

# JavaScript - uvod

Adaptirana verzija slajdova.  
Zahvaljujemo kolegici Viktoriji Kirst

## Plan rada

### **U ovoj lekciji:**

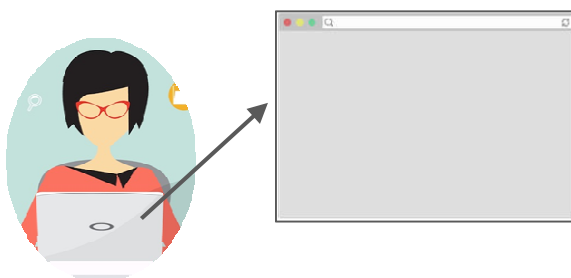
- Uvod u jezik JavaScript
  - Što je JavaScript?
  - Pregled mogućnosti jezika

# Kako pristupamo veb-stranici

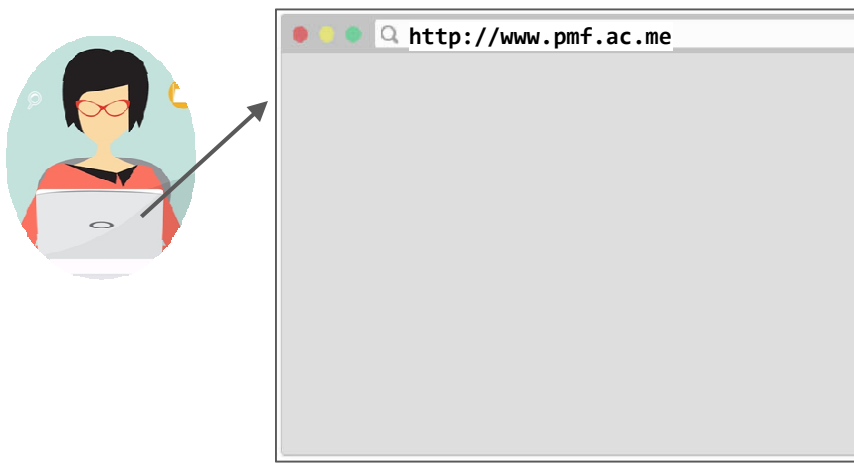
Vi na vašem laptop računar

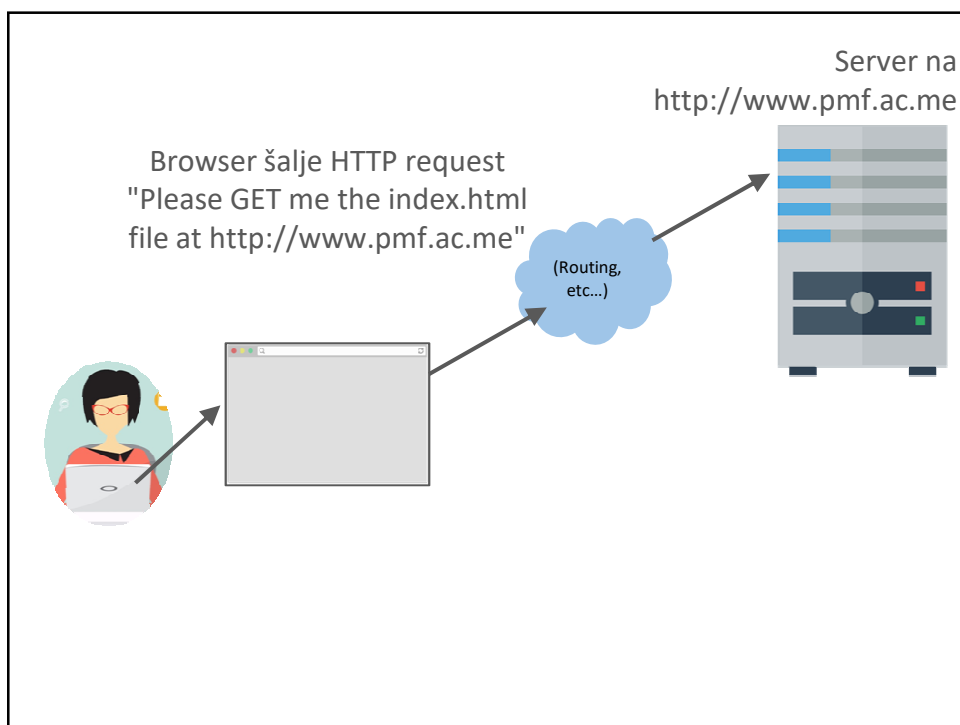
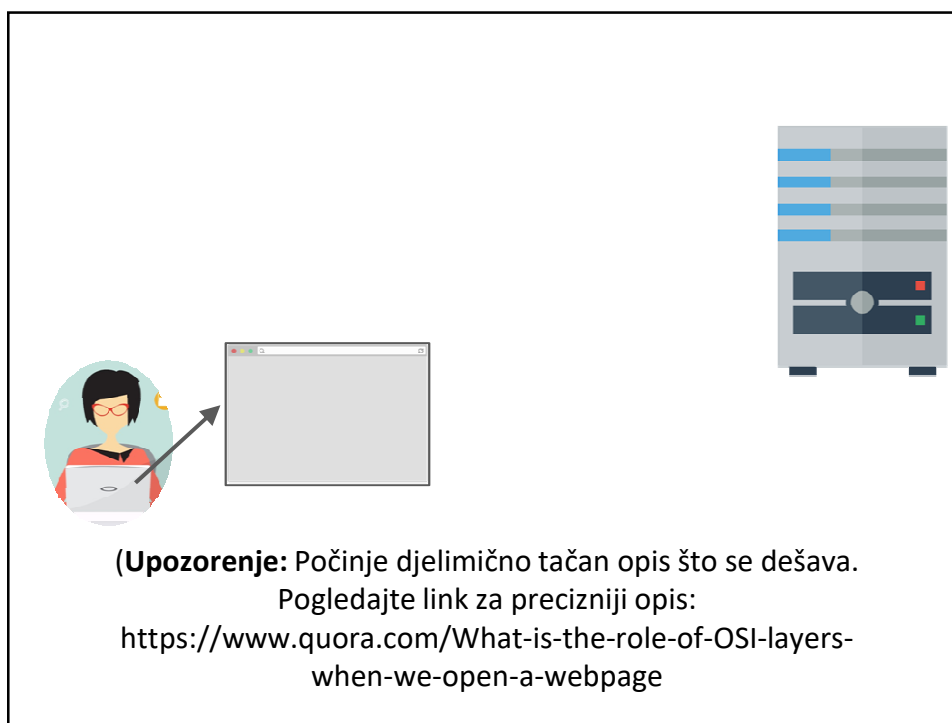


