JavaScript W3school

JS Output

Mozemo na vise nacina da prikazemo podatke:

- Pisanje unutura HTML elementa innerHTML
  - Pristupili smo elementu preko id-a i napisali smo neki text unutar <p> elementa

```
<!DOCTYPE >
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My First Paragraf</p>
    <p id="demo"></p>

    <script>
      document.getElementById("demo").innerHTML = 5 + 5;
    </script>
  </body>
</html>
```

- Pisanje unutar HTML izlaza document.write()
  - Kada korisitmo document.write sve nakon toga sto se tice HTML-a nece se prikazati

```
<!DOCTYPE >
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My First Paragraf</p>
    <p id="demo">test</p>

    <script>
      document.getElementById("demo").write("10");
      document.write("test");
    </script>
  </body>
</html>

My First Web Page
```

```
My First Paragraf
test


<!DOCTYPE >
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My First Paragraf</p>
    <p id="demo">test</p>

    <script>
      document.write("test");
    document.getElementById("demo").write("10");
    </script>
  </body>
</html>

My First Web Page
My First Paragraf
test
test
```

- Jos jedan primer kada ce sve da se izbrise jer pirkazujemo document.write

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<button type="button" onclick="document.write(5 + 6)">Try it</buttor

</body>
</html>

My First Web Page
My first paragraph.
Try it
```

```
Nakon:
11
```

- Pisanje unutar alert kutije window.alert()
  - window objekat je globalni objekat. Sto znaci ne moramo da koristimo window nego mzemo samo da kazem aleart, jer promenljive,metode i tako to po difoltu pripadaju objektu window

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
alert(5 + 6);
</script>

</body>
</html>
```

- Pisanje unutar bewoser konzole console.log()
  - Ovo obicno koristimo za debugging svrhe

```
<!DOCTYPE html>
<html>
<body>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

JavaScript Variables

Mozemo da deklarisemo promenljive na 4 nacina:

- Automacki

```
x=5
```

- var

```
var x = 5
```

- let

```
let x = 5
```

- const

```
const x = 5
```

Prvo se koristio var a zatim su ubacili let i const, var treba da se koristi samo za starije browsere

Kada koristiti var, let, const?

- Uvek trebamo da deklarisemo vrednost
- Uvek korisitmo const ako ne zelimo da se vrednost menja
- Uvek koristimo const ako ne zelimo da se tip promenljive menja
- Koristimo let ako ne mozemo da koristimo const
- Koristimo var kada moramo da suportamo stare browsere

JavaScript LET

- Promenljive deklarisane sa let imaju block scope
- Promneljive deklarisane sa let moraju da budu deklarisane pre upotrebe

```
console.log(x); // Ovo će izazvati grešku jer x nije deklarirana prije
let x = 5;
```

- Promenljive deklarisane sa let ne mogu da budu ponovo deklarisane u istom scop-u

Promenljive deklarisane sa var uvek imaju globalni scope, ne mogu da imaju block scope

Promenljive deklarisane sa var mogu biti rediklarisane. Redeklarisanje promenljive u bloku promenice vrednost i izvan bloka jer je to globalna promenljiva

	Scope	Redeclare	Reassign	Hoisted	Binds this
var	No	Yes	Yes	Yes	Yes
let	Yes	No	Yes	No	No
const	Yes	No	No	No	No

Var moze da bude hoisted sto znaci mozemo da korisitmo promenljivu negde pri vrhu i da je deklarisemo ispod.

```
carName = "Volvo";
var carName;
```

JavaScript CONST

- ne moze da bude rediklarisana
- Ne mozemo da joj promenimo vrednost
- Ima block scope
- Moramo da joj damo vrednost tokom deklarisanja

Kada koristiti const?

Kada deklarisemo :

- new array
- new object
- new function
- new RegExp

Const object and array

Const u ovom slucaju ne definise const vrednost vec const referencu na vrednost.

Sto znaci:

- Ne mozemo:
  - Rediklarisati const vrednost
  - Rediklarisati const niz
  - Rediklarisati const objekat
- Mozemo:
  - Da promenimo vrednost elemenata u const niz
  - Da promenimo vrednost const objekta

```
// You can create a constant array:
const cars = ["Saab", "Volvo", "BMW"];

// You can change an element:
cars[0] = "Toyota";

// You can add an element:
cars.push("Audi");



const cars = ["Saab", "Volvo", "BMW"];

cars = ["Toyota", "Volvo", "Audi"];    // ERROR
```

```
// You can create a const object:
const car = {type:"Fiat", model:"500", color:"white"};

// You can change a property:
car.color = "red";

// You can add a property:
car.owner = "Johnson";
```

```
const car = {type:"Fiat", model:"500", color:"white"};

car = {type:"Volvo", model:"EX60", color:"red"};    // ERROR
```