

Element canvas

- **Canvas** je pravougaona oblast na veb stranici, podrazumevano nema okvir niti sadržaj
- Može se koristiti za grafičko iscrtavanje, može se animirati, može reagovati na JavaScript događaje
- Predstavlja dvodimenzionu mrežu, gornji levi ugao ima koordinate (0,0)
- Svojstva **width** i **height** određuju njegovu širinu i visinu

```
<canvas id="slika" width="200" height="100" style="border:1px solid black;">
</canvas>
```



Crtanje duži

- `getContext()` – vraća objekat koji raspolaže svojstvima i metodama za crtanje
- `moveTo(x,y)` – definiše početnu tačku duži
- `lineTo(x,y)` – definiše krajnju tačku duži
- `stroke()` – iscrtava objekat
- `lineWidth` – određuje debljinu linije u pikselima
- `strokeStyle` – određuje boju, gradijent ili uzorak kojim crtamo

```
var c = document.getElementById("slika");  
var ctx = c.getContext("2d");  
ctx.moveTo(0,0);  
ctx.lineTo(200,100);  
//ctx.lineWidth = 3;  
//ctx.strokeStyle = "green";  
ctx.stroke();
```



Crtanje, bojenje, brisanje pravougaonika

- `fillStyle` – određuje boju, gradijent ili uzorak kojim bojimo
- `fillRect(x,y,a,b)` – crta se i boji pravougaonik sa gornjim levim temenom (x, y) dimenzije $a \times b$ piksela
- `strokeRect(x,y,a,b)` – samo se iscrtava dati pravougaonik (ne boji se)
- `clearRect(x,y,a,b)` – brišu se vrednosti piksela u okviru datog pravougaonika

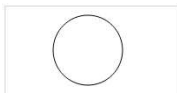
```
var c = document.getElementById("slika");  
var ctx = c.getContext("2d");  
ctx.fillStyle = "#FF0000";  
ctx.fillRect(0,0,150,75);  
ctx.clearRect(100,50,30,15);
```



Crtanje kruga

- `beginPath()` – započinje novu putanju ili resetuje tekuću
- `arc(x,y,r,pocetniUgao,završniUgao)` – generiše se luk sa centrom u tački (x, y) poluprečnika r ; ako želimo da iscrtamo krug potrebno je postaviti početni ugao na 0, a završni na $2 * \text{Math.PI}$

```
var c = document.getElementById("slika");  
var ctx = c.getContext("2d");  
ctx.beginPath();  
ctx.arc(95,50,40,0,2*Math.PI);  
ctx.stroke();
```



Gradijenti

- Gradijenti se mogu koristiti za popunjavanje pravougaonika, kruga, duži, teksta,...
- Postoje dva tipa gradijenta:
 - `createLinearGradient(x, y, x1, y1)` – linearni gradijent
 - `createRadialGradient(x, y, r, x1, y1, r1)` – kružni gradijent
- Potrebno je postaviti dve ili više boja pozivom metoda `addColorStop()` – zadaje se boja i njena pozicija (vrednost između 0 i 1)
- Da bismo iskoristili gradijent potrebno je postaviti svojstvo `fillStyle` ili `strokeStyle` na vrednost gradijenta i onda iscrtati oblik
- Primeri: 01_crtanje.html, 02_crtanje.html, analogni_sat.html

Gradijenti

```
var c=document.getElementById("slika");  
var ctx=c.getContext("2d");  
  
var gradijent=ctx.createLinearGradient(0,0,200,0);  
gradijent.addColorStop(0,"red");  
gradijent.addColorStop(1,"white");  
  
ctx.fillStyle=gradijent;  
ctx.fillRect(10,10,150,80);
```



Ispisivanje teksta

- **font** – definiše font za tekst
- **fillText(text,x,y)** – iscrtava “popunjeni” tekst počev od pozicije (x, y)
- **strokeText(text,x,y)** – iscrtava tekst (bez popunjavanja)
- **fillStyle** – definiše boju teksta
- **textAlign** – definiše horizontalno poravnanje teksta (vrednosti: left, right, center,...)
- **textBaseline** – postavlja baznu liniju za tekst (vrednosti: top, middle, bottom,...)

```
var c = document.getElementById("slika");  
var ctx = c.getContext("2d");  
ctx.font = "30px Arial";  
ctx.fillText("Zdravo svete",10,50);  
//ctx.strokeText("Hello World",10,50);
```

Zdravo sveteHello World

jQuery osnove

- jQuery je JavaScript biblioteka koja pojednostavljuje programiranje u JavaScript-u
- Uobičajeni zadaci se “pakuju” u metode koji se pozivaju jedinstvenom naredbom u kodu
- jQuery pojednostavljuje i DOM manipulaciju

Korišćenje jQuery biblioteke

- Postoji nekoliko načina na koje je moguće koristiti jQuery
 - preuzimanjem kompresovane ili nekompresovane jQuery biblioteke sa `jquery.com`

```
<head>  
  <script src="jquery-3.2.1.min.js"></script>  
</head>
```

