

Kreiranje igara u programskom jeziku JavaScript

Prvi dio

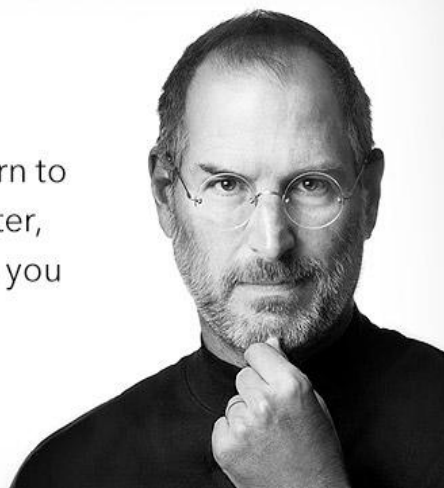


Pregled

- Predstavljanje i par napomena
- Opis kursa
- Instalacija alata
- HTML, par primjera
- CSS, par primjera
- JavaScript

"Everybody in this country should learn to program a computer, because it teaches you how to think"

STEVE JOBS



Predstavljanje

- Ko sam ja?
- Ko ste vi?
- Za koga je kurs i zašto smo ovdje
- Događaj EU Code Week
 - Naša nedelja je jedan mjesec



Napomene

- Pratite šta predavač priča i bilježite, **disciplina**
- Kurs će trajati minimum 4 sedmice
 - 2h, sa pauzom od 10 minuta
- Projekat na kraju
 - U više grupa, svaka grupa će trebati na napravi igru
 - Igru će biti potrebno i prezentovati
 - Uvijek ću vam biti na raspolaganju i tokom i nakon kursa za bilo kakvu pomoć
- Trudite se da se međusobno upoznate
 - Možda ćete nekad baš vi sa nekim iz ove grupe napraviti nešto veliko
- Sav kod na GitHub [linku](#)
- Pristup slajdovima dobićete posle časa
- Sva obavještenja dobijaćete na mail, a biće i na [sajtu](#)



Savjeti

- Stalno učite, ko voli programiranje to je zabava!
- Jezik nije bitan, bitan je uvijek koncept i logičko razmišljanje
- Naučite da guglate, ono oko čega se trenutno mučite da riješite, velika je šansa je neko prije vas imao isti problem
 - **To ne znači nikako da radite copy/paste!**



Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.

— Linus Torvalds —

AZ QUOTES

Opis kursa

- Instalacija alata
- Osnovne HTML-a
- Šta je CSS
- JavaScript
- Canvas
- Kreiranje igre

HTML



CSS

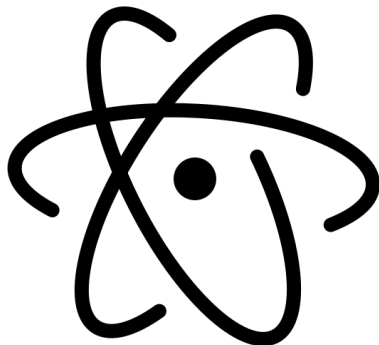
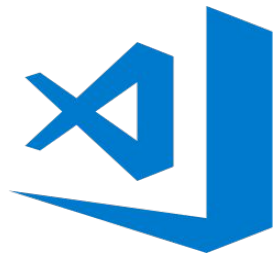


