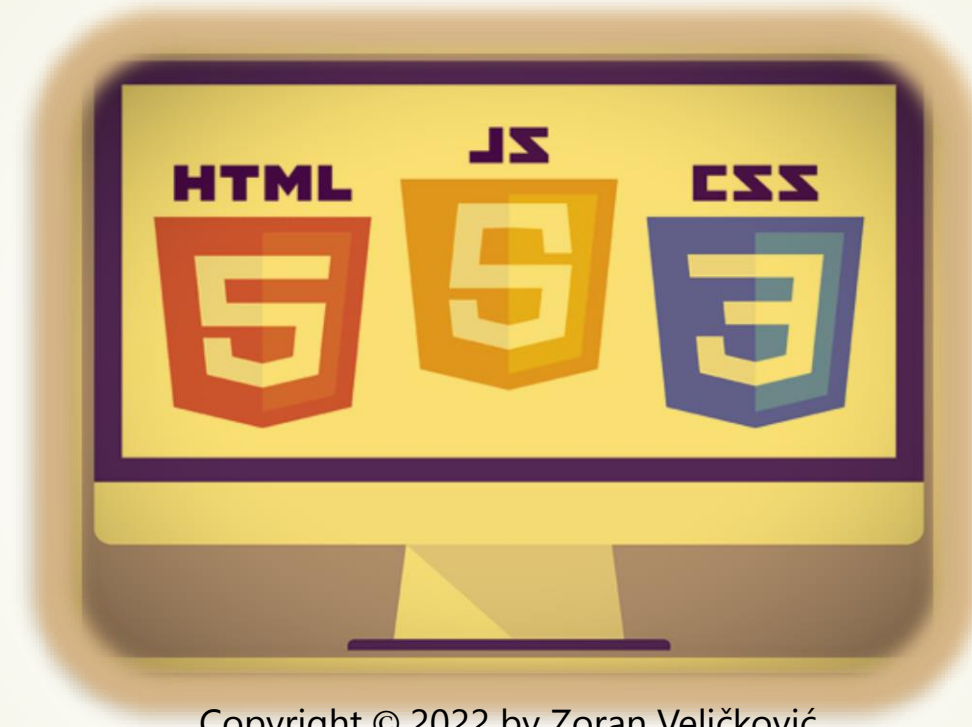




Akademija tehničko-vaspitačkih strukovnih studija



Copyright © 2022 by Zoran Veličković

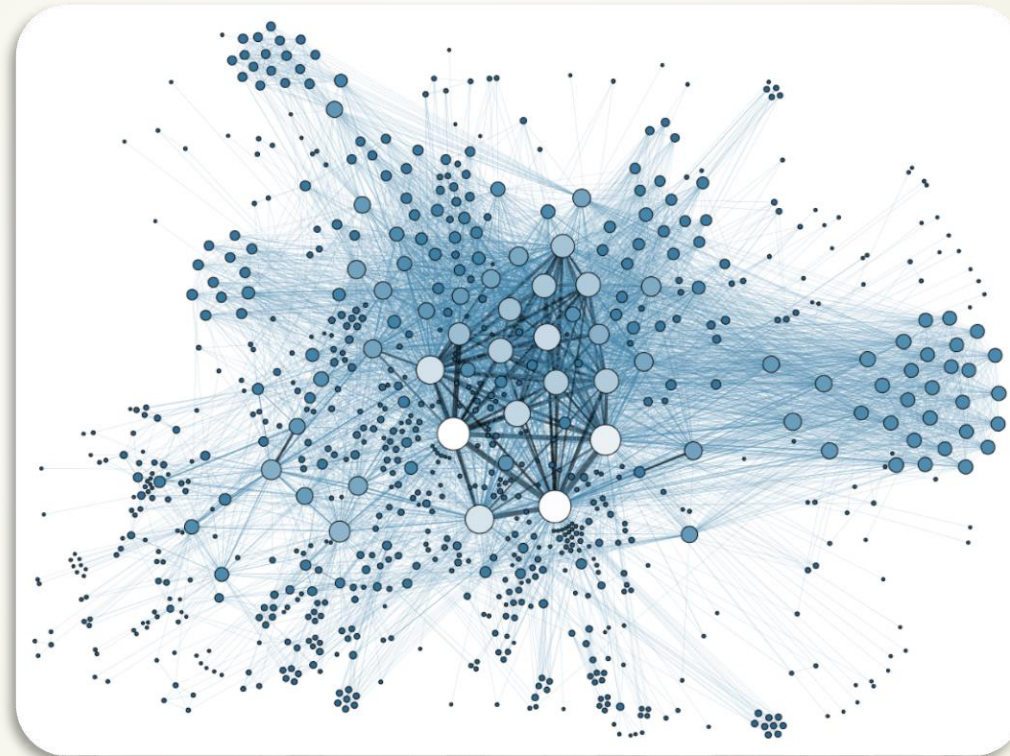
INTERNET TEHNOLOGIJE

Prof. dr Zoran Veličković, dipl. inž. el.

2022/23.

Prof. dr Zoran Veličković, dipl. inž. el.

Internet tehnologije



Grafika na Web-u i semantički Web

(10/11)

Sadržaj

- ▶ GRAFIKA NA WEB-U
 - ▶ BIT-MAPIRANA GRAFIKA U HTML 5
 - ▶ Oznaka Canvas
 - ▶ Canvas i JavaScript
 - ▶ Inicijalizacija Canvas-a
 - ▶ Canvas: Metode i svojstva za crtanje
 - ▶ Primer: Analogni sat i Canvas, W3C
 - ▶ VEKTORSKA GRAFIKA NA WEB-U - SVG
 - ▶ Primer SVG grafike
 - ▶ SVG grafika na V3Schools.com
 - ▶ WEB ČITAČI I GRAFIKA
 - ▶ Biblioteka Modernizr
- ▶ SEMANTIKA I WEB
 - ▶ Razvoj semantike na Webu
 - ▶ Semantika i Web 3.0
 - ▶ Povezani podaci na Webu: LinkedData
 - ▶ Prve Web App u Web 3.0
 - ▶ Semantika u HTML 4/HTML 5
 - ▶ Semantičke oznake sekcija
- ▶ WEB GRAFIKA I PROGRAMSKI JEZIK JAVA
 - ▶ Web i programski jezik Java i Bajt-kod
 - ▶ Javina Virtuelna Mašina JVM
 - ▶ Preuzimanje i izvršavanje apleta
 - ▶ Java Web Start – JAWS
- ▶ MULTIMEDIJE - AUDIO/VIDEO U HTML 5

Bit-mapirana grafika u HTML 5

- ▶ U **HTML 5** je za rad sa **BIT-MAPIRANIM** delom Web stranice uvedena oznaka **<canvas> ... </canvas>**.
- ▶ **MANIPULACIJA** ovim objektom, kao i sa bilo kojim drugim objektom na Web stranici, je omogućena **JavaScript**-om.
- ▶ Takođe, za rad sa ovim tipom objekta razvijen je specijalizovani **CANVAS API** (*engl.* Application Programming Interface) koji se **MOŽE POZVATI** iz JavaScripta.
- ▶ Pre rada sa ovim objektom na Web stranici, potrebno je definisati **LOKACIJU canvas**-a na Web stranici.
- ▶ Osnovni **CANVAS API** podržava **2D KONTEKST** koji programeru omogućava **CRTANJE RAZLIČITIH OBLIKA, RENDEROVANA TEKSTA** kao i **PRIKAZA BIT-MAPIRANIH** slika u prozoru Web čitača.
- ▶ Pored ostalog, **CANVAS API** omogućava:
 - ▶ **rotaciju slike**, gradijentno popunjavanje,
 - ▶ **transparentnost**, manipulaciju bitovima,
 - ▶ **različite tipove linija, krivih, površina, ...**

*3D grafika i Web

- Realizovan je i WEB API za prikaz 3D objekte kroz biblioteku **three.js**.
- Ova biblioteka je zasnovana na **WebGL**-u, odnosno, biblioteci **OpenGL**-u.
- Za izvođenje određenih operacija se koristi **GRAFIČKI PROCESOR** video kartice čime se rasterećuje CPU.
- Rad sa bibliotekom **three.js** izlazi izvan ovog kursa, a informacije se mogu naći na **www.threejs.org**.

Inicijalizacija Canvas-a

```
<!DOCTYPE html>
<html>
  <head>
    </title><Canvas API</title>
    <script src="canvas.js"> </script>
  </head>
  <body>
    <section id = "canvasbox">
      <canvas id = "mycanvas" width = "500" height = "300"> </canvas>
    </section>
    . . .
  </body>
</html>
```

Učitavanje **canvas.js** biblioteke.

Generiše se "prazna" bela pravougaona površina – platno slična onoj iz **<div>** elementa, ali sasvim druge namene – za formiranje **grafičkih** elemenata.

Oznaka **<canvas>** nema sopstvene attribute za crtanje već predstavlja samo **KONTEJNER** za grafičke elemente!

Dimenzije platna se definišu atributima **width** i **height**.