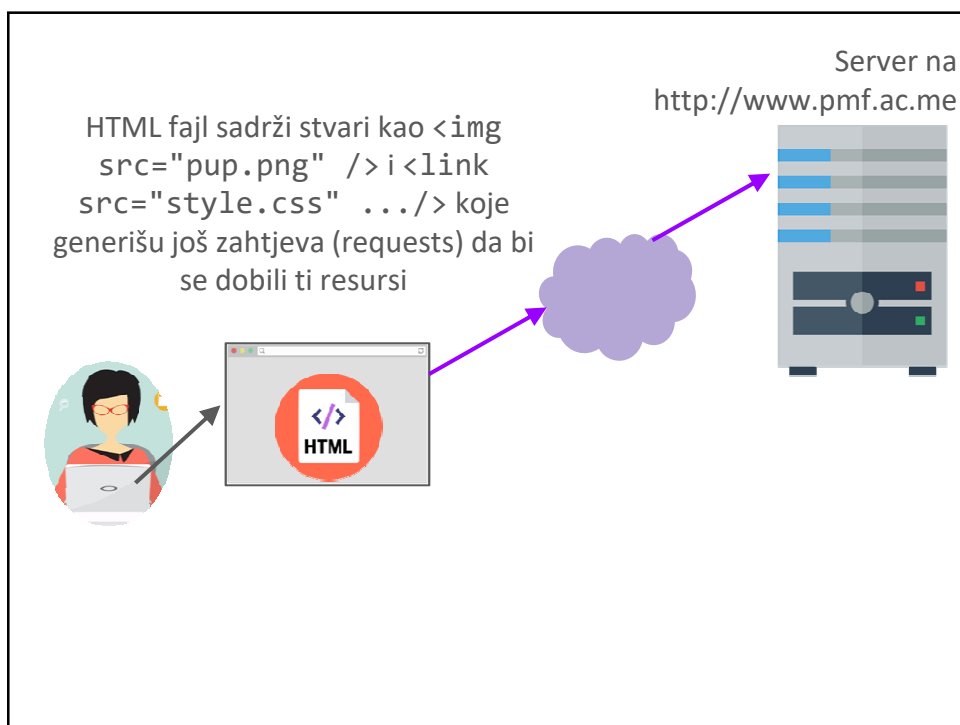
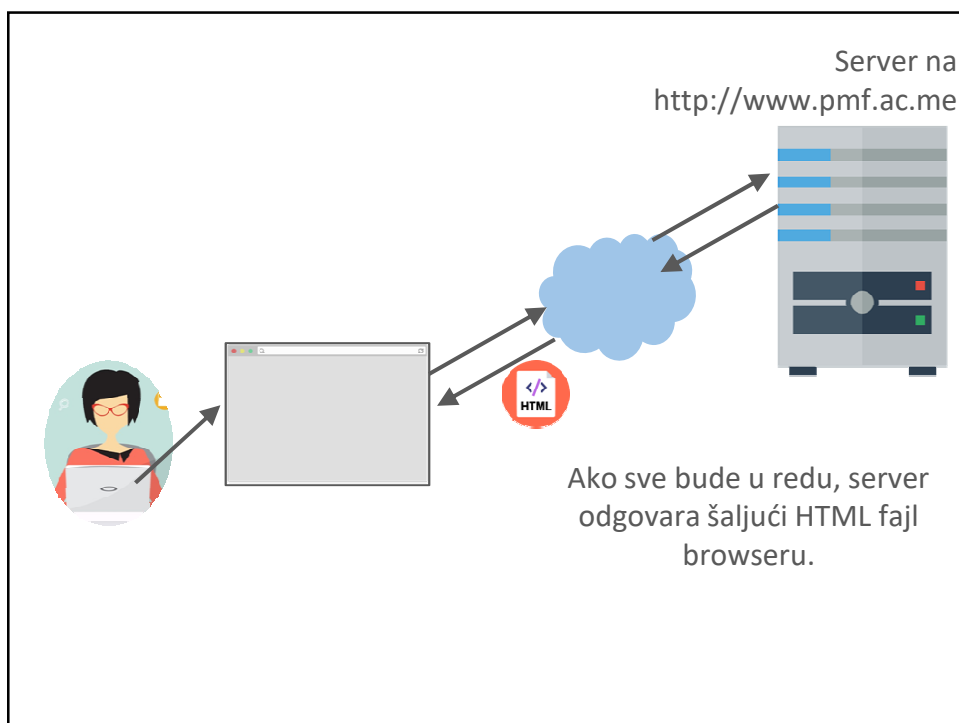
Na laptopu se radi  
browser, npr.  
Chrome

