



JavaScript

Uvod



Pregled

- Uvod
- Dodavanje JSa na stranicu
- Console i Web API
- JS Statements
 - Vrijednosti (fiksne i promjenljive)
 - Varijable, načini deklarisanja
 - Operatori
 - Izrazi (expressions)
 - Keywords
 - Komentari
- Tipovi podataka
 - Primitivni tipovi
 - Nizovi i objekti, deklaracija
- Zadaci
- Pitanja



Obnavljanje

- Test i domaći 2
- Šta smo do sada naučili
- Pogledajmo ponovo Network i Console u pretraživaču
- Kratko podsjećanje na klijent/server komunikaciju



Uvod



- JavaScript je programski jezik koji je danas vrlo zastupljen u Webu
- Nekad se koristio samo za frontend
 - I dalje jedna od glavnih uloga JS je sređivanje frontenda (Angular, React, pure JS)
 - Danas se koristi i za Backend (NodeJS framework)
 - Koristi se i pri izradi Mobilnih aplikacija (npr. React Native)
 - Koristi se i pri izradi Desktop aplikacija (npr. Electron)
 - Danas je vrlo dobro poznavati JS
- Kreiran 1995 godine od strane [Brendana Eich](#)
- JavaScript nema baš nikakve povezanosti sa Java jezikom
 - Zašto onda JavaScript? U periodu kad je izašao JS programski jezik Java je bio vrlo popularan, i zbog marketinške kampanje, dobija ovaj naziv



Dodavanje JSa na stranicu

- Slično kao što imamo dodavanje CSS fajlova u HTML fajl, tu mogućnost imamo i sa JS fajlovima
- Dva se osnovna načina dodavanja JS skripti na stranicu
 - Eksterno
 - Interno
- Kod eksternog dodavanja radimo sledeće
 - Kreiramo element **script** sa atributom **src** koji predstavlja putanju do JS fajla
 - `<script src="path/file.js"></script>`
 - U htmlu fajlu ovaj dio se dodaje u **head** element ili na samom **dnu body elementa**
- Kod internog dodavanja radimo sledeće
 - Kreiramo element script koji sadrži JS kod
 - `<script>JS CODE</script>`
- Za puno više detalja o načinu dodavanja JS skripti na stranicu, pogledati [članak](#)
- Kasnije ćemo da govorimo o **defer** i **async** atributima za script element



Dodavanje JSa na stranicu, nastavak

- Koji način onda koristiti i gdje dodati script element?
- Najčešće se koristi ekstreno dodavanja skripti i to na dnu body elementa
 - Jedan od glavnih razloga za to je što se skripta tek pokreće nakon što pretraživač učitava i prikaže sav HTML i CSS sadržaj
 - Ovo je posebno bitno ako sa JS skriptom pokušavate da nađete neki HTML element koji se još nije učitao/prikazao na stranici. Slična logika i za CSS (ako pokušate da izijenite JSom stil prije nego se primijeni CSS)

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS 193X</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    ... sadržaj stranice...
    <script src="filename.js"></script>
  </body>
</html>
```



Console i Web API (in context of Browser)

- **Web API** (u kontekstu pretraživača) je omogućen od strane pretraživača i koristimo ga za komunikaciju sa njim za rješavanje raznih front-end problema
 - Često ćemo se vraćati na Web API, koristićemo razne Web API-je kao što je manipulacija HTML elementima, odlaganje izvršavanja određenog dijela koda, itd.
 - Ako ste koristili do sada `setTimeout` evo odakle to stiže [pogledajte](#)
 - Kompletna [lista](#)
- **Console** je dio Web APIa, često se koristi, a najviše se primjenjuje u slučaju testiranja određenih dijelova koda.
- Isprobajmo jedan primjer
- Osim **`console.log`**, postoje i drugi načini za prikaz rezultata u konzoli pretraživača kao što je `console.error(args)`, `console.warn(args)`, itd. ali nama je za sada dovoljan **`console.log(args)`**
- Za više detalja o Console Web APIu pogledati [članak](#)



JS Statements

- JS program sastoji se od liste programskih “izjava” (statements)
- JS statements sastoje se od vrijednosti, operacija, izraza (expressions), ključnih riječi (keywords) i komentara
- JS Statements razdvajaju se sa ;
 - Kod JS ; nije obavezno ali jeste preporučljivo koristiti