Canvas i JavaScript

- ▶ Za iscrtavanje u elementu `<canvas>` koriste se **JAVASCRIPT METODE**, ali pre svega, treba formirati **PROGRAMSKU REFERENCU** na grafički objekt.
- ▶ Izvršno okruženje i referenciranje **CANVAS OBJEKTA** se obavlja korišćenjem JavaScript metoda `getContext()` i `getElementById()`.

...

```
var canvas = document.getElementById("mycanvas");
```

```
if (canvas.getContext)
```

```
{
```

```
    var ctx = canvas.getContext("2d");
```

```
    ctx.fillStyle = "rgb(255,0,0)";
```

```
    ctx.fillRect(50,50,100,100);
```

```
    . . .
```

```
}
```

Referenca na **canvas** objekt

Provera podrške browser-a **canvas**-u (Mozilla). Bolje je koristiti svojstvo **canvas** iz biblioteke **Modernizr**.

Zadavanje konteksta u kome se želi iscrtavanje grafičkih objekata

Za crtanje u konterjneru **canvas** koristi se JavaScript metode!

Canvas: Metode i svojstva za crtanje (1)

SVOJSTVA	OBJAŠNENJE
<u>fillStyle</u>	Sets or returns the color, gradient, or pattern used to fill the drawing
<u>strokeStyle</u>	Sets or returns the color, gradient, or pattern used for strokes
<u>shadowColor</u>	Sets or returns the color to use for shadows
<u>shadowBlur</u>	Sets or returns the blur level for shadows
<u>shadowOffsetX</u>	Sets or returns the horizontal distance of the shadow from the shape
<u>shadowOffsetY</u>	Sets or returns the vertical distance of the shadow from the shape

METODA	OBJAŠNENJE
<u>createLinearGradient()</u>	Creates a linear gradient (to use on canvas content)
<u>createPattern()</u>	Repeats a specified element in the specified direction
<u>createRadialGradient()</u>	Creates a radial/circular gradient (to use on canvas content)
<u>addColorStop()</u>	Specifies the colors and stop positions in a gradient object

Canvas: Metode i svojstva za crtanje (2)

Rectangles

METODE	OBJAŠNENJE
<code>rect()</code>	Creates a rectangle
<code>fillRect()</code>	Draws a "filled" rectangle
<code>strokeRect()</code>	Draws a rectangle (no fill)
<code>clearRect()</code>	Clears the specified pixels within a given rectangle

Paths

METODE	OBJAŠNENJE
<code>fill()</code>	Fills the current drawing (path)
<code>stroke()</code>	Actually draws the path you have defined
<code>beginPath()</code>	Begins a path, or resets the current path
<code>moveTo()</code>	Moves the path to the specified point in the canvas, without creating a line
<code>closePath()</code>	Creates a path from the current point back to the starting point
<code>lineTo()</code>	Adds a new point and creates a line to that point from the last specified point in the canvas
<code>clip()</code>	Clips a region of any shape and size from the original canvas
<code>quadraticCurveTo()</code>	Creates a quadratic Bézier curve
<code>bezierCurveTo()</code>	Creates a cubic Bézier curve
<code>arc()</code>	Creates an arc/curve (used to create circles, or parts of circles)
<code>arcTo()</code>	Creates an arc/curve between two tangents

Canvas: Metode i svojstva za crtanje (3)



Transformations

METODE	OBJAŠNENJE
<code>scale()</code>	Scales the current drawing bigger or smaller
<code>rotate()</code>	Rotates the current drawing
<code>translate()</code>	Remaps the (0,0) position on the canvas
<code>transform()</code>	Replaces the current transformation matrix for the drawing
<code>setTransform()</code>	Resets the current transform to the identity matrix. Then runs <code>transform()</code>



Text

SVOJSTVA	OBJAŠNENJE
<code>font</code>	Sets or returns the current font properties for text content
<code>textAlign</code>	Sets or returns the current alignment for text content
<code>textBaseline</code>	Sets or returns the current text baseline used when drawing text

METODE	OBJAŠNENJE
<code>fillText()</code>	Draws "filled" text on the canvas
<code>strokeText()</code>	Draws text on the canvas (no fill)
<code>measureText()</code>	Returns an object that contains the width of the specified text

Canvas: Metode i svojstva za crtanje (4)



Pixel Manipulation

SVOJSTVA	OBJAŠNENJE
<code>width</code>	Returns the width of an ImageData object
<code>height</code>	Returns the height of an ImageData object
<code>data</code>	Returns an object that contains image data of a specified ImageData object

METODE	OBJAŠNENJE
<code>createImageData()</code>	Creates a new, blank ImageData object
<code>getImageData()</code>	Returns an ImageData object that copies the pixel data for the specified rectangle on a canvas
<code>putImageData()</code>	Puts the image data (from a specified ImageData object) back onto the canvas



Compositing

SVOJSTVA	OBJAŠNENJE
<code>globalAlpha</code>	Sets or returns the current alpha or transparency value of the drawing
<code>globalCompositeOperation</code>	Sets or returns how a new image are drawn onto an existing image



Other

METODE	OBJAŠNENJE
<code>save()</code>	Saves the state of the current context
<code>restore()</code>	Returns previously saved path state and attributes
<code>createEvent()</code>	
<code>getContext()</code>	
<code>toDataURL()</code>	

Analogni sat i Canvas

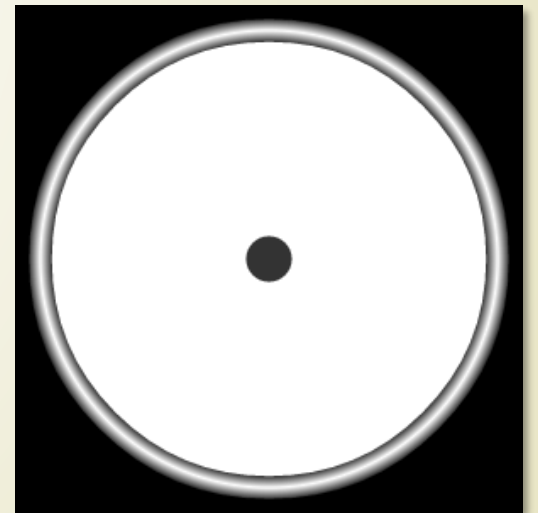
1

```
<!DOCTYPE html>
<html>
  <body>
    <canvas id="canvas" width="400" height="400" style="background-color:#333"> </canvas>
  <script>
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");
    var radius = canvas.height / 2;
    ctx.translate(radius, radius);
    radius = radius * 0.90;
    setInterval(drawClock, 1000);
    function drawClock() {
      drawFace(ctx, radius);
      drawNumbers(ctx, radius);
      drawTime(ctx, radius);
    }
  </script>
</body>
</html>
```

Analogni sat: drawFace

```
function drawFace(ctx, radius) {  
  var grad;  
  ctx.beginPath();  
  ctx.arc(0, 0, radius, 0, 2*Math.PI);  
  ctx.fillStyle = 'white';  
  ctx.fill();  
  grad = ctx.createRadialGradient(0,0,radius*0.95, 0,0,radius*1.05);  
  grad.addColorStop(0, '#333');  
  grad.addColorStop(0.5, 'white');  
  grad.addColorStop(1, '#333');  
  ctx.strokeStyle = grad;  
  ctx.lineWidth = radius*0.1;  
  ctx.stroke();  
  ctx.beginPath();  
  ctx.arc(0, 0, radius*0.1, 0, 2*Math.PI);  
  ctx.fillStyle = '#333';  
  ctx.fill();  
}
```

2

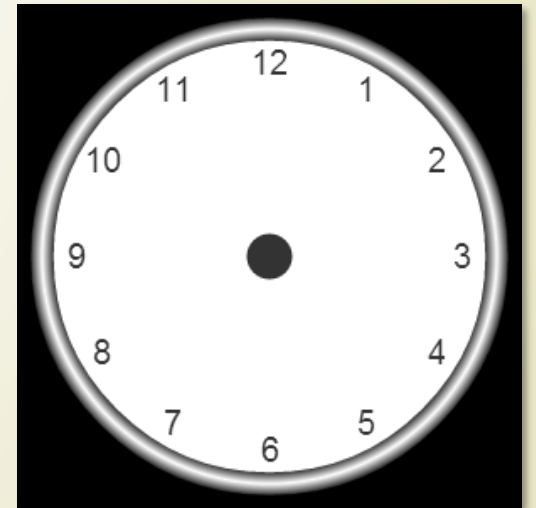


Analogni sat: drawNumbers

```
function drawFace(ctx, radius) {  
  var grad;  
  ctx.beginPath();  
  ctx.arc(0, 0, radius, 0, 2*Math.PI);  
  ctx.fillStyle = 'white';  
  ctx.fill();  
  grad = ctx.createRadialGradient(0,0,radius*0.95, 0,0,radius*1.05);  
  grad.addColorStop(0, '#333');  
  grad.addColorStop(0.5, 'white');  
  grad.addColorStop(1, '#333');  
  ctx.strokeStyle = grad;  
  ctx.lineWidth = radius*0.1;  
  ctx.stroke();  
  ctx.beginPath();  
  ctx.arc(0, 0, radius*0.1, 0, 2*Math.PI);  
  ctx.fillStyle = '#333';  
  ctx.fill();
```

```
}
```