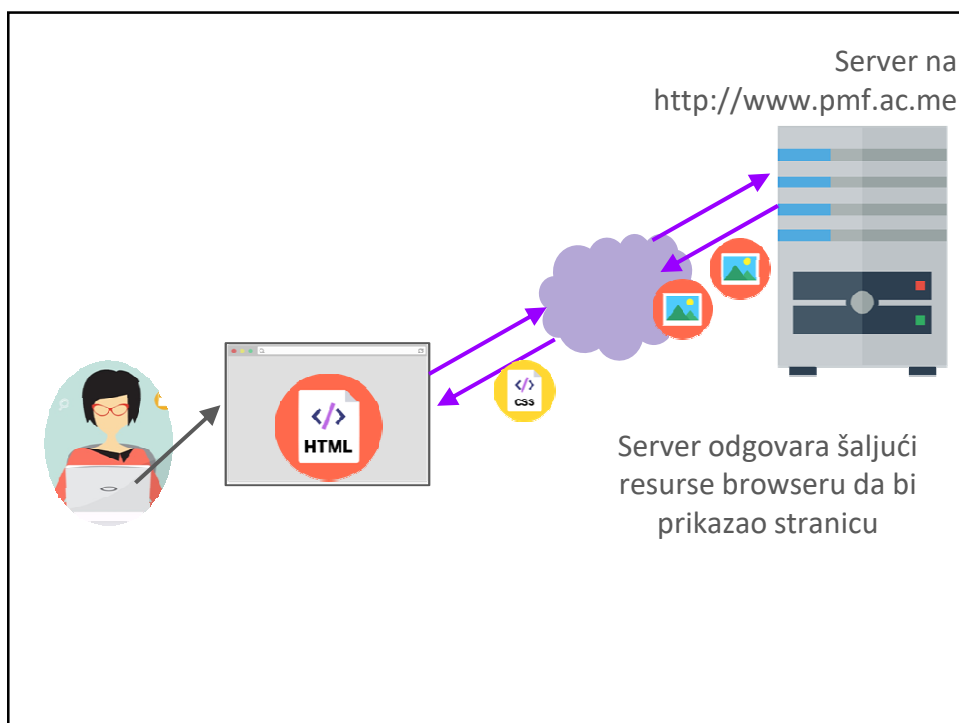


Unosite URL u  
adresnoj liniji i  
pritiskate "enter"









Konačno, kada se učitaju svi resursi,  
vidimo učitanu veb-stranicu





## Naučili smo (nadam se...)

Da kreiramo veb-stranice koje:

- Izgledaju kako mi želimo
- Mogu se povezati sa drugim stranicama
- Prikazuju se različito na različitim uređajima



Ali naše stranice ništa ne rade :

- Get user input
- Save user input
- Interaktivno akivaju i otkrivaju elemente na zahtjev korisnika



**Enter JavaScript!**

# JavaScript

## JavaScript

JavaScript je programski jezik.