- uključivanjem jQuery biblioteke sa mreža za dostavu sadržaja (*CDN*, *Content Delivery Network*), kao što je Google

```
<head>  
  <script  
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">  
  </script>  
</head>
```

jQuery sintaksa

- Osnovna sintaksa je `$(selector).action()`
 - znak `$` označava pristup jQuery-u
 - (selector) služi za traženje HTML elemenata
 - action() označava akciju koja će se izvesti nad elementima
 - pozivanjem funkcije `$()` i prosleđivanjem selektora pravi se novi jQuery objekat, koji raspolaže svojstvima i metodama
- Primer: `$("p").hide()` – sakriva sve elemente `<p>`
- Da bi se sprečilo izvršavanje jQuery metoda pre nego što se stranica učita, metod se može umetnuti u događaj da je dokument spreman:

```
$(document).ready(function(){  
    // ovde se piše jQuery method  
});
```

jQuery selektori

- jQuery selektori su zasnovani na CSS selektorima, uz neke dopune
- Svi selektori počinju znakom \$ i navode se u zagradama: \$(...)
- Osnovni selektori:
 - ime elementa: \$("p") – primer: 03_jq.html
 - identifikator elementa: \$("#istaknuti_pasus") – primer: 04_jq.html
 - identifikator klase: \$(".istaknut") – primer: 05_jq.html
- Dodatni selektori:
 - \$("*") – označava sve elemente
 - \$(this) – označava tekući HTML element
 - \$("p:first") – prvi element <p>
 - \$("tr:even") i \$("tr:odd") – parni i neparni redovi tabele
 - \$("[src]") – svi elementi sa atributom src
 - \$("a[target='_blank']") i \$("a[target!='_blank']") – svi linkovi kod kojih je vrednost atributa target jednaka, odnosno nije jednaka '_blank'

jQuery događaji

- Događaji mišem: `click`, `dblclick`, `mouseenter`, `mouseleave`, `hover`, ...
- Događaji tastaturom: `keypress`, `keydown`, `keyup`, ...
- Događaji dokumenta: `load`, `unload`, `resize`, ...
- Događaji formulara: `submit`, `change`, `focus`, `blur`, ...
- Primeri: 06_jq.html

jQuery efekti

- HTML elementi se mogu sakriti i ponovo prikazati metodama `hide()`, `show()` i `toggle()`
- Element se može postepeno pojavljivati ili postepeno gubiti sa slike metodama `fadeIn()`, `fadeOut()`, `fadeToggle()`
- Element se može postepeno pojavljivati “klizanjem naniže”, odnosno postepeno nestajati “klizanjem naviše” metodama `slideDown()`, `slideUp()`, `slideToggle()`
- Sve ove funkcije mogu imati vrednost argumenata “slow”, “fast” ili broj milisekundi
- Metod `stop()` se koristi za zaustavljanje efekta pre nego što se završi
- Primeri: 07_jq.html, 08_jq.html, 09_jq.html

Callback funkcije

- U JavaScript-u se naredbe izvršavaju linijski, međutim naredna linija koda može krenuti sa izvršavanjem iako se prethodni efekat nije u potpunosti završio
- **Callback funkcija** je funkcija koja se izvršava tek kada se tekući efekat u potpunosti završi
- Sintaksa: `$(selector).hide(brzina,callback)`
- Primeri: 10_jq.html

Ulančavanje metoda

- U jQuery-u je moguće ulančavati akcije/metode čime se u jednoj naredbi izvršava veći broj metoda na istom elementu
- Ovo je moguće jer metode za postavljanje vrednosti vraćaju objekat na kome su pozvane
- Na ovaj način pregledači ne moraju više puta da pronalaze isti element
- I na ovaj način naredna akcija može krenuti sa radom pre nego što se prethodna u potpunosti završi
- Primeri: 11_jq.html

Manipulacija CSS-om korišćenjem jQuery-a

- `addClass()` – dodaje jednu ili više klasa datom elementu
- `removeClass()` – uklanja jednu ili više klasa sa datog elementa
- `toggleClass()` – naizmenično dodaje i uklanja jednu ili više klasa sa datog elementa
- `css()` – postavlja ili vraća stilski atribut
 - vrednost svojstva se vraća: `css("naziv_svojstva")`
 - vrednost jednog svojstva se postavlja sa:
`css("naziv_svojstva", "vrednost")`
 - vrednost više svojstava se postavlja sa:
`css("naziv_svojstva": "vrednost", "naziv_svojstva": "vrednost", ...)`
- Primeri: 12_jq.html, 13_jq.html

Manipulacija DOM-om korišćenjem jQuery-a

- Set/get metode
 - `text()` – postavlja ili vraća tekstualni sadržaj označenog elementa
 - `html()` – postavlja ili vraća sadržaj označenog elementa (uključujući HTML oznake)
 - `val()` – postavlja ili vraća vrednost polja formulara
 - `attr()` – postavlja ili vraća vrednost atributa označenog elementa
 - Primeri: 14_jq.html, html.html
- Metode za obilazak DOM stabla
 - `parent()`, `parents()` – vraća roditeljski element, odnosno sve pretke
 - `children()`, `find()` – vraća sinove, odnosno sve potomke
 - `next()`, `prev()` – vraća narednog, odnosno prethodnog brata
 - Primeri: 15_jq.html