3



Analogni sat: drawTime

```
function drawTime(ctx, radius) {
    var now = new Date();
    var hour = now.getHours();
    var minute = now.getMinutes();
    var second = now.getSeconds();

    // časovi
    hour=hour%12;
    hour=(hour*Math.PI/6)+(minute*Math.PI/(6*60))+(second*Math.PI/(360*60));
    drawHand(ctx, hour, radius*0.5, radius*0.07);

    // minuti
    minute=(minute*Math.PI/30)+(second*Math.PI/(30*60));
    drawHand(ctx, minute, radius*0.8, radius*0.07);

    // sekunde
    second=(second*Math.PI/30);
    drawHand(ctx, second, radius*0.9, radius*0.02);
}
```

Analogni sat: drawTime

```
function drawHand(ctx, pos, length, width)
```

```
{
```

```
  ctx.beginPath();
```

```
  ctx.lineWidth = width;
```

```
  ctx.lineCap = "round";
```

```
  ctx.moveTo(0,0);
```

```
  ctx.rotate(pos);
```

```
  ctx.lineTo(0, -length);
```

```
  ctx.stroke();
```

```
  ctx.rotate(-pos);
```

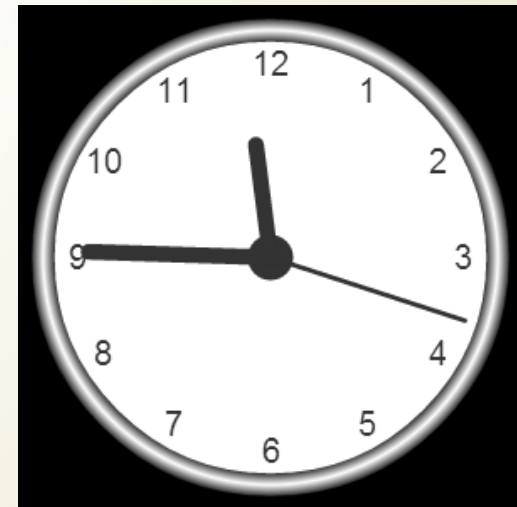
```
}
```

```
</script>
```

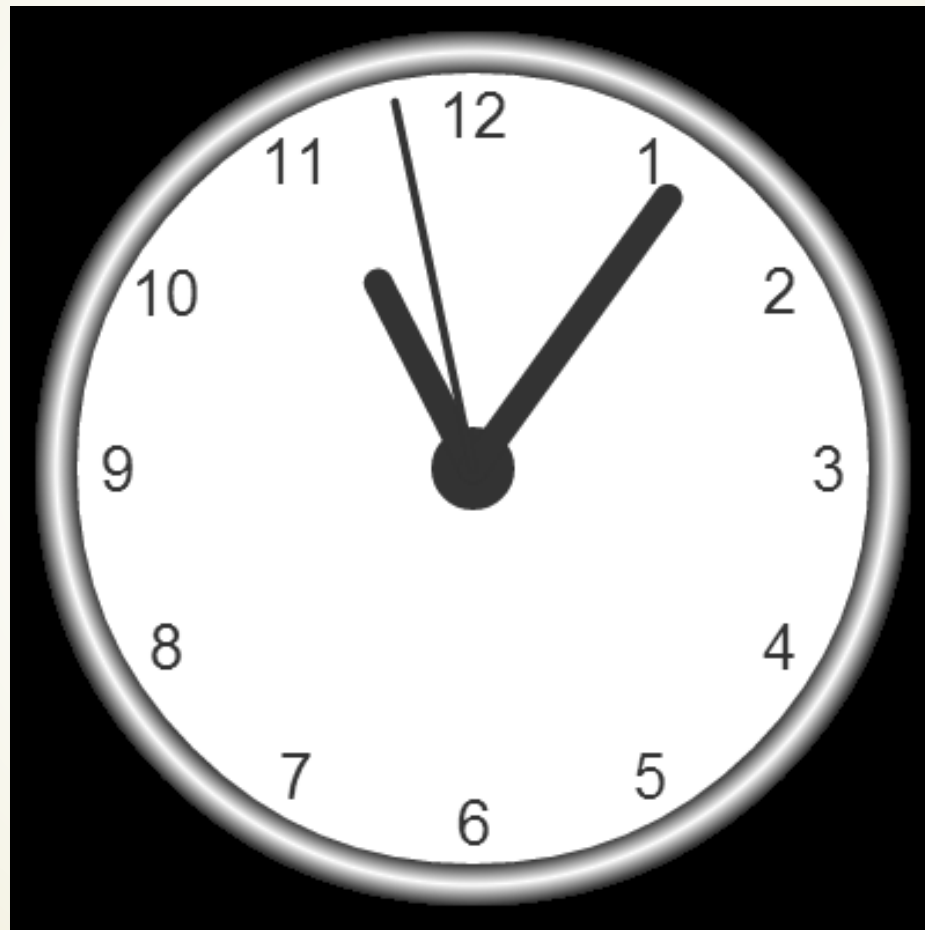
```
</body>
```

```
</html>
```

5



Analogni sat i Canvas, komplet



http://www.w3schools.com/canvas/canvas_clock_start.asp

Vektorska grafika na Web-u SVG

- Na prethodnim slajdovima je pokazano da je za iscrtavanje grafike u **CANVAS**-u je **POTREBAN JavaScript**.
- Da li je moguće formirati **NOVE HTML OZNAKE** za ove potrebe?
- Nova oznaka koja može poslužiti ovoj svrsi je **<svg>...</svg>** sa nizom **NOVIH** (**viewBox**, **style**, ...) i **STARIH atributa** (**width**, **height**).
- **SVG** (engl. **Scalable Vector Graphics**) je zapravo **XML jezik – REČNIK** za prikazivanje **VEKTORSKE GRAFIKE** (navesti osnovne karakteristike vektorske grafike).
- **SVG** poseduje **PREDEFINISANE** HTML oznake - forme koje se mogu po potrebi koristiti:
 - **Rectangle** (pravougaonik): **<rect>**
 - **Circle** (krug): **<circle>**
 - **Ellipse** (elipsa): **<ellipse>**
 - **Line** (linija): **<line>**
 - **Polyline**: **<polyline>**
 - **Polygon**: **<polygon>**

SVG grafika izlazi izvan opsega ovog predmeta.

Primer SVG grafike

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head> <title>SVG example 2</title> </head>
```

```
  <body>
```

```
    <svg id="mysvg" viewBox = "0 0 320 240" style = "outline: 1px solid #999;  
                                                    width: 320px; height: 240px;">
```

```
      <rect x="50" y="50" width="100" height="100" style="fill: rgb(255,0,0)"> </rect>
```

```
      <line x1="50" y1="50" x2="150" y2="150" style="stroke:  
                                                    rgb(0,127,127); stroke-width: 5;">
```

```
    </line>
```

```
  </svg>
```

```
</body>
```

```
</html>
```

Definisanje **prozora** SVG grafike – atribu **viewBox**

Crtanje pravougaonika i linije

SVG grafika na W3Schools.com

W3Schools Online Web Tut x | HTML5 SVG x | SVG Tutorial x +

https://www.w3schools.com/graphics/svg_intro.asp

HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP HOW TO JQUERY W3.CSS PYTHON XML MORE REFERENCES EXAMPLES FORUM

HTML Graphics
Graphics HOME

Google Maps
Maps Intro
Maps Basic
Maps Overlays
Maps Events
Maps Controls
Maps Types
Maps Reference

SVG Tutorial
SVG Intro
SVG in HTML5
SVG Rectangle
SVG Circle
SVG Ellipse
SVG Line
SVG Polygon
SVG Polyline
SVG Path
SVG Text
SVG Stroking
SVG Filters Intro
SVG Blur Effects
SVG Drop Shadows
SVG Linear
SVG Radial
SVG Examples
SVG Reference

< Previous

Next >

SVG stands for Scalable Vector Graphics.
SVG defines vector-based graphics in XML format.

Examples in Each Chapter

With our "Try it Yourself" editor, you can edit the SVG, and click on a button to view the result.

SVG Example

```
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```

Try it Yourself »

COLOR PICKER

HOW TO

- Tabs
- Dropdowns
- Accordions
- Side Navigation
- Top Navigation
- Modal Boxes

What you should already know

SVG grafika: putanja - path

W3Schools Online Web Tutorials x SVG Path x +

w3schools.com/graphics/svg_path.asp

w3schools.com THE WORLD'S LARGEST WEB DEVELOPER SITE

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP HOW TO W3.CSS JQUERY XML MORE ▾ REFERENCES ▾ EXERCISES ▾ FORUM

SVG Tutorial

- SVG Intro
- SVG in HTML5
- SVG Rectangle
- SVG Circle
- SVG Ellipse
- SVG Line
- SVG Polygon
- SVG Polyline
- SVG Path**
- SVG Text
- SVG Stroking
- SVG Filters Intro
- SVG Blur Effects
- SVG Drop Shadows
- SVG Linear
- SVG Radial
- SVG Examples
- SVG Reference

Canvas Tutorial

- Canvas Intro
- Canvas Drawing
- Canvas Coordinates
- Canvas Gradients
- Canvas Text
- Canvas Images
- Canvas Reference

Canvas Clock

- Clock Intro
- Clock Face
- Clock Numbers
- Clock Hands

Get Certified in HTML, CSS, and JavaScript [Get Certified >](#)

SVG <path>

[< Previous](#) [Next >](#)

SVG Path - <path>

The <path> element is used to define a path.