Trenutno jedini jezik koji browser izvršava (natively). (Pokušaji da se to promijeni: [link](#))

Dakle, morate naučiti JavaScript.





## JavaScript

- Kreiran 1995 – Brendan Eich  
(jedna od osnivača Mozilla-e; otpušten 2014)
- JavaScript nema veze sa jezikom Java
  - Ime dato zbog marketinga [marketing reasons](#)
- Prva verzija napisana za 10 dana
- Nekoliko važnih odluka je donešeno zbog politike kompanije a ne zbog tehničkih razloga

*"I was under marketing orders to make it look like Java but not make it too big for its britches ... [it] needed to be a silly little brother language." ([source](#))*

## JavaScript

- Kreiran 1995 – Brendan Eich  
(jedna od osnivača Mozilla-e; otpušten 2014)
- JavaScript nema veze sa jezikom Java
  - Ime dato zbog marketinga [marketing reasons](#)
- Prva verzija napisana za 10 dana
- Nekoliko važnih odluka je donešeno zbog politike kompanije a ne zbog tehničkih razloga

Drugim riječima:

**JavaScript is messy and full of drama...  
and our only option.**

(mada, postaje malo bolje posljednjih godina)

# JavaScript in the browser

## Kod na veb-stranicama

HTML can embed JavaScript files into the web page via the `<script>` tag.

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

## console.log

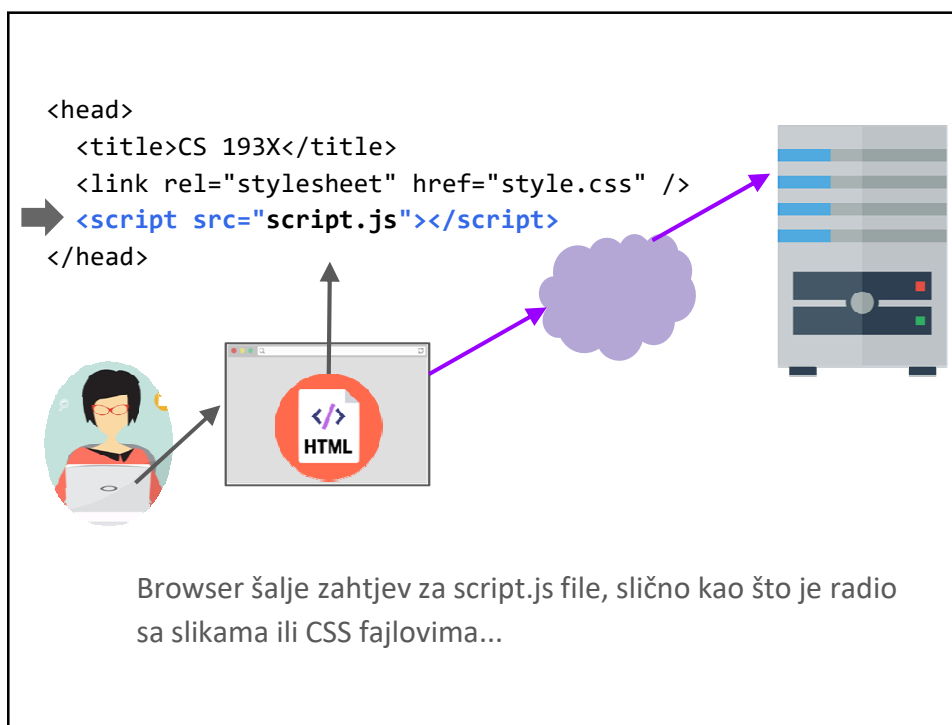
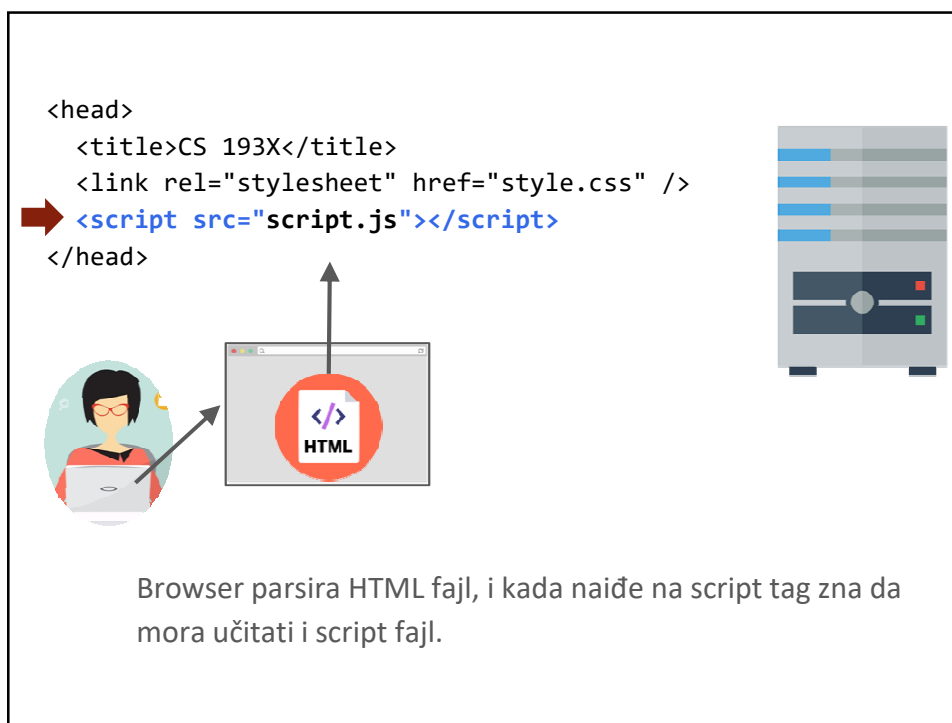
Štampanje poruka pozivom `console.log()`:

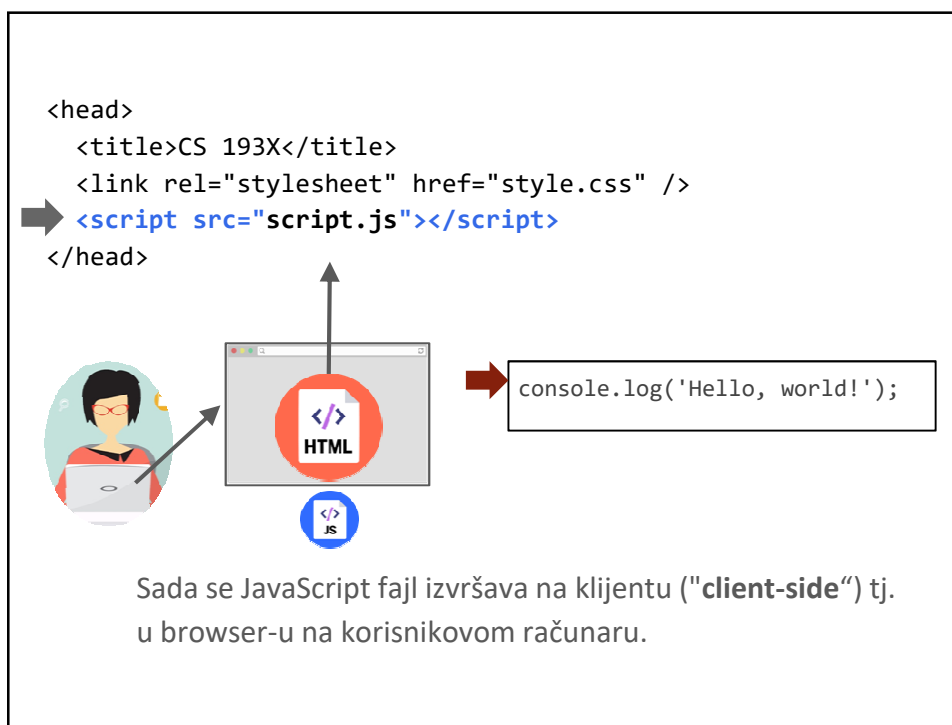
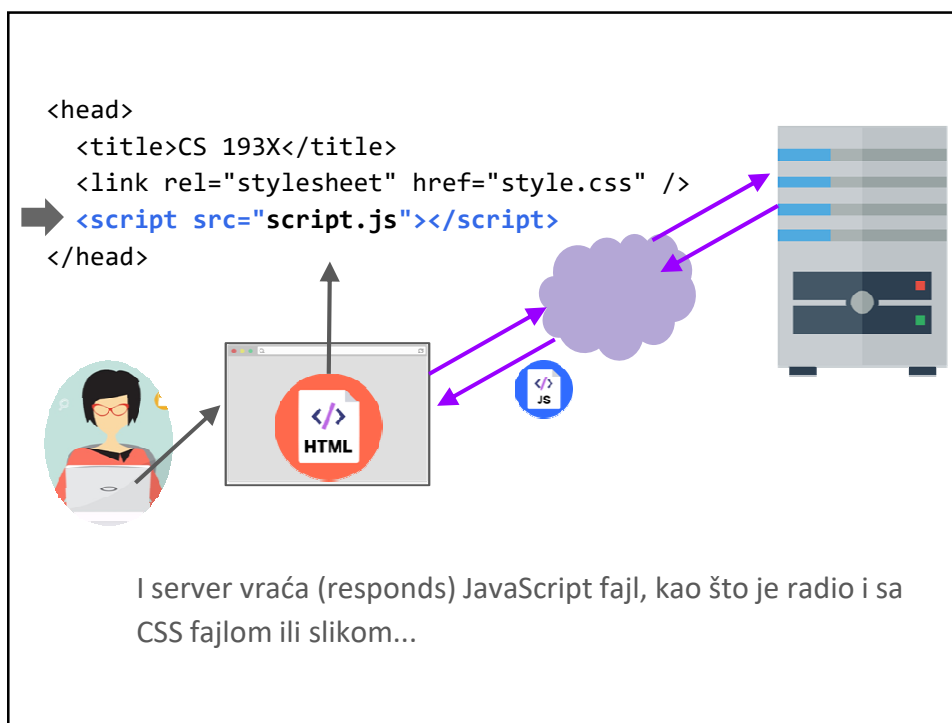
`script.js`

```
console.log('Hello, world!');
```