JS

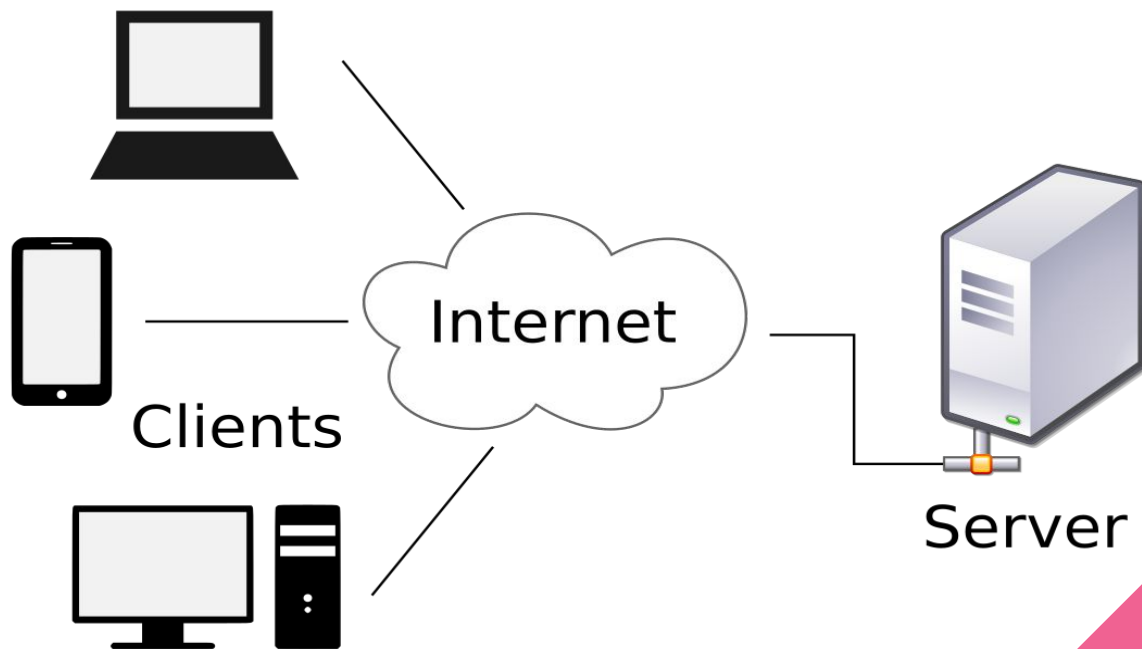


Instalacija alata (VSCode - preporuka)

- Zašto baš VSCode
 - [Download](#)
- Osnovne funkcionalnosti
- Dodavanje plugin
- Ostali alati



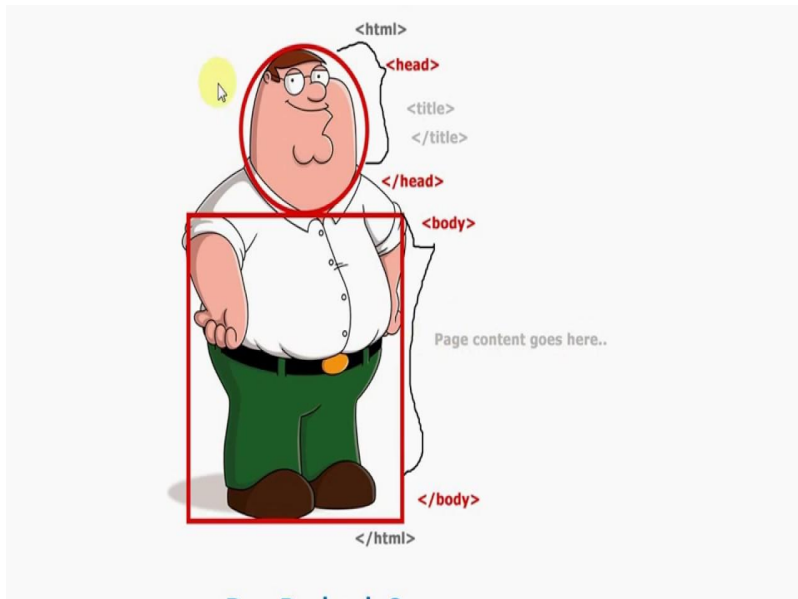
Klijent-server komunikacija (web)