The following commands are available for path data:

- M = moveto
- L = lineto
- H = horizontal lineto
- V = vertical lineto
- C = curveto
- S = smooth curveto
- Q = quadratic Bézier curve
- T = smooth quadratic Bézier curveto
- A = elliptical Arc
- Z = closepath

Note: All of the commands above can also be expressed with lower letters. Capital letters means absolutely positioned, lower cases means relatively positioned.

Example 1

The example below defines a path that starts at position 150,0 with a line to position 75,200 then from there, a line to 225,200 and finally closing the path back to 150,0:



VEEAM

#1 Windows Backup & Recovery

Trusted by more than 350k customers worldwide. Try a Free 30-day Trial!

Primer SVG grafike



```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<!-- Generator: Adobe Illustrator 10.0, SVG Export Plug-In . SVG Version: 3.0.0 Build 76) -->
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="193" height="34" viewBox="-1 -1 191 33">
<path fill="#F00" d="M22.477,29.654
c-2.515-1.575-4.367-7.011-4.367-13.469c0-0.237,0.001-0.471,0.007-0.704l0.003,0.003c-6.994-1.099-12.048-4.111-12.177-7.672
L5.922,7.853C4.21,9.98,3.233,12.439,3.233,15.059c0,7.71,8.463,14.027,19.198,14.586L22.477,29.654z M0,15.622
C0,6.995,10.736,0,23.98,0c13.244,0,23.98,6.995,23.98,15.622S37.224,31.244,23.98,31.244C10.736,31.244,0,24.249,0,15.622z
M23.98,2.254c2.557,0,4.731,3.878,5.537,9.291l0.003-0.008c4.974-0.774,8.457-2.561,8.457-4.639c0-2.79-6.267-5.048-13.997-5.048
c-7.73,0-13.997,2.259-13.997,5.048c0,2.078,3.482,3.864,8.457,4.639l0.003,0.008C19.249,6.133,21.424,2.254,23.98,2.254z
M25.528,29.645c10.735-0.559,19.199-6.876,19.199-14.586c0-2.62-0.978-5.079-2.689-7.206l-0.021-0.041
c-0.129,3.561-5.182,6.573-12.177,7.672l0.003-0.003c0.005,0.233,0.007,0.467,0.007,0.704c0,6.458-1.854,11.894-4.367,13.469
L25.528,29.645z M23.98,11.946c1.146,0,2.259-0.049,3.324-0.143l-0.001,0.001c-0.53-3.469-1.818-5.917-3.323-5.917
s-2.792,2.448-3.323,5.917l-0.001-0.001C21.721,11.897,22.834,11.946,23.98,11.946z M23.98,24.792c1.925,0,3.495-4.006,3.579-9.03
l0.01,0.003c-1.16,0.107-2.36,0.164-3.589,0.164c-1.229,0-2.429-0.056-3.589-0.164l0.011-0.003
C20.485,20.787,22.055,24.792,23.98,24.792z"/>
<path fill="#F00" d="M122.362,15.622
c0-6.594,5.346-11.939,11.939-11.939c6.596,0,11.941,5.346,11.941,11.939c0,6.595-5.346,11.939-11.941,11.939
C127.708,27.561,122.362,22.216,122.362,15.622z M134.301,23.183c3.683,0,6.668-3.386,6.668-7.562
c0-4.175-2.985-7.562-6.668-7.562c-3.681,0-6.665,3.386-6.665,7.562C127.636,19.797,130.621,23.183,134.301,23.183z"/>
<path fill="#F00" d="M159.79,8.657v18.362h-5.03V8.657h0.017h-7.035V4.505h19.066v4.151h-7H159.79z"/>
<path fill="#F00" d="M172.893,21.954l-0.005,0.005l-1.93,5.061
h-5.875l9.288-22.514h3.132h3.129l9.289,22.514h-5.876l-1.93-5.061l-0.005-0.005h-4.607H172.893z M177.503,18.154h3.165
l-0.002,0.001l-3.142-8.24h-0.044l-3.144,8.24l-0.001-0.001H177.503z"/>
<path fill="#F00" d="M67.057,8.657v18.362h-5.031V8.657h0.017h-7.036V4.505h19.067v4.151h-7H67.057z"/>
<path fill="#F00" d="M75.362,15.622
c0-6.594,5.346-11.939,11.941-11.939c6.594,0,11.94,5.346,11.94,11.939c0,6.595-5.346,11.939-11.94,11.939
C80.708,27.561,75.362,22.216,75.362,15.622z M87.303,23.183c3.681,0,6.667-3.386,6.667-7.562c0-4.175-2.985-7.562-6.667-7.562
c-3.682,0-6.667,3.386-6.667,7.562C80.636,19.797,83.621,23.183,87.303,23.183z"/>
<path fill="#F00" d="M110.802,27.019h2.551v-8.443V18.47
18.776-13.965h-5.874l-5.453,9.463l-5.452-9.463h-5.876l8.777,13.965v0.105v8.443H110.802z"/>
</svg>
```

Web čitači i grafika

- ▶ **NEKOMPATIBILNOST** Web čitača u praksi često može izazavati **NEŽELJEN IZGLED**, odnosno, ponašanje Web aplikacije.
- ▶ Dobro je znati **KOJE SU OD FUNKCIJA PODRŽANE** od strane browsera, a koje **NISU**, kako bi se programski kod **PRILAGODIO** mogućnošću korisnikovog Web čitača.
- ▶ Otkrivanje **PODRŽANIH FUNKCIJA WEB ČITAČA** se pojednostavljuje korišćenjem JavaScript biblioteke Modernizr.
- ▶ Korišćenje funkcija iz biblioteke **Modernizr** se **PREPORUČUJE ZA BUDUĆE** APLIKACIJE.
- ▶ Tako se na primer, može koristiti biblioteka **Modernizer** ZA **OTKRIVANJE PODRŠKE** funkcije **canvas** elementu.
- ▶ Za svaku funkciju koja se detektuje, čuva se **BULOVA PROMENLJIVA** u objektu **Modernizr** koji se formira od **REZULTATA TESTIRANJA** podrške.
- ▶ U sledećem primeru se testira **PODRŠKA CANVASU** putem Modernizr.canvas svojstva.

JavaScript biblioteka Modernizr

```
<!DOCTYPE html>
<html>
  <head>
    <script src="http://code.jquery.com/jquery-1.6.1.js"></script>
    <!--Veza sa bibliotekom Modernizr korisnom za detektovanje funkcija podrške -->
    <script src="http://ajax.cdnjs.com/ajax/libs/modernizr/1.7/modernizr-1.7.min.js"> </script>
    <script type="text/javascript">
      if(Modernizr.canvas){
        // Ovde se postavlja kod koji zahteva podršku canvas elementa
        alert ("Vaš Web čitač podržava element canvas");
      } else {
        alert ("Vaš Web čitač ne podržava element canvas");
        // Web čitač ne podržava canvas elemente
        // treba upotrebiti alternativno rešenje
      }
    </script>
  </head>
  <body>
    ...
  </body>
</html>
```