Ovo je ekvivalentno sa Javinim  
`System.out.println`, `print`, `printf`, itd..

Kako se učitava JavaScript?





## JavaScript izvršavanje

Ne postoji "main method"

- Script fajl se izvršava od vrha prema dnu.

Nema prevođenja(**compilation**) od strane programera

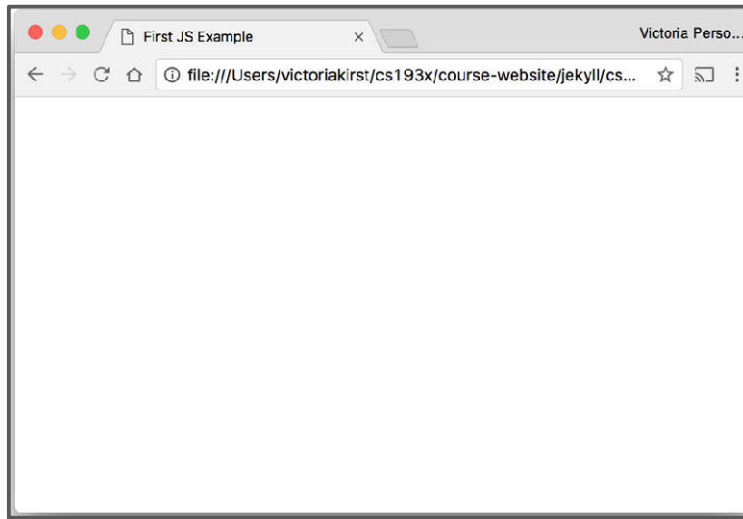
- JavaScript se kompajlira i izvršava od strane browser-a  
Napomena: Ovo je malo drugačije od "interpretiranja":  
pogledajte [just-in-time \(JIT\) compilation](#))