HTML

- Sadržaj na web stranici

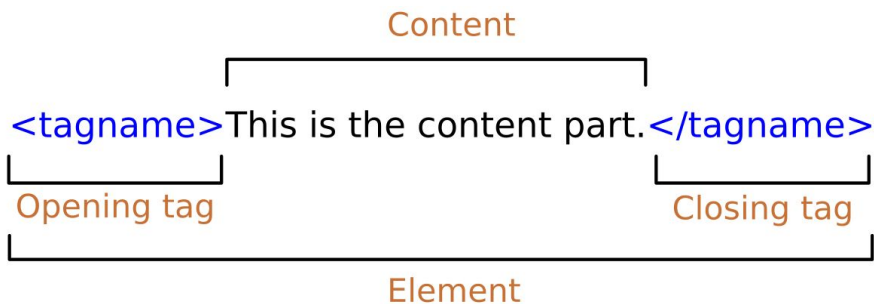
```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```



HTML elementi i atributi

- Čine ono što vidite na web stranici. Da pomenemo neke:

- Tekst
- Linkovi
- Div i span
- Liste
- Slike
- Tabele
- Forme
- Media elementi
- **Canvas**



- HTML atributi dodatno opisuju HTML element
- Neki elementi nemaju closing tag (void elementi)

Primjer 1 (code 1, code 2)



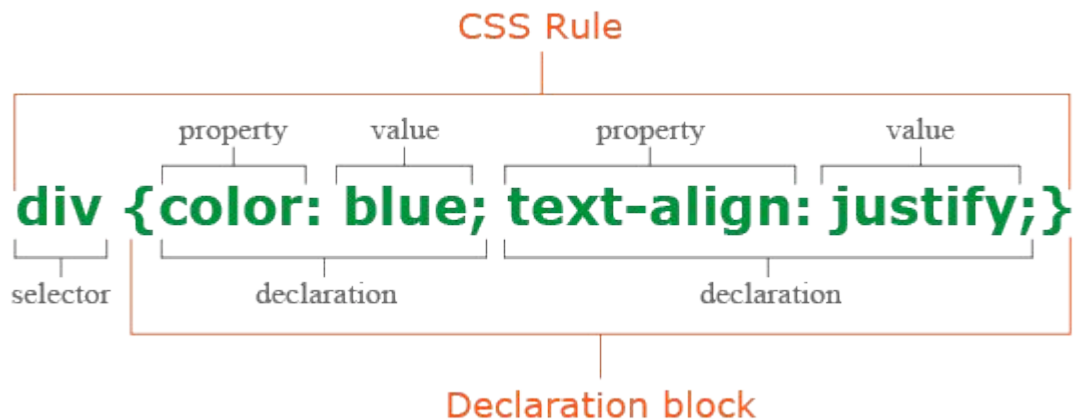
CSS



- Definišete izgled vaše stranice
 - Stranica bez CSS je stranica koju niko neće htjeti da otvori drugi put
- Pogledajte kako izmjenom CSSa na stranici gdje imamo isti HTML kod mijenjamo dobijamo različite varijante strane
 - [ZenGarden](#)
- CSS kod obično pišete u odvojenom fajlu čija je ekstenzija .css
- Taj fajl onda trebate na neki način da ubacite u HTML
 - U head elementu HTMLa dodate ovaj kod (path je putanja do CSS fajla, a mystyle.css ime CSS fajla, fajl nije obavezno nazvati style)
 - `<link rel="stylesheet" type="text/css" href="path/style.css">`

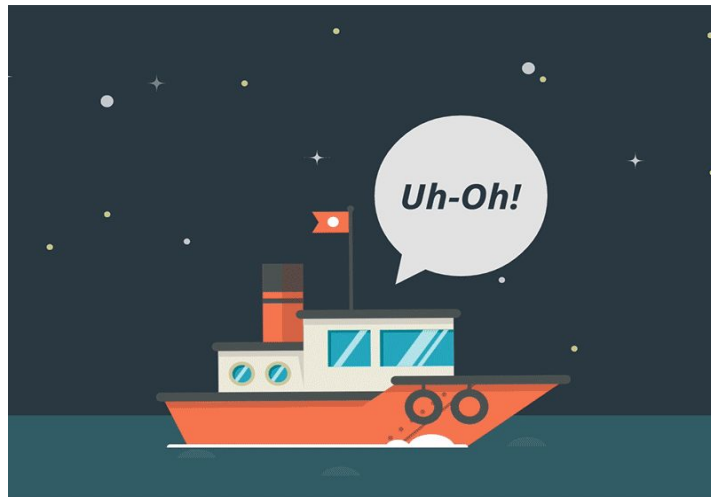
CSS pravila i selektori

- CSS selektori vam omogućavaju da selektujete jedan ili više elemenata
- U deklarativnom dijelu CSS pravila definišete izgled selektovanog HTML elementa



Primjeri