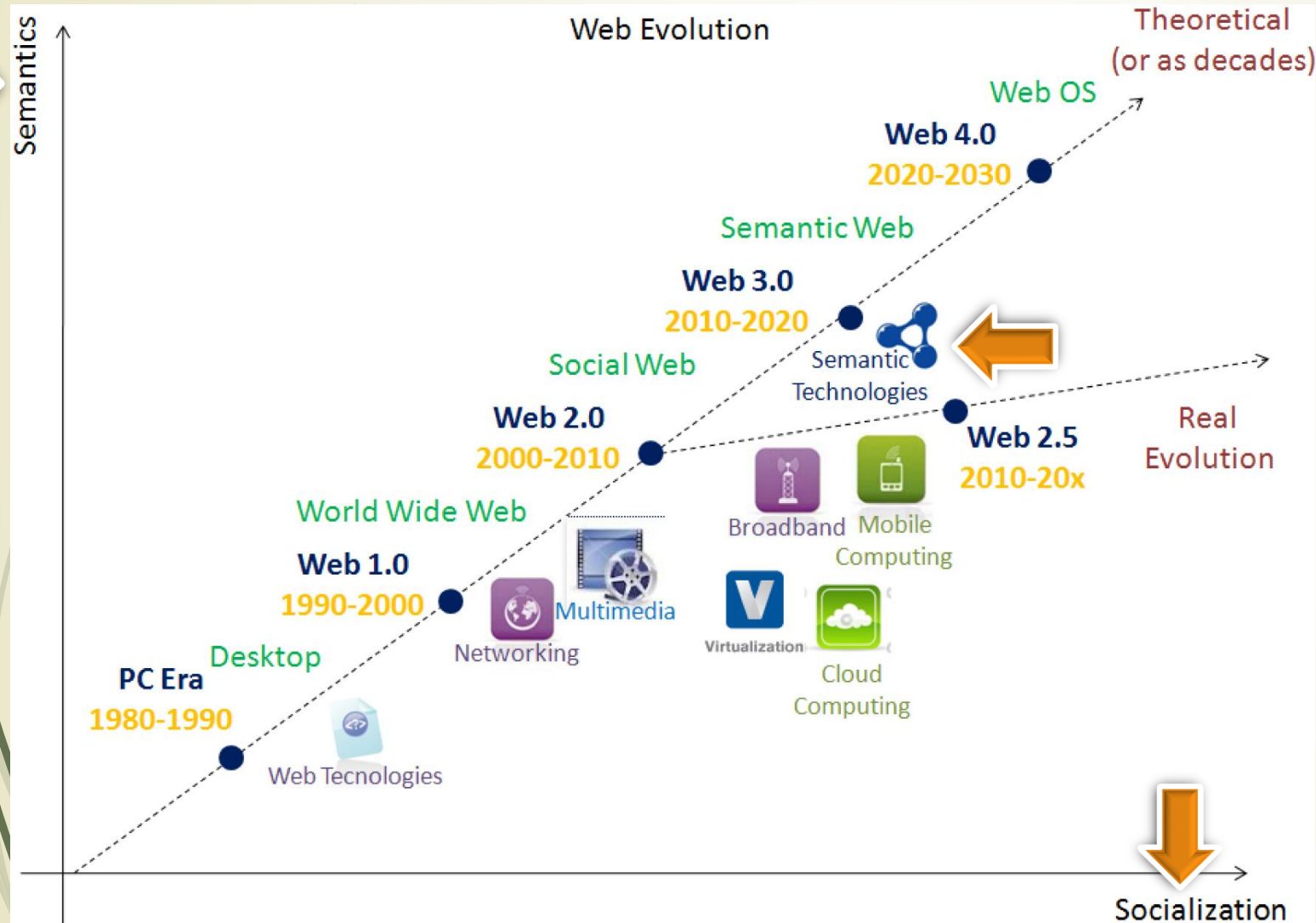
Učitavanje JS
biblioteka jQuery i
Modernizr

Ispitivanje podrške
Web čitača za
canvas element

Semantika na Web-u

- ▶ Da bi se Web dokument pravilno interpretirao, a potom i prikazao, potrebno ga je **DEKLARISATI** na samom početku (primer oznaka `<!DOCTYPE>`).
- ▶ Elementi **HTML 5** su načinjeni tako da predstavljaju **KONKRETNE OBJEKTE IZ ŽIVOTA** kako bi dokument bio **ČITLJIVIJI** (čitaj samoopisujući).
- ▶ Na osnovu **NAZIVA HTML 5 OZNAKA** jasno se mogu **PREPOZNATI** delovi dokumenta koji predstavljaju paragraf, sliku ili zaglavlje (preuzeta ideja od XML-a).
- ▶ Ova osobina HTML 5 se naziva **SEMANTIKOM**.
- ▶ Međutim, opis dokumenata na ovaj način je često **NEPOGODAN**, naročito kada sadrži detalje o tome **KAKO** Web čitač treba da prikaže sadržaj!
- ▶ Problem je naročito izražen prilikom prikaza Web sadržaja na **MOBILNIM UREĐAJIMA** jer oni **NE MOGU** korektno da primene ove zahteve.

Razvoj semantike na Webu

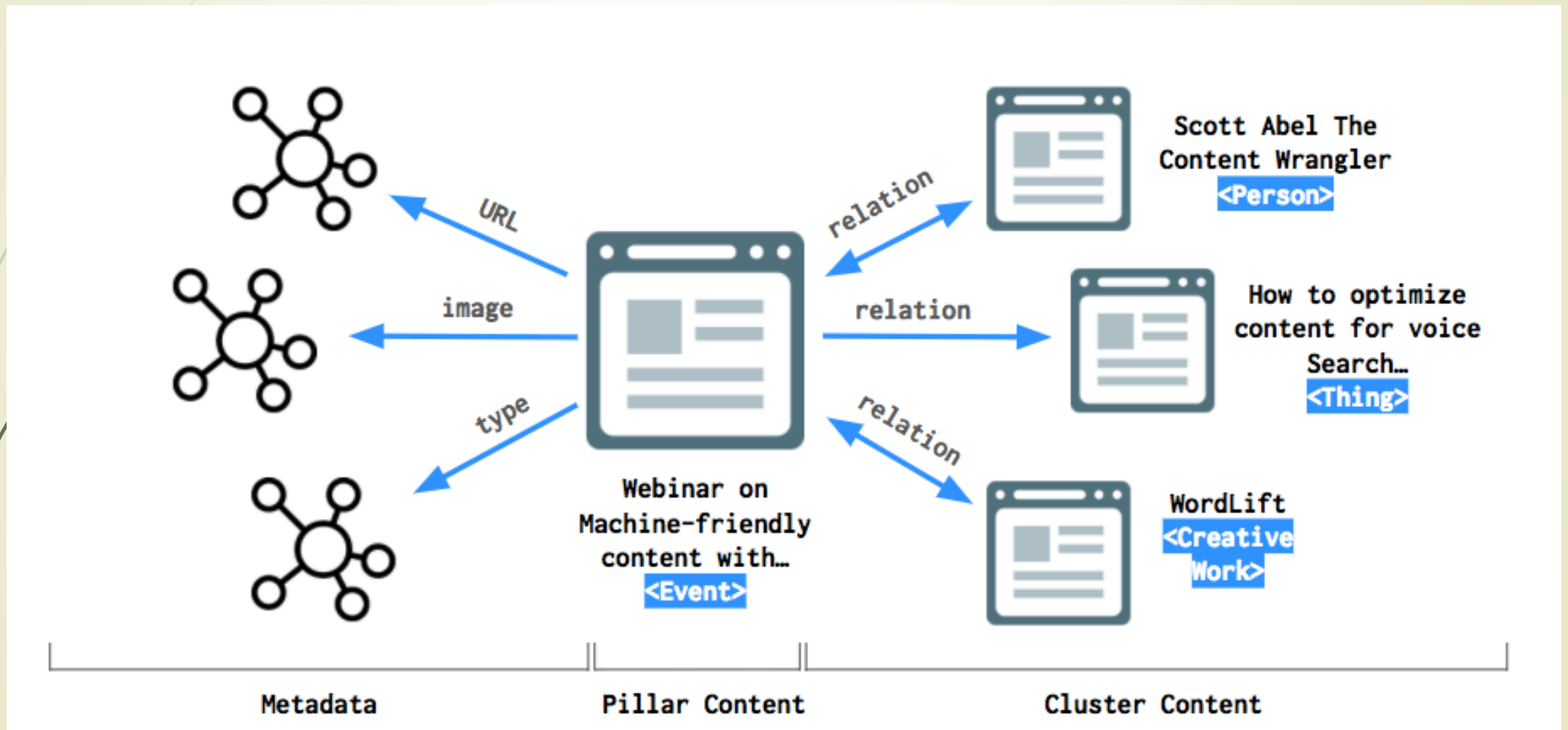


	Web 2.0 apps	Web 3.0 apps (powered by blockchain)
Browser		brave
Storage		IPFS
Video and audio calls		EXPERTY
Operating system		
Social network		
Messaging		status
Remote job	Up	Ethlance