```
var x, y;
```

```
// How to declare variables
```

```
x = 5; y = 6;
```

```
// How to assign values
```

```
z = x + y;
```

```
// How to compute values
```




JS vrijednosti

- JS koristi dva tipa vrijednosti i to fiksirane i varijabilne.
- Za fiksiranje (**literali**):
 - Za brojeve pišu se sa ili bez decimalne tačke
 - Za Stringove pišu se sa jednostrukim ili dvostrukim navodnicima
- Za varijabilne (**varijable**)
 - Kao i u ostalim programskim jezicima, varijable se koriste za smještanje vrijednosti
 - Imenovati varijable što intuitivnije onome šta je njihov sadržaj i ne koristiti previše opšte nazive (npr. data), najčešće se koristi **camelCase**
 - Postoje tri načina deklarisanja varijabli i to: **var**, **let** i **const**
 - Inicijalizacija se obavlja preko znaka jednako =
 - JS je **dynamically typed** programski jezik, što znači da se pri samom izvršavanju koda određuje tip promjenljive

```
var x;  
  
x = 6;
```



Promjenljive, načini deklarisanja `const`/`let`/`var`

- Kao što smo već pomenuli postoje tri načina deklarisanja promjenljivih i to sa:
 - `const`, `let` i `var`
- Da biste bolje razumjeli koja je razlika između ova 3 načina deklarisanja promjenljivih, potrebno je da prođemo određeni dio gradiva, posebno lekciju Scope koju ćemo raditi kad budemo pominjali i funkcije.
- Promjenljive su se nekad deklarisle sa `var`, danas se to rijetko koristi ali i dalje skroz eliminisan jer ima pojedinih prednosti u odnosu na `const` i `let` (npr. Sa `var` možete da odradite novu deklaraciju iste promjenljive, dok sa `const` i `let` ne možete)
- Za sad je dovoljno da znate da promjenljive koje su deklarisanje sa `const` ne mogu biti ponovo inicijalizovane (**napomena**: ovo važi samo kod primitivnih tipova podataka o kojima će biti više riječi u nastavku), a njihov opseg važenja je samo blok u kome su definisane (za razumijevanje ovog dijela lekcija Scope). Za `let` isto važi kao i za `const` osim što je njihova vrijednost promjenljiva.



JS operatori

- Za sad pomenimo samo osnovne operatore:
 - Aritmetički
 - `+`, `-`, `*`, `/`, `mod`, `++`, `--`
 - Logički
 - `&&`, `||`, `!`
 - Operator dodjelivanja vrijednosti
 - `=`, `+=`, `-=`, `*=`, `/=`, `%=`
 - Operatori poređenja
 - `==`, `===`, `>`, `<`, `>=`, `<=`, `!=`, `!==`, `?`
 - Type operator
 - `Typeof`, `instanceof`
 - Itd.
- U dodatku (imate pdf imate gotovo kompletnu listu operatora kao i težinu i redosled izvršavanja svakog od njih)
- O većini operatora ćemo govoriti pojedinačno u nastavku



JS izrazi (Expressions)

- Sastoje se od kombinacije vrijednosti, varijabli i operatora, čije računanje daje vrijednost.
- Primjeri izraza
 - `5 * 10`
 - `x * 10`
 - `"John" + " " + "Doe"`
- I o izrazima ćemo puno detaljnije govoriti kako budemo prelazili još gradiva



JS ključne riječi (keywords)

- Koriste se za identifikaciju akcija koje trebaju biti izvršenje kao što je **var** za deklarisanje promjenljive
- Isto tako za **for**, **if**, **while**, **break**, itd.
- Listu ključnih riječi možete pogledati na [linku](#)
- Nazivi promjenljivih i funkcija ne mogu koristiti ključne riječi za identifikatore (npr. promjenljivu ne možete nazvati for)



JS komentari

- Ne razlikuje se od većina programskih jezika
 - `//` za jednolinijske komentare
 - `/**` za multiline komentare
 - Kad vodite dokumentaciju o kodu postoji specifično definisan način komentarisanja koda
 - Korisno pročitati ovaj [članak](#)



Tipovi podataka

- Primitivni tipovi podataka
 - Undefined
 - Default vrijednost za neinicijalizovane promjenljive
 - Razlika između undefined i undeclared !
 - String
 - Tekst
 - Number
 - Float, Int
 - Boolean
 - True ili False
 - Null
 - Symbol
- Složeni tipovi podataka
 - Object
 - U JSu je sve Object, ali to ćemo vidjeti kasnije kako i zašto
 - *Function*



Tipovi podataka, primjer