- Pogledajmo neke o primjera
 - Code 3
 - Code 4
- Primjeri (sa animacijama)
 - [Code 1](#)
 - [Code 2](#)
 - [Code 3](#)
 - [Code 4](#)
- Kako da nađete još ovakvih primjera?
- Odličan dodak koji možete koristiti za animacije
 - `animate.css`



Usporimo malo

- Ovo nije kurs HTML-a i CSS-a
- Za ovaj kurs nije potrebno poznavanje ni HTML ni CSS ali je bitno da znate šta je razlika između njih i gdje se koristi HTML, a gdje CSS
- Šta nam nedostaje da bismo krenuli da kreiramo igru?

HTML

Structure



CSS

Style



JavaScript

Behaviour





Software is eating the world.

Marc Andreessen

 quote fancy

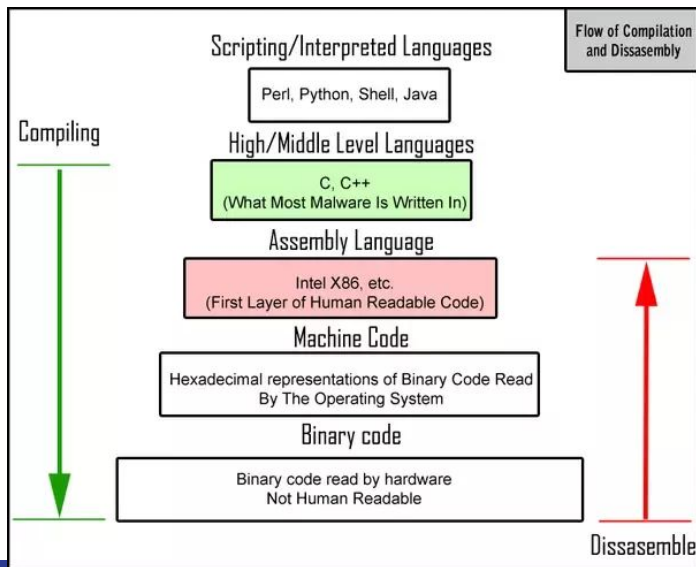
JavaScript, uvod

- Molim vas ne brkajte JavaScript i Java programske jezike, ova dva jezika se potpuno razlikuju!
- 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, interpreted

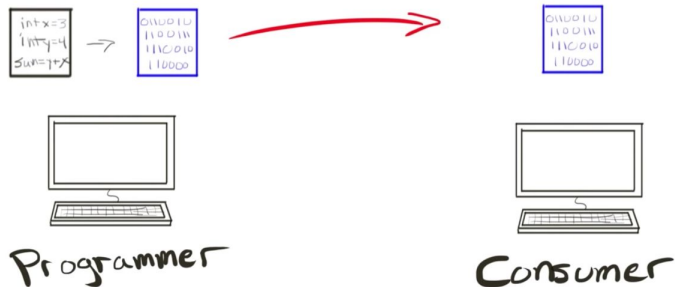
- JavaScript je interpreted programski jezik. A kakav bi eventualno još mogao da bude?
- Zašto se uvode programski jezici



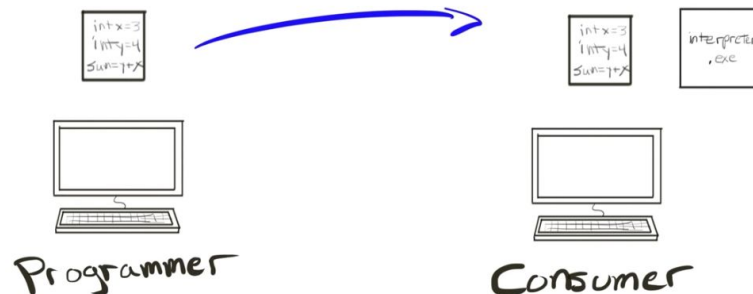
Programski jezici, tipovi

- Šta znači kad kažem high level programming language?
- Šta je razlika između procesa compile i interprete kod programskih jezika?

Compiled



Interpreted



Compile vs Interpret procedure

Programmer/Developer's Computer