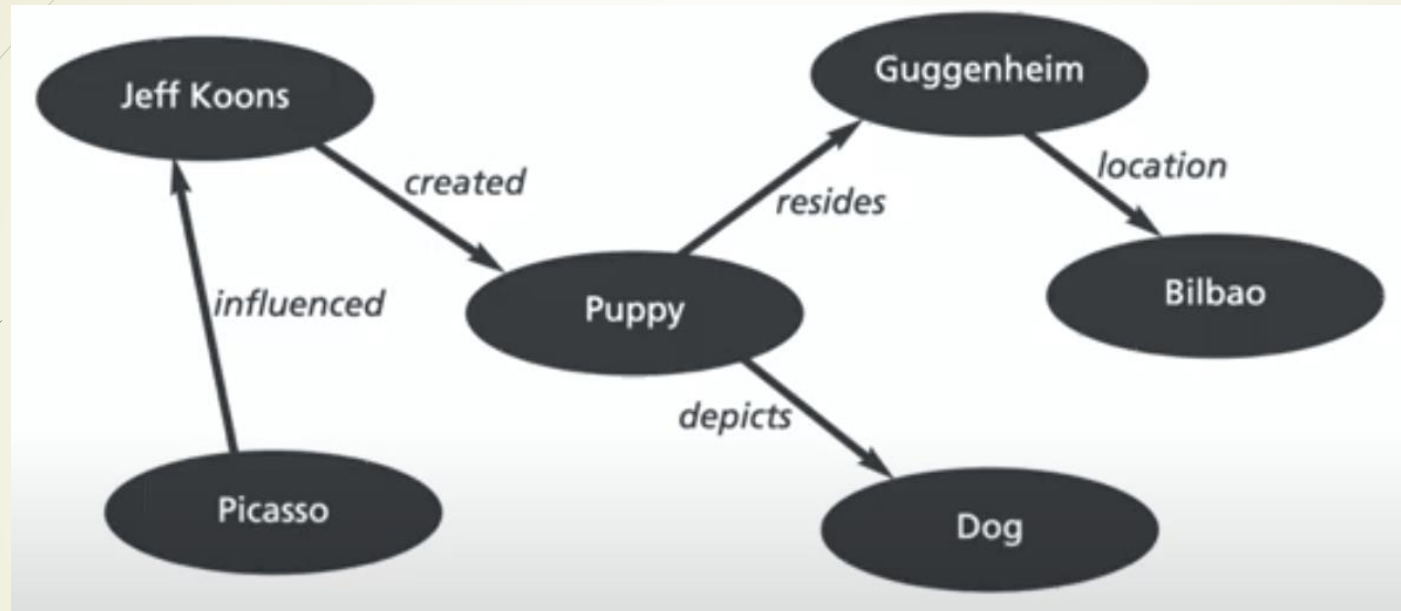
Semantika i Web 3.0

- **Web 3.0** se naziva i **SEMANTIČKIM** Web-om.
- U Web 3.0 eri, **RAČUNARI** su sposobni da **TUMAČE INFORMACIJE** poput ljudi korišćenjem:
 - **Veštačke inteligencije,**
 - **Blockchain-a,** i
 - **Mašinskog učenja.**
- Ove tehnologije omogućavaju da se **INTELIGENTNO GENERIŠE** i **DISTRIBUIRA** koristan sadržaj prilagođen **POSEBNIM POTREBAMA KORISNIKA**.
- Nove tehnologije Web 3.0 omogućavaju da **RAČUNARI** mogu da **RAZUMEJU ZNAČENJE INFORMACIJA**, i samim tim, da **NAUČE** ono što zanima korisnika.
- Ovo može pomoći korisniku da **LAKŠE PRONAĐE ŽELJENE INFORMACIJE** na Web-u.
- Mogu se **PREPOZNTI**: ljudi, mesta, događaji, kompanije, proizvodi, filmovi itd.
- Mogu se **RAZUMETI ODNOSI** između ljudi i stvari.

Povezani podatki na Webu: LinkedData



Povezivanje podataka na Webu (1)



- ▶ TED - https://www.youtube.com/watch?v=OM6XIIcm_qo , Tim Berners-Lee
- ▶ https://www.youtube.com/watch?v=qg8e01_kFTo , Semantic Web
- ▶ <https://www.youtube.com/watch?v=S4ie65FpaN4> , Web 3.0

Prve Web App u Web 3.0

► **Meta**

- Facebook se preimenovao u „Meta“ jer želi da razume ponašanje ljudi.
- Ako mogu da razumeju ljudsko ponašanje, onda mogu da stvore "Metaverz" - imaginarni svet uz korišćenje VR i AR tehnologije.
- Avatarima se omogućava da se druže, komuniciraju i kreću slobodno bez ograničenja.

► **Siri**

- Posедуje ugrađene funkcije glasovne kontrole da obavlja zadatke kao što su telefonski pozivi, zakazivanje sastanaka i pretraživanje interneta.
- Obavlja fokusirane pretrage Interneta, zakazivanje sastanaka i slično.
- Virtuelni asistenti (Amazonov Alexa, Cortana, ...) nastavljaju da se razvijaju i postaju interaktivniji.

► **Brave Browser, Wolfram Alpha, ...**

Semantika u HTML 4/HTML 5

- ▶ HTML 4 poseduje samo **DVA ATRIBUTA** za **SEMANTIČKI OPIS** HTML elemenata, a to su Vama već dobro poznati atributi: "**id**" i "**class**".
- ▶ Atribut "**id**" je **JEDINSTVENI IDENTIFIKATOR**, koji svojom vrednošću može dodatno da opiše značaj elementa kome je dodeljen, dakle, poseduje **SEMANTIČKU VREDNOST**.
- ▶ Atribut "**class**" **NIJE** jedinstven identifikator, ali obezbeđuje da se **VIŠE KLASA** mogu primeniti na **JEDAN** HTML element. Pogledajte sledeći primer:

HTML	ZNAČENJE
<code><p></code>	Paragraf
<code><p id = "author"></code>	Paragraf koji predstavlja pojedinog autora
<code><p class = "bio"></code>	Paragraf koji predstavlja biografiju
<code><p class = "author bio"></code>	Paragraf koji predstavlja biografiju autora

Istraživanje: najčešće vrednosti atributa `class` i `id`

Izvedene HTML 5 oznake

Najkorišćeniji "id" atributi HTML 4

Najkorišćeniji "class" atributi HTML 4

CLSIDD27CD
FOOTER
BTAMARKER
HEADER
CONTENT
AREATITLE
LAYOUT
NOBULLETCONTENT
SQBULLETCONTENT
LOGO
SEARCH
MAIN
BANNER
CONTAINER
TOP
SIDEBAR
NAV
WRAPPER
+
LEFT

HEADER
BANNER
TITLE
HEADER
TOP

ARTICLE / SECTION
CONTENT CONTAINER
TEXT BODY MAIN
MAIN CONTENT

MENU
MENU

TIME
DATE

ASIDE
SIDEBAR
LEFT

NAV
NAV
NAV

INPUT
TYPE=SEARCH
SEARCH

FOOTER
LOGO
FOOTER
FOOTER

SMALL
COPYRIGHT
SMALL
SMALLTEXT

FOOTER
MENU
TITLE
SMALL
TEXT
CONTENT
HEADER
NAV
COPYRIGHT
BUTTON
MAIN
SEARCH
MSONORMAL
DATE
SMALLTEXT
BODY
STYLE1
TOP
WHITE
LINK



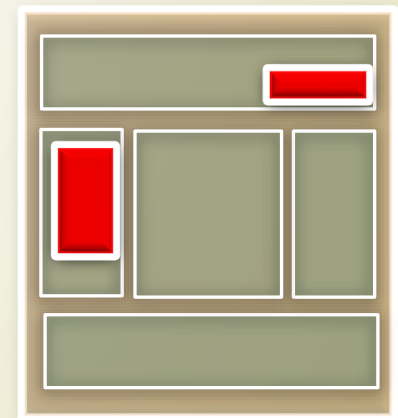
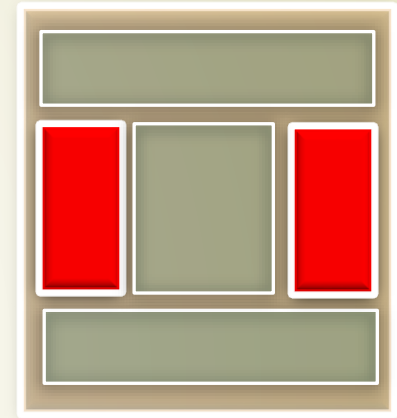
Nove HTML 5 oznake sekcija

- ▶ **HTML 5** dodaje dva nova elementa za **DEFINISANJE STRUKTURE** Web stranice - **STRUKTURNE OZNAKE** <section> i <article>.
- ▶ Oznaka **<section>** se koristi za **TEMATSKO GRUPISANJE** sadržaja.
- ▶ Oznaka **<article>** je dizajnirana za distribuciju i **PONOVNO KORIŠĆENJE**.
- ▶ HTML elementi koji se koriste za **IMPLICITNO STRUKTUIRANJE** dokumenata su: <header> i <hgroup>.
- ▶ Oznaka **<header>** se obično locira na vrh Web dokumenta, i sadrži **GLAVNO ZAGLAVLJE DOKUMENTA** i vrlo često **NAVIGACIJU** sa prozorom za pretraživanje sajta.
- ▶ HTML oznaka **<hgroup>** se koristi pri postojanju **PODNASLOVA**.
- ▶ Ove oznake su značajne za algoritme koji **GENERIŠU SADRŽAJE** Web stranica.
- ▶ Iako se sličan efekt može postići i u HTML 4, nove oznake **STANDARDIZUJU** ovaj posao (**Wikipedija** je dobar primer).

```
<sup></sup> <sub></sub> <small></small> <code></code> <pre></pre> {{цитат|}}  
<ref></ref> <ref name="" /> {{reflist|}} {{efn|name=|}} {{notelist|}} <includeonly></includeonly>
```