```
typeof foo;           // "undefined"  
typeof "foo";         // "string"  
typeof 123;           // "number"  
typeof true;          // "boolean"  
typeof { a: 1 };      // "object"  
typeof function() { alert(e); }; // "function"
```

- Šta se dešava sa null ?



Primitivni tipovi podataka, Quiz 1

```
1 var foo;  
2 typeof foo;           // ???  
3  
4 var bar = typeof bar;  
5 bar;                  // ???  
6 typeof bar;           // ???  
7  
8 typeof typeof 2;      // ???
```

Primitivni tipovi podataka, specijalne vrijednosti

- Lista specijalnih vrijednosti
 - NaN
 - Infinity, -Infinity
 - Null
 - Undefined (void)
 - +0, -0

```
1 var a = "a" / 2;  
2  
3 a;           // NaN  
4 typeof a;    // "number"  
5 isNaN(a);    // true  
6  
7 isNaN("foo"); // true! WAT!?
```

```
1 var foo = 0 / -3;  
2 foo === -0;    // true ... yay?  
3 foo === 0;     // true ... doh!  
4 0 === -0;      // true ... grr!  
5 (0/-3) === (0/3); // true ... sigh  
6  
7 foo;           // 0 ... WAT!?
```

```
1 var baz = 2;
2 typeof baz;           // ???
3 var baz;
4 typeof baz;           // ???
5 baz = null;
6 typeof baz;           // ???
7
8 baz = "baz" * 3;
9 baz;                  // ???
10 typeof baz;          // ???
11
12 baz = 1 / 0;
13 baz;                 // ???
14 typeof baz;          // ???
```



Native Types, functions i constructors

- Lista native constructors tipova:
 - `new String()`
 - `new Number()`
 - `new Boolean()`
 - `new Function()`
 - `new Object()`
 - `new Array()`
 - `new RegExp()`
 - `new Date()`
 - `new Error()`
- Bez **new** keyworda dobijamo listu native funkcija
- Testirajmo šta dobijamo sa **new String()** i **String()** npr.



Native types

```
1 var foo = new String("foo");
2 foo; // ???
3 typeof foo; // "object"
4 foo instanceof String; // true
5 foo instanceof string; // false; ref error!
6
7 foo = String("foo");
8 typeof foo; // "string"
9
10 foo = new Number(37);
11 typeof foo; // "object"
```



Nizovi i objekti, deklaracija

- Za nizove se koristi Array literal []
- Za objekte se koristi Object literal {}

```
1 var foo;  
2  
3 foo = new Array(1,2,3); // don't!  
4  
5 foo = [1,2,3];          // do!  
6  
7 foo = new Object();     // don't!  
8 foo.a = 1;  
9 foo.b = 2;  
10 foo.c = 3;  
11  
12 foo = { a:1, b:2, c:3 };// do!
```



Zadaci

- Kreirati niz svih polaznika kursa. Za svakog se čuva ime i prezime, email adresa i datum rođenja. U konzoli pretraživača prikazati niz. Koristiti eksterno učitavanje skripte u HTML dokumentu. HTML dokument treba da sadrži samo osnovnu strukturu, s tim što body element treba da sadrži skriptu koja sadrži kod za štampanje i kreiranje niza



Pitanja