```
int x = 3;  
int y = 5;  
int sum = x + y;  
Sketchout &&sum;
```



```
010010001010  
1011110010001  
0110011000  
1100001111  
10001101110  
10110001100
```

Programmer/Developer's Computer

```
int x = 3;  
int y = 5;  
int sum = x + y;  
Sketchout &&sum;
```

Consumer's Computer

```
010010001010  
1011110010001  
0110011000  
1100001111  
10001101110  
10110001100
```



Consumer's Computer

```
int x = 3;  
int y = 5;  
int sum = x + y;  
Sketchout &&sum;
```



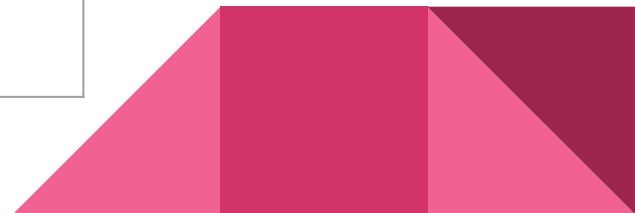
```
0110011001  
1001100101  
0111000010  
10001111
```



Compile vs Interpret, prednosti/mane

Compilation	
Prednosti	Mane
Brz	Nije cross platform
Source kod je private	Zahtijeva dodatni korak kompilacije (duži razvoj)

Interpretation	
Prednosti	Mane
Cross platform	Sporiji
Bez dodatnog koraka (lakši debug)	Public source kod

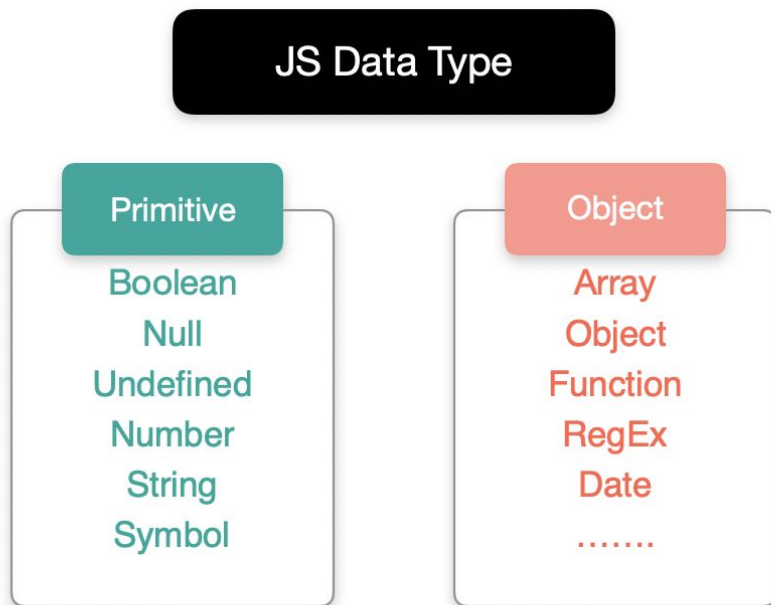


Dodavanje JSa na stranicu i varijable

- JS kod dodaje se unutar HTML elementa koji se zove script, a na src atribut se kači lokacija do JS koda. Taj element se obično stavlja na samom dnu body elementa
- Varijable
 - Ne koristi se tip podatka već neka od ključnih riječi
 - **let**
 - **const**
 - **var**



JS tipovi podataka



Primitivni tipovi podataka

- Brojevi
 - Cijeli ili decimalni
 - Za pretvaranje stringa u broj koristite **parseInt** ili **parseFloat** ugrađene funkcije
 - Provjerimo rezultat sabiranja $0.1 + 0.2$
 - Pazite na ovo
 - Math objekat
- Stringovi
 - Ne postoji char kao tip podatka već samo string
 - Elementu stringa pristupate sa `some_string[index]`
 - String je **immutable** tip podatka
 - Metodi za rad sa stringovima
- Boolean
 - True ili false vrijednost, logički operatori
- Null i undefined



Neprimitivni tipovi podataka


- Nizovi

- Dinamički se širi veličina niza
- Kreiraju se sa []
- Elementi niza mogu da budu različiti tipovi podataka
- Neke od metoda

- Objekti

- Imaju oblik key:value i kreiraju se sa {}
- Koristićemo ih dosta
- Primjer

- let **game** = {name: "Crash", genre: "adventure", start: function(){console.log("play game")}}

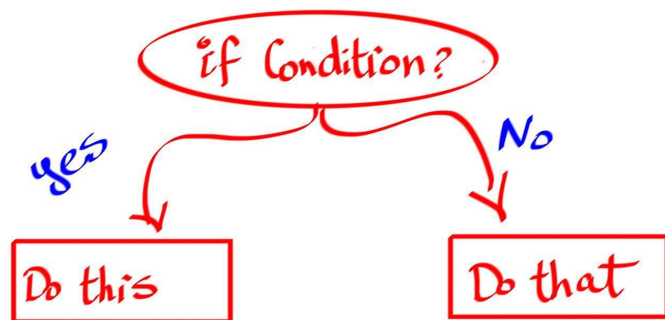
Object	Properties	Methods
	<p>car.name = Fiat</p> <p>car.model = 500</p> <p>car.weight = 850kg</p> <p>car.color = white</p>	<p>car.start()</p> <p>car.drive()</p> <p>car.brake()</p> <p>car.stop()</p>

Truthy i falsy vrijednosti

- Koristite `===` za poređenje, nikad ne koristiti `==`

Falsy	Truthy
<code>""</code>	<code>"foo"</code>
<code>0, +0, -0</code>	<code>23</code>
<code>null</code>	<code>{ a:1 }</code>
<code>NaN</code>	<code>[1,3]</code>
<code>false</code>	<code>true</code>
<code>undefined</code>	<code>function(){..}</code>
	<code>...</code>

Kondicije