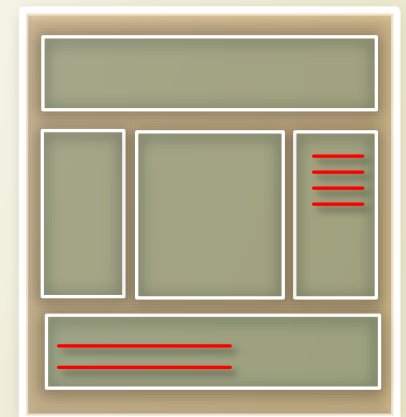
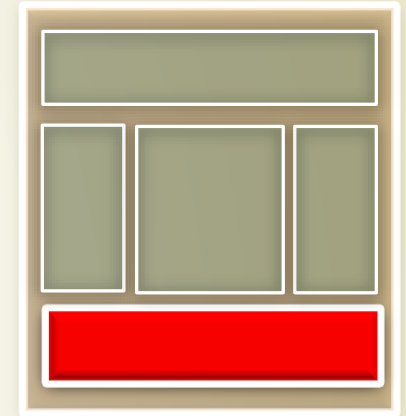
HTML 5: Semantičke oznake sekcija (1)

- ▶ HTML 5 element **<aside>** se koristi za prikaz tekstova koji NISU DIREKTNO VEZANI za sam tekući tekst, ali pomažu njegovom indirektnom razumevanju.
- ▶ Obično se u ovim elementima prikazuju definicije, istorijski razvoj, navigacija ...
- ▶ HTML 5 element **<nav>** se koristi za navigaciju na sajtu.
- ▶ Mogu se koristiti VIŠE NAV OZNAKA na Web stranici.
- ▶ Obično se u zaglavlju dokumenta smašta **GLAVNI NAVIGACIONI MENI**, dok se u **<aside>** elementu smeštaju **LOKALNI** menii.



HTML 5: Semantičke oznake sekcija (2)

- ▶ HTML oznaka **<footer>** se generalno stavlja na kraj HTML dokumenta, sekcije ili artikla.
- ▶ Kao i **<header>** oznaka najčešće sadrži metainformacije (detalji o autoru, uslovi korišćenja ili linkove ka drugim sadržajima).
- ▶ Može sadržavati **<section>** elemente.
- ▶ HTML element **<small>** se nalazi u **<footer>** ili **<aside>** elementima (sadrži zakonske regulative, štampanje, ...).
- ▶ Nije namenjen da načini tekst manjim!
- ▶ Stilovi za prikaz se mogu slobodno koristiti.



Sve semantičke oznake sekcija u HTML 5

OZNAKA	OPIS
<code><article></code>	Defines an article
<code><aside></code>	Defines content aside from the page content
<code><details></code>	Defines additional details that the user can view or hide
<code><figcaption></code>	Defines a caption for a <code><figure></code> element
<code><figure></code>	Specifies self-contained content, like illustrations, diagrams, photos, code listing
<code><footer></code>	Defines a footer for a document or section
<code><header></code>	Specifies a header for a document or section
<code><main></code>	Specifies the main content of a document
<code><mark></code>	Defines marked/highlighted text
<code><nav></code>	Defines navigation links
<code><section></code>	Defines a section in a document
<code><summary></code>	Defines a visible heading for a <code><details></code> element
<code><time></code>	Defines a date/time



Grafičke aplikacije i Java

- ▶ Za razvoj visokokvalitetnih aplikacija za Web neophodno je obezbediti upravljanje **MULTIMEDIJALNIM SADRŽAJIMA** (grafičkim elementima i zvukom) ili pristup **UDALJENIM PODACIMA**.
- ▶ S obzirom na to da HTML **NIJE IMAO ADEKVATNU PODRŠKU** za ove potrebe, u prošlosti je iskorišćen programski jezik **JAVA** i njegova **PLATFORMSKA NEZAVISNOST**.
- ▶ Na programskom jeziku Java, **APLIKACIJE** za Web se nazivaju **APLETI** (engl. *applet*).
- ▶ Upotrebom HTML oznaka **<applet>**, **<object>** ili **<embed>**, programski kod **JAVA APLET**-a se integriše – postavlja na Web stranicu.
- ▶ Postupak **POSTAVLJANJA APPLETA** na Web stranicu je **SLIČAN** POSTAVLJANJU SLIKE na Web stranicu korišćenjem oznake ****.

Platformaska nezavisnost i Bajt-kod (1)

- ▶ Kada Web čitač naiđe na oznaku **<applet>**, **<object>** ili **<embed>** na HTML stranici, on će sa servera **1. PREUZETI** i **2. IZVRŠITI** programski kod aplet-a.
- ▶ Glavna osobina Java apleta je **PLATFORMSKA NEZAVISNOST**, odnosno, sposobnost da se **JEDAN TE ISTI** aplet jednako dobro izvršava i na **Windows** i na **Linux**- platformama.
- ▶ Kako obezbediti **PLATFORMSKA NEZAVISNOST**? Osnove slede u nastavku.
- ▶ Dakle, kako bi se postigla platformaska nezavisnost, **APLETI** ne sadrže programski kod **SPECIFIČAN ZA PROCESORE**, već GENERIČKI SKUP NAREDBI (u ***.class** fajlu) koji se naziva - **BAJT-KOD** (ili među-kod).

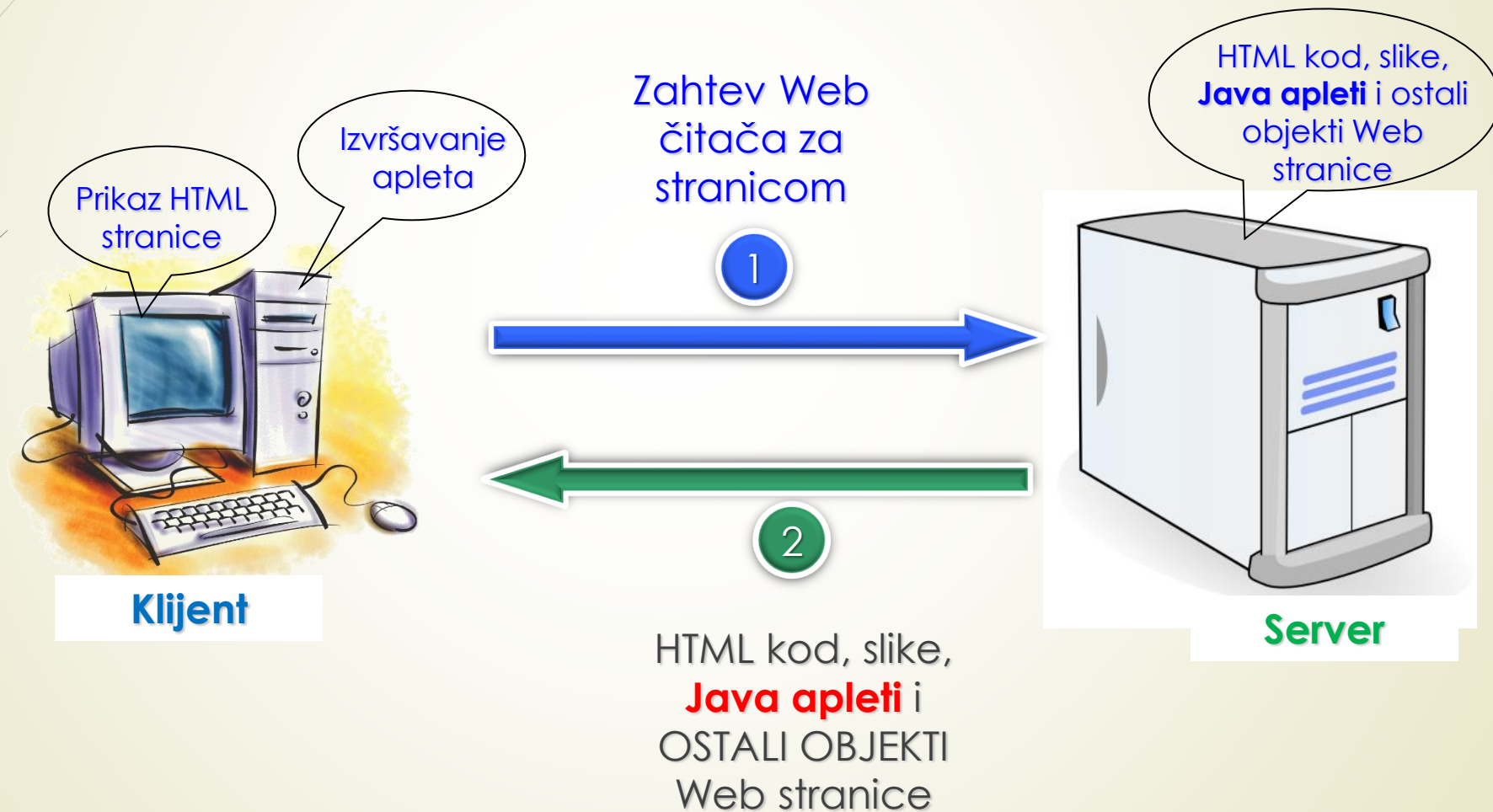
Platformaska nezavisnost i Bajt-kod (2)

- **PO PREUZIMANJU APLETA**, generički - **BAJT-KOD** se **PREVODI U KOD ODREDIŠNOG CPU-a** (ovaj princip je obezbedio platformsku nezavisnost).
- Korišćenje **BAJT-KODA**, koji je **ISTI** za **SVE PLATFORME**, obezbeđuje podršku **RAZLIČITIM** sistemskim platformama.
- **WEB ČITAČ** mora da **PREVEDE PROGRAMSKI MEĐU-KOD APLETA** u **PROGRAMSKI KOD** specifičan za korisnički procesor **SVAKI PUT** kada se aplikacija pokrene, što izaziva **KAŠNJENJE** u pokretanju programa.
- Kôd koji čitač Weba generiše je **MANJE KVALITETAN** nego onaj koji generiše kompajler programskog jezika!
- Međutim, programeri su spremni da plate ovu cenu zarad izvršavanja **ISTOG KODA** na **RAZLIČITIM** platformama!