**first-js.html**

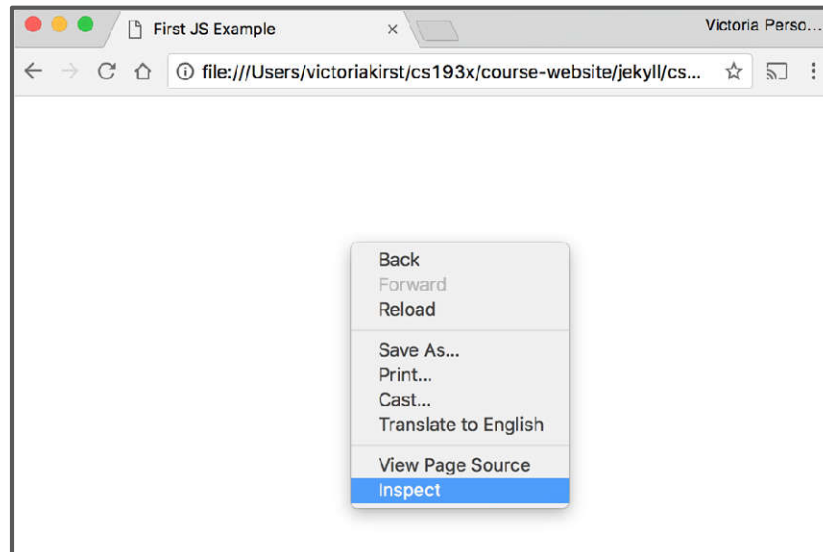
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>First JS Example</title>
    <script src="script.js"></script>
  </head>
  <body>
  </body>
</html>
```

**script.js**

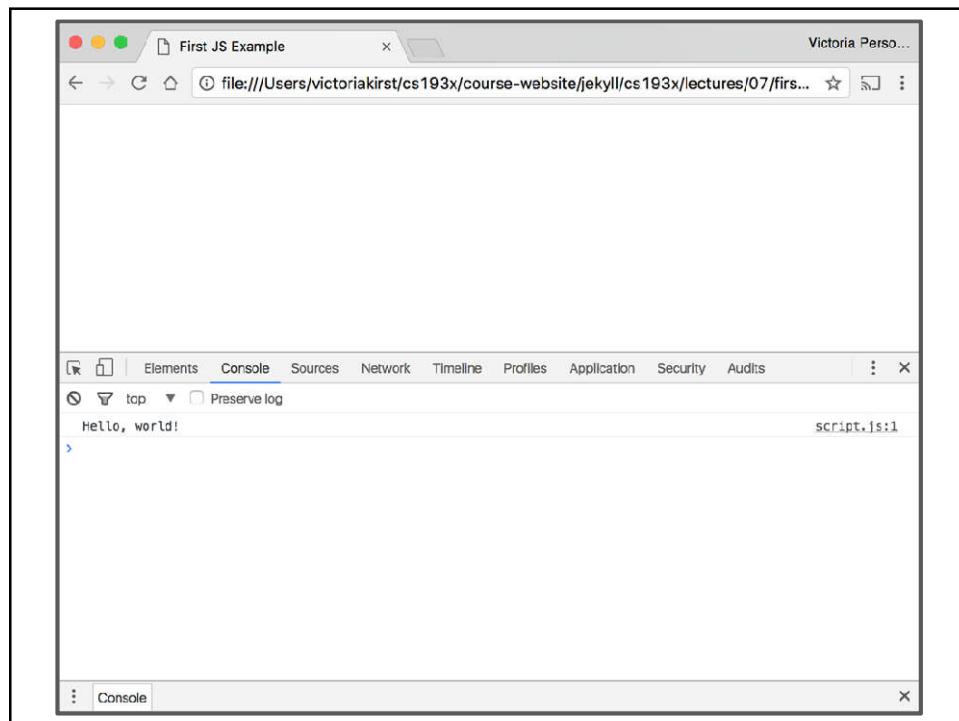
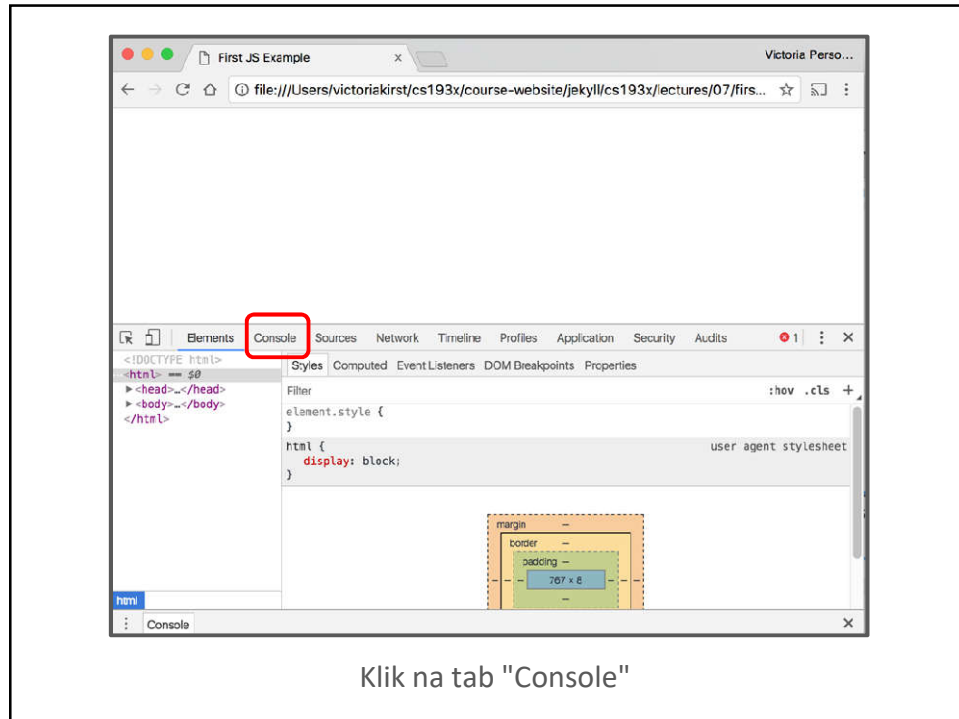
```
console.log('Hello, world!');
```



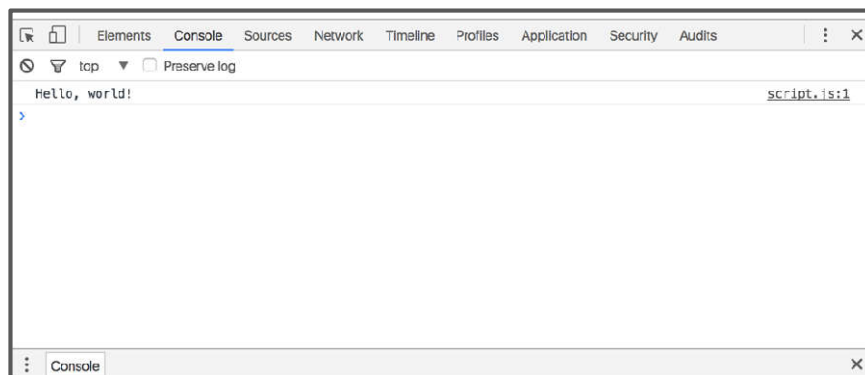
Ništa se ne dešava!



Desni klik (ili control-click na Mac-u) i izaberite "Inspect"







Tab "Console" je takođe [REPL](#), ili interaktivni shell, pa možete kucati izraze, itd.

## JavaScript osobine jezika

## Isto kao u Java/C++/C

for-ciklus:

```
for (let i = 0; i < 5; i++) { ... }
```

while-ciklus:

```
while (notFinished) { ... }
```

komentari:

```
// comment or /* comment */
```

Uslovne naredbe (if statements):

```
if (...) {  
  ...  
} else {  
  ...  
}
```

## Funkcije

Jedan način definisanja funkcija u jeziku JavaScript :

```
function name() {  
  statement;  
  statement;  
  ...  
}
```

**script.js**

```
function hello() {  
  console.log('Hello!');  
  console.log('Welcome to JavaScript');  
}  
  
hello();  
hello();
```



top

Hello!  
Welcome to JavaScript  
Hello!  
Welcome to JavaScript

> |

Console output

**script.js**

```
function hello() {  
  console.log('Hello!');  
  console.log('Welcome to JavaScript');  
}  
  
hello();  
hello();
```

Browser prvo “izvršava” definiciju funkcije, ali se samo kreira funkcija hello (koja se ne izvršava), slično deklaraciji promenljive.



top

Hello!  
Welcome to JavaScript  
Hello!  
Welcome to JavaScript

> |

Console output

**script.js**

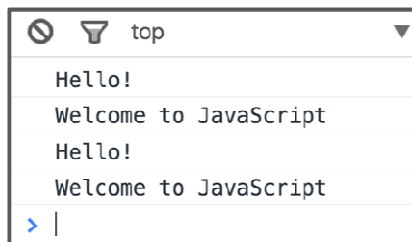
```
hello();  
hello();  
  
function hello() {  
  console.log('Hello!');  
  console.log('Welcome to JavaScript');  
}
```

**Q: Radi li ovo?****script.js**

```
hello();  
hello();  
  
function hello() {  
  console.log('Hello!');  
  console.log('Welcome to JavaScript');  
}
```

**A: Da, za baš ovu sintaksu.**

Deklaracije funkcija su "**hoisted**" ([mdn](#)). Možete smatrati da se definicije "pomjeraju" na vrh opsega u kom su definisane (iako se to stvarno ne dešava).

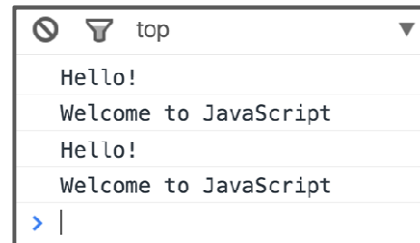
**Console output**

**script.js**

```
hello();  
hello();  
  
function hello() {  
  console.log('HUello!');  
  console.log('Welcome to JavaScript');  
}
```

**Upozorenje:**

- Postoje drugi načini definisanja funkcija kada nema hoisting-a.
- Pokušajte da se ne oslanjate na hoisting kada programirate ([it gets bad](#)).

**Console output**