```
if (condition1) {  
    block of code to be executed if condition1 is true  
} else if (condition2) {  
    block of code to be executed if the condition1 is false and condition2 is true  
} else {  
    block of code to be executed if the condition1 is false and condition2 is false  
}
```

Petlje

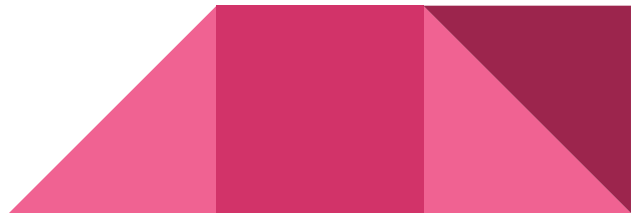
- Koriste se najčešće kada želite da obavite neku operaciju nad skupom/nizom elementa

While

```
while (condition) {  
    code block to be executed  
}
```

For

```
for (statement 1; statement 2; statement 3) {  
    code block to be executed  
}
```



Funkcije

- Zna li zašto se koriste funkcije u programiranju?

```
function name(parameter1, parameter2, parameter3) {  
    code to be executed  
}
```

- Šta je razlika između parametra i argumenta?
- Zna li da kod JS, funkcija može da primi drugu funkciju kao argument (callback) ili da vrati drugu funkciju (closure i higher order function)



Zadaci

- Uvodni zadaci
 - Pogledajmo fajl examples.js
- Unity i Unreal Engine
 - C# i C++
- Zna li za [PlayCanvas](#)
 - Pogledajmo primjere
- Zna li sa [Phaser](#)
 - Pogledajmo primjere
- [Unity vs PlayCanvas](#)
- [Phaser vs PlayCanvas](#)
- Mi ćemo da koristimo samo JS
 - Da napravimo dvije vrlo jednostavne igre



Par napomena za kraj prvog predavanja

- Napomena: Svi ćete dobiti pristup cjelokupnom kodu i slajdovima za kurs **Web programiranje** (ukoliko neko želi sam da uči dublje CSS i JavaScript)
- Predlažem freeCodeCamp, Udemy, ili neku od sličnih platformi za online kurseve
- Programiranje se uči svakog dana!
- Treba li mi matematika za pravljenje igara?
 - Odgovor je naravno
- Treba li mi matematika da bi se bavio web programiranjem
 - Obično ne, ali matematika je bitna za nešto što se zove **problem solving**
- **Predavanja ne pokrivaju ni 5% procesa učenja programiranja**
- **Domaći 1**

Kraj prvog dijela

Sledeći čas radimo Canvas, pripremamo polako teren za kreiranje prve igre

[Code 1](#)

[Code 2](#)

Igra za sledeći čas (ping-pong)