Javina virtuelna mašina - JVM

- **JAVA APLETI** se **IZVRŠAVAJU SPORIJE** nego programi napisani (i kompajlirani) specijalizovanim programskim jezikom za određeni procesor.
- Svaki put kada korisnik potražuje aplet, čitač **MORA DA GA PREUZME**.
- Za razliku od **ActiveX** Web aplikacija, apleti se **SVAKI PUT** MORAJU PREUZETI sa servera!
- Java apleti se **POKREĆU** i **IZVRŠAVAJU** samo u okviru čitača!
- Ovaj koncept **ELIMINIŠE UTICAJ VIRUSA** i krađe podataka.
- Java apleti **NE MOGU** da pristupe resursima koji se nalaze na korisnikovom disku.
- **POSEBAN SOFTVER** (ugrađen u sam Web čitač) koji ima zadatak DA IZVRŠAVA APLETE naziva se **JAVINA VIRTUELNA MAŠINA - JVM** .

Preuzimanje i izvršavanje **apleta**

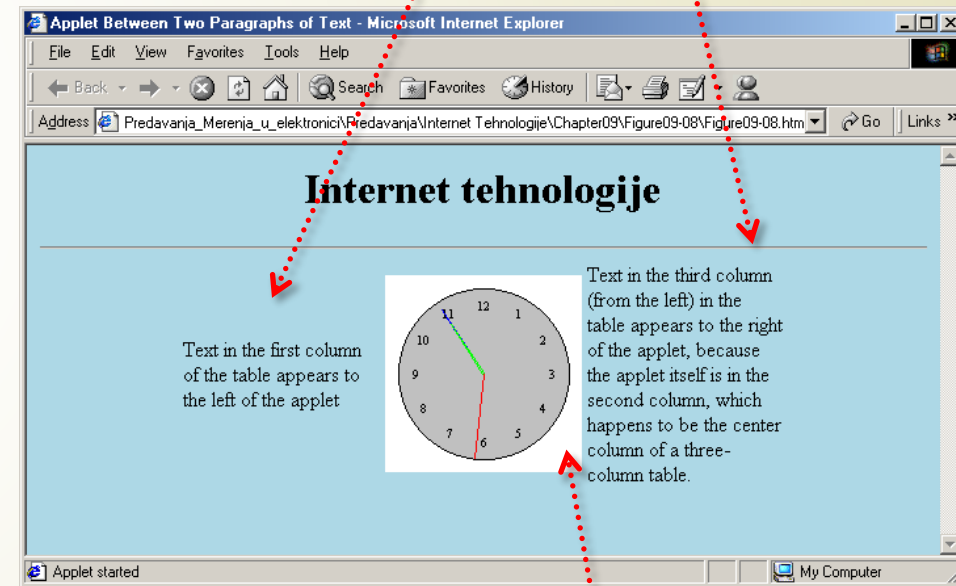


Ugrađivanje i izvršavanje apleta

- ▶ Kada browser naiđe na oznaku **<applet>** ili **<object>** u HTML kodu, predaje aplet Javinoj virtuelnoj mašini (JVM), koja ga:
 1. **Preuzima**,
 2. **Prevodi** bajt-kod u kod specifičan za ciljni procesor i
 3. **Izvršava** aplet.
- ▶ Da bi se ugradio aplet na Web stranicu, koristi se početna i završna (zastarela) oznaka za aplet **<applet> ... </applet>**.
- ▶ Unutar oznake **<applet>**, koristi se atribut "**code**" kako bi se zadalo **IME DATOTEKE** Java apleta
- ▶ Primer Sun (sada Oracle-ovog) apleta za analogni časovnik:

```
<applet code = "JavaClock.class" width = "150" height = "150" ></applet>
```

Deo Web aplikacije kreiran HTML-om.



Grafički deo Web aplikacije kreiran **APLETOM**.

*Java Web Start (1)

- **JAVA WEB START** (ili JavaWS, JAWS) je **NOVI RADNI OKVIR** za pokretanje Java aplikacija **DIREKTNO** u **WEB PRETRAŽIVAČU** (bez potrebe za dodatkom engl. *plugin*).
- Ključne prednosti ove tehnologije su **JEDNOSTAVNOST AŽURIRANJA** globalno distribuiranih aplikacija kao i veća **KONTROLA DODELJIVANJA MEMORIJE** u JVM-u.
- Za razliku od Java appleta, **JAVA WEB START** aplikacije se **NE PRIKAZUJU** unutar Web čitača.
- **JAVA WEB START** aplikacije se podrazumevano izvršavaju **U ISTOM BEZBEDNOM OKRUŽENJU** (*sandbox*-u) kao i apleti uz minorna proširenja koja se odnose na učitavanje i pamćenje datoteka koje su **EKSPPLICITNO SELEKTOVANE OD STARNE KORISNIKA**.
- Samo **BEZBEDNO POTPISANE APLIKACIJE** se mogu konfigurisati da imaju dodatne dozvole.

*Java Web Start (2)

- ▶ **JAVA WEB START** aplikacije imaju **PREDNOST NAD APLETIMA** jer rešavaju mnoge probleme vezane za KOMPATIBILNOST SA **JAVINIM DODACIMA** i različitim JVM verzijama.
- ▶ Za razliku od apleta, Web Start programi **NISU INTEGRALNI DEO WEB STRANICE** - oni su **NEZAVISNI PROGRAMI** koji se izvršavaju u zasebnom okviru.
- ▶ **WEB START** tehnologijom se čak **mogu pokrenuti** nemodifikovani apleti zapakovani u **.jar** datoteke, pisanjem odgovarajuće **.JNLP** (engl. **Java Network Launching Protocol**) datoteke.
- ▶ **JNLP** datoteka predstavlja **XML** zapis koji **OPISUJE ZAHTEVE ZA APLIKACIJU**, definiše se **LOKACIJA** koda, **PARAMETRE** i dodatne **DOZVOLE** (ako ih ima).
- ▶ **WEB ČITAČ** preuzima **.JNLP** datoteku, otvara je pomoću **Web Start** alata i **PREUZIMA** sve potrebne **RESURSE** i **POKREĆE** aplikaciju.
- ▶ Takvi apleti se takođe izvršavaju u **ZASEBNOM OKVIRU**.
- ▶ Kao i apleti, **JAVA WEB START** tehnologija je **CROSS-PLATFORMSKA**.

*Primer .jnlp (XML) datoteke

```
<?xml version="1.0" encoding="UTF-8"?>
<jnlp spec="1.0+" codebase="http://ultrastudio.org/upload" href="">
  <information>
    <title>Launch applet with Web Start</title>
    <vendor>Foo Bar Inc.</vendor>
    <offline-allowed/>
  </information>
  <resources>
    <j2se version="1.5+" href="http://java.sun.com/products/afsd1/j2se"/>
    <jar href="Ray-2.3-4ca60e46-0956-3f22-983c-e3ed986df013.jar" main="true" />
  </resources>
  <applet-desc name="Ray diagram applet" main-class="raydiagramapplet.Main"
    width="300" height="200">
  </applet-desc>
  <update check="background"/>
</jnlp>
```

INFORMACIJE

RESURSI

LOKACIJA

Multimedije - audio/video u HTML 5

- ▶ Za pojednostavljenje postavljanja **MULTIMEDIJALNOG SADRŽAJA** u HTML 5 su dodate **DVE** nove oznake: `<audio>` `</audio>` i `<video>` `</video>`.
- ▶ Ovo ima za posledicu da u HTML 5 **NISU POTREBNI DODATNI PROGRAMI** (kao u prethodnom primeru), čime on postaje **NEZAVISAN** od tehnologija drugih proizviđača.

Audio	Video
<pre><audio src="myaudio.ogg" controls> </audio></pre>	<pre><video src="myvideo.ogv" controls> </video></pre>
	

src atribut

Kontrole pridodate oznaci audio

src atribut

Kontrole pridodate oznaci video