

JavaScript a SVG

Cvičení 2

Z8144 Počítačová grafika v kartografii

Jaro 2020

Filip Leitner

	Téma	Zadanie	Odovzdanie	Body	Poznámka
9.3	Základy JavaScriptu, SVG	1			
16.3	JavaScript a SVG	2	1	10	do cvika
23.3	Interaktivita SVG + JavaScript	3	2	10	do cvika
30.3	Konzultácie				
6.4	Tvorba a export SVG	4	3	10	do 2.4.2021
13.4	Konzultácie				
20.4	SVG definice, gradienty, filtry				
27.4	HTML5 Canvas	5+6	4	40	do 23.4.2021
4.5	Konzultácie		5	nebodované	do cvika
11.5	Konzultácie				
18.5	Konzultácie		6	60	do cvika
25.5	Záver		Opravy		

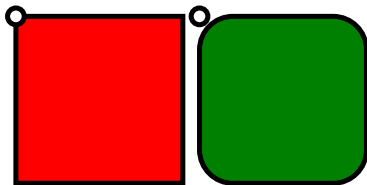
- Minimum 60% bodov
- Odovzdanie vždy do 4.00 k danému dátumu
 - ▶ Cvičenie 1 => do 16.3 4:00, Cvičenie 4 => 23.4 4:00

ZÁKLADNÍ TVARY – OPAKOVÁNÍ

rectangle	rect	obdélník/čtverec
circle	circle	kruh
ellipse	ellipse	elipsa
line	line	úsečka
polygon	polygon	
polyline	polyline	lomená čára
path	path	trasa
text	text	

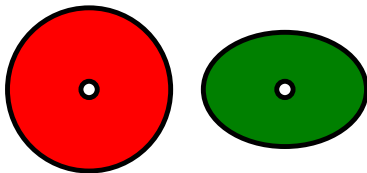
OBDELNÍK / ČTVEREC

```
<svg width="250" height="120">  
<rect x="10" y="10" width="100" height="100" stroke="black"  
  stroke-width="3" fill="red"/>  
<rect x="120" y="10" rx="20" ry="20" width="100" height="100"  
  stroke="black" stroke-width="3" fill="green"/>  
</svg>
```

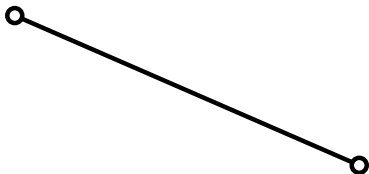


KRUH / ELIPSA

```
<svg width="250" height="120">  
  <circle cx="60" cy="60" r="50" stroke="black" stroke-width="3"  
    fill="red"/>  
  <ellipse cx="180" cy="60" rx="50" ry="35" stroke="black"  
    stroke-width="3" fill="green"/>  
</svg>
```

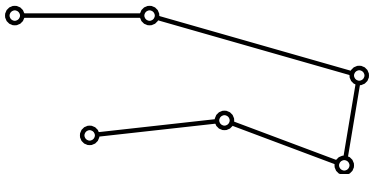


```
<svg width="250" height="120">  
  <line x1="10" y1="10" x2="240" y2="110" stroke="black"  
    stroke-width="3"/>  
</svg>
```



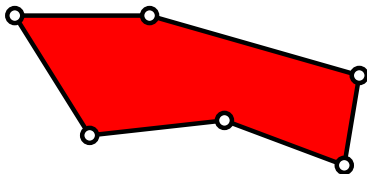
LOMENÁ ČÁRA

```
<svg width="250" height="120">  
  <polyline points="10,10 100,10 240,50 230,110 150,80 60,90"  
    stroke="black" stroke-width="3" fill="none"/>  
</svg>
```



POLYGON (N-ÚHELNÍK)

```
<svg width="250" height="120">  
  <polygon points="10,10 100,10 240,50 230,110 150,80 60,90"  
    stroke="black" stroke-width="3" fill="red"/>  
</svg>
```




```
<svg width="250" height="120" >  
  <text x="10" y="30" fill="black" font-family="Verdana"  
    font-size="24">Počítačová grafika</text>  
  <text x="10" y="60" fill="black" font-family="Verdana"  
    font-size="20">v kartografii</text>  
</svg>
```

