,sponse){
47     sponse.send(request.body);
48     console.log(request.body);
49   }
50 );

```

POST ⌵ http://localhost:3000 Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Code Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

<input checked="" type="checkbox"/> firstname	mikko
<input checked="" type="checkbox"/> lastname	laurila

Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 41 ms Size: 277 B ...

Pretty Raw Preview **JSON** 🔍

```

2  "firstname": "mikko",
3  "lastname": "laurila"
4  ]

```

