

JS 4 React

Dio JavaScripta koji se često koristi u Reactu i dobro je znati

Sadržaj

Osnove	1
Tipovi podataka.....	1
Izrazi	1
Operatori.....	1
Varijable i konstante	2
Uvjetni izrazi.....	2
Petlje	2
This	2
Ternary Operator	2
Destructuring	2
Spread operator	2
Array methods	3
Arrow functions	3
Promises.....	3
Fetch API	3
Async / Await	4
ES modules i import / Export	4

Osnove

Tipovi podataka

Jednostavni (Primitive): number, string, boolean, null, undefined, ...

Složeni (Reference): nizovi (Array), objekti (Objects), funkcije, kolekcije (Map, Set), datum

Izrazi

Izraz je sintaksno važeći dio koda koji rezultira sa nekom vrijednosti.

Operatori

Pridruživanje: =

Aritmetički: +, -, *, /, ++, --, ...

Relacijski: ==, ===, <, >, <=, >=, !=, !==

Logički: && (Logički I), || (Logičko ILI), ! (Logičko Ne)

Varijable i konstante

Deklaracija i inicijalizacija: var x = 10;

Ključne riječi let i const

Uvjetni izrazi

If...else

Switch...case

Petlje

for, for...of (niz, string), for...in (objekt)

while

do...while

This

U globalnom kontekstu je window objekt

Unutar funkcije se odnosi na objekt koji poziva funkciju

Pomoću funkcije bind() možemo promijeniti this kontekst unutar funkcije

<funkcija>.bind(<objekt>) – unutar <funkcija> this ima kontekst <objekt>

Konstruktor funkcije automatski kreiraju vezu između this i objekta kojeg kreiraju

Callback funkcije se izvršavaju u zasebnom kontekstu

Arrow funkcije se izvršavaju unutar konteksta u kojem su napisane

Ternary Operator

condition ? 'true' : 'false'

Destructuring

Izdvađa elemente nizova ili objekata u zasebne varijable

```
const objects = ['table', 'iPhone', 'apple']
```

```
const [furniture, mobile, fruit] = objects
```

```
// furniture = 'table'...
```

Spread operator

Iterirajući tip podatka (niz) rastavlja na individualne elemente: ...array

```
const [person, setPerson] = useState({  
  name: ''
```

```
});
```

```
setPerson([...person, {name: "Ivan"}]);
```

Array methods

`push(<element>)`: dodaje element na kraj niza

`pop()`: uklanja zadnji element niza

`slice(<pocetni_index>, <zavrzni_index>)`: vraća dio niza kao novi niz

`filter(<funkcija>)`: vraća novi niz sa elementima za koje funkcija daje true

```
const words = ['limit', 'elite', 'exuberant', 'destruction'];
```

```
const result = words.filter((word) => word.length > 6);
```

```
// result = ["exuberant", "destruction"]
```

`map(<funkcija>)`: vraća novi niz sa elementima koji nastaju primjenom funkcije na elemente niza

```
const array1 = [1, 4, 9, 16];
```

```
const map1 = array1.map((x) => x * 2);
```

```
// map1 = [2, 8, 18, 32]
```

Arrow functions

Alternativa standardnim function izrazima

Sintaksne opcije:

```
() => expression
```

```
param => expression
```

```
(param) => expression
```

```
(param1, paramN) => expression
```

```
() => {  
  statements  
}
```

```
param => {  
  statements  
}
```

```
(param1, paramN) => {  
  statements  
}
```

Promises

Koristi se za implementaciju asinhronih opreacija

```
const prom = new Promise((resolve, reject) => {  
  let broj = Math.floor(Math.random()*10)+1;  
  broj < 6 ? resolve(broj) : reject(broj);  
});
```

```
prom  
  .then((br) => console.log(br + ': 1-5'))  
  .catch((br) => console.log(br + ': 6-10'));
```

Fetch API

Iz browsera kreira asinhroni HTTP request prema web serveru

```
fetch('url') // Dohvaca resurs sa url-a
.then(response => response.json()); // Cita response stream
.then(data => console.log(data)); // Ispisuje podatke
```

Async / Await

Omogućava zaustavljanje izvršenja programa dok Promise ne postane raspoloživ

```
const gen = new Promise(resolve => {
  let broj = Math.floor(Math.random()*10)+1;
  resolve(broj);
});
```

```
async function noviBroj() {
  const genBr = await gen;
  console.log(genBr);
}
```

```
noviBroj();
```

ES modules i import / Export

Omogućava razdvajanje koda na zasebne datoteke

Default export: jedna vrijednost po modulu

```
export default <vrijednost>
import <vrijednost> [ as <novo_ime_vrijednosti> ] from <modul>
```

Named export: više vrijednosti po modulu

```
export <vrijednost>
import { <vrijednost> [ as <novo_ime_vrijednosti> ] } from <modul>
```