Počítačová grafika
v kartografii

TRASA PATH

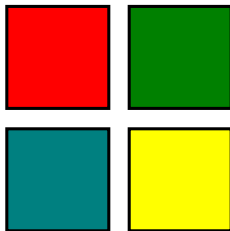
- **Velká** písmena – absolutní pozice
- **malá** písmena – relativní pozice

moveto	M(x y) (posun do bodu x y)
closepath	Z (uzavření trasy)
lineto	L(x y) (vykreslí čáru do bodu x y)
horizontal lineto	H(x) (horizontální čára až do)
vertical lineto	V(y) (svislá čára)
curveto	C(x1 y1 x2 y2 x y)
smooth curveto	S(x2 y2 x y)
quadratic Bézier curve	Q(x1 y1 x y)
smooth quadratic Bézier curveto	T(x y)
elliptical Arc	A(...) (eliptická výšeč/úseč)

<http://www.w3.org/TR/SVG/paths.html>

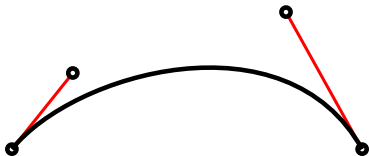
TRASA – PŘÍKLAD 1

```
<svg width="250" height="250" style="background-color: white;">  
  <path stroke="black" stroke-width="3" fill="red"  
    d="M10,10 L110,10 L110,110 L10,110 L10,10"/>  
  <path stroke="black" stroke-width="3" fill="green"  
    d="M130,10 l100,0 10,100 l-100,0 10,-100"/>  
  <path stroke="black" stroke-width="3" fill="teal"  
    d="M10,130 H110 V230 H10 z"/>  
  <path stroke="black" stroke-width="3" fill="yellow"  
    d="M130,130 h100 v100 h-100 z"/>  
</svg>
```



TRASA – PŘÍKLAD 2

```
<svg width="250" height="120" style="background-color: white;">  
  <path stroke="black" stroke-width="3" fill="none"  
    d="M10,100 C50,50 190,10 240,100"/>  
</svg>
```



1. ÚKOL

V SVG:

- vytvořte sněhuláka pomocí 3 kruhů, nos (mrkev) jako trojúhelník a oči
- vytvořte hvězdu pomocí trasy
- můžete použít CodePen (<http://codepen.io/pen/>)

1. ÚKOL - RIEŠENIE

```
<svg width="250" height="600" style="background-color:
  rgb(83, 83, 83);">
  <g class="snowman">
    <circle cx="100"cy="50"r="25" stroke-width="0"
      fill="white"/>
    <circle cx="86"cy="40"r="3" stroke-width="0"
      fill="black"/>
    <circle cx="104"cy="50"r="5" stroke-width="0"
      fill="black"/>
    <polygon points="95 45 63 48 95 55" fill="red"/>
    <circle
      cx="100"cy="120"r="50"stroke="black"stroke-width="0" f
    <circle
      cx="100"cy="220"r="80"stroke="black"stroke-width="0" f
  </g>
</svg>
```

1. ÚKOL - RIEŠENIE

```
<svg width="250" height="600" style="background-color:
  rgb(83, 83, 83);">
  <g class="star">
    <path fill="yellow"
      d="M25,400 120,0 110,-20 110,20 120,0 1-17,10
        110,20 1-20,-10 1-20,10 15,-20"/>
  </g>
</svg>
```

JAVASCRIPT – ZÁKLADNÍ EVENTY (UDÁLOSTI)

HTML atribut	název eventu	akce
<code>onclick</code>	<code>click</code>	kliknutí myši
<code>onmousedown</code>	<code>mousedown</code>	stisknutí tlačítka myši
<code>onmouseup</code>	<code>mouseup</code>	uvolnění tlačítka myši
<code>onmouseover</code>	<code>mouseover</code>	najetí myši na prvek
<code>onmouseout</code>	<code>mouseout</code>	odjetí myši z prvku
<code>onkeydown</code>	<code>keydown</code>	stisk klávesy
<code>onkeyup</code>	<code>keyup</code>	konec stisku klávesy

Rozdíly mezi eventy *click*, *mousedown*, *mouseup* viz <https://stackoverflow.com/a/14805233>.

ONCLICK

```
<div class="test" style="background-color:red;"  
  onclick="this.style.backgroundColor='green'">  
  <h3>Testovací div</h3>  
</div>
```

Příklad: <https://codepen.io/LeSimon/pen/ywgovB>

ONMOUSEOVER / ONMOUSEOUT

```
function testmouseover(element){
  element.style.backgroundColor='green';
};
function testmouseout(element){
  element.style.backgroundColor='red';
};
```

```
<div class="test" style="background-color:red;"
  onmouseover="testmouseover(this)"
  onmouseout="testmouseout(this)">
  <h3>Testovací div</h3>
</div>
```

Příklad: <https://codepen.io/LeSimon/pen/oVBeqe>

ONKEYDOWN / ONKEYUP

```
function testKeyUp(event){
  event.target.style.backgroundColor='red';
}
function testKeyDown(event){
  event.target.style.backgroundColor='green';
}
document.getElementById("input").addEventListener("keyup",
  testKeyUp);
document.getElementById("input").addEventListener("keydown",
  testKeyDown);
```

```
<input id="input" class="test"
  style="background-color:white;"></input>
```

Příklad: <https://codepen.io/LeSimon/pen/ywgoEL>

2. ÚKOL

Pomocí HTML a JS:

Vytvořte odstavec^(div) s textem, který po najetí myši změní tloušťku písma na tučné^(fontWeight:bold) a po odjetí myši se změní zpátky na normální^(fontWeight:normal).

Můžete použít CodePen (<http://codepen.io/pen/>)

2. ÚKOL - RIEŠENIE

```
...  
<div>  
  <p id="i">Some random text</p>  
</div>  
...
```

```
document.getElementById('i').addEventListener('mouseover', (e) =>{  
  e.target.style.fontWeight = 'bold';  
})  
function mouseOut (e){  
  e.target.style.fontWeight = 'normal';  
}  
document.getElementById('i').addEventListener('mouseout', mouseOut )
```

JAVASCRIPT – PRÁCE S SVG

V zásadě stejné jako doposud s HTML.

EVENTY V SVG DOKUMENTU

```
<svg width="250" height="120" style="background-color: white;">
  <circle cx="60" cy="60" r="50" stroke="black"
    stroke-width="3" fill="red"
    onmouseover="this.style.fill='green';"
    onmouseout="this.style.fill='red';"/>
</svg>
```

EVENTY POMOCÍ JS

```
<svg width="250" height="120" style="background-color: white;">
  <circle id="thecircle" cx="60" cy="60" r="50" stroke="black"
    stroke-width="3" fill="red" />
</svg>
```

```
const theCircle = document.getElementById("thecircle");

function circleOver(event){
  event.target.setAttribute('fill', 'green');
}
function circleOut(event){
  event.target.setAttribute('fill', 'red');
}
theCircle.addEventListener("mouseover", circleOver);
theCircle.addEventListener("mouseout", circleOut);
```


DOMÁCÍ ÚKOL

- Vytvořte **svg objekt** obsahující dva červené kruhy
- Po kliknutí se kruh vždy přemístí na jiné místo (náhodné) – použijte event `click`
- Úlohu je možné řešit více způsoby, dobře promyšlené řešení vám **ušetří spoustu úsilí**
- Úkol odevzdávejte jako **jeden HTML** soubor nebo **archiv .zip**
- Termín odevzdání: 23. 3. 4:00

Díky za pozornost!

<https://discord.gg/TKt6s3n>