

JavaScriptu a SVG

Cvičení 1

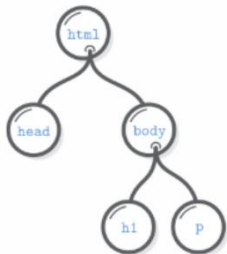
Aneta Ryglová

Z8144 Počítačová grafika v kartografii

Jaro 2024

Nástroje

- Textový editor: Visual Studio Code, Sublime Text, Atom, ...
- Internetový prohlížeč, vývojářská konzole (F12)
- Výuka JavaScriptu, reference:
<https://javascript.info/>
- Reference JavaScript:
<https://www.w3schools.com/jsref/>
- Online HTML/CSS/JS editor:
<https://codepen.io/pen/>



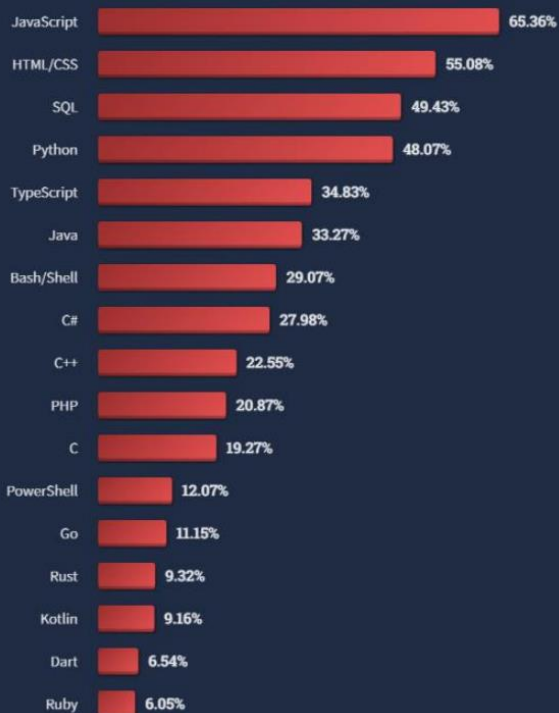
HTML



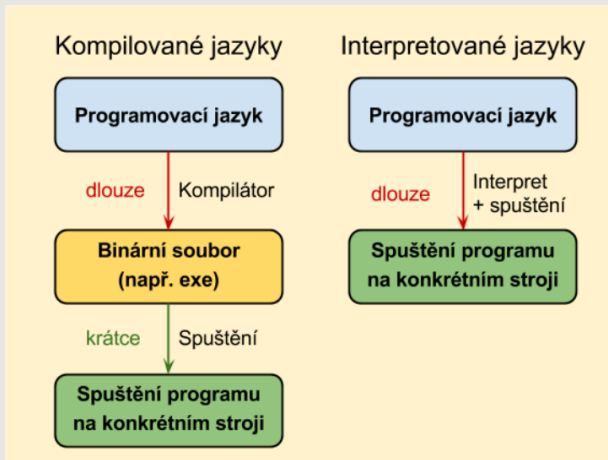
CSS



JAVASCRIPT



Javascript - principy

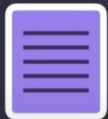


Interpretery v JS

- JS engine (Chrome V8, SpiderMonkey, JavaScriptCore) just-in-time compilation
- Developer tools(F12)
- Firefox Scratchpad (odebráno od verze 72, nahrazeno multi-line konzolí)
- Chrome snippets(<https://developers.google.com/web/tools/chrome-devtools/snippets>)
- Online editor <http://codepen.io/pen>

ECMAScript

Specification



JavaScript

Programming Language



Použití

```
1 console.log('Yeeeeee, I am in cosole!');  
2 alert('Nooooo an annoying popup alert window: /');
```


JavaScript v HTML5

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Page Title</title>
5
6          <script>...</script>
7      </head>
8      <body>
9      </body>
10 </html>
```

Co umí JS?

Práce s HTML/XML DOM:

- změna obsahu
- změna atributů
- změna stylů (HTML i CSS)

Změna obsahu

```
1 document.getElementById("test").innerHTML = "Hello World";  
2 const testConst = document.getElementById("test");  
3 testConst.innerHTML = "Hello World";
```

Změna atributů

Výpis atributů elementů: [console.dir\(\)](#)

```
1 document.getElementById("image").width = 100;
2 const myImage = document.getElementById("image");
3 myImage.width = 100;
```

Změna stylů

```
1 document.getElementById("paragraph").style.fontWeight = "bold";  
2 const myPar = document.getElementById("paragraph");  
3 myPar.style.fontWeight = "bold";
```

Operátory - přiřazení

=	přiřazení
+=	přičtení, ale také připojení řetězce
*= , -=, /=	prinásobení, odečtení, "přidělení"
++	přičtení 1
--	odečtení 1

Operátory - početní

+	sčítání, spojování řetězců
-	odčítání, unární negace
*	násobení
/	dělení

Operátory - logické

==	rovnost
!=	nerovnost
<,<=,>=,>	aritmetické srovnání
&&	logické AND (a zároveň)
	logické OR (nebo)
!	logické NOT (negace)
? :	podmínkový výběr (ternární operátor)
,	logické spojení (třeba v zápisu parametrů funkcí)

Rozdíl mezi == a ===

```
1 console.log("1" === 1) //false
2 console.log("1" == 1) //true
```

Rozdíl mezi var a let

```
1 {  
2     let x = 2;  
3  
4 } // x can NOT be used here  
5  
6 {  
7     var x = 2;  
8  
9 } // x CAN be used here
```

```
1  let speedKmph = distanceM / durationS / 3.6
2  let isEven = 90 % 2 === 0;
3  console.log(activity.type !== "bike");
4  let msg = "Your speed is " + speedKmph;
```

Operátory

Addition assignment (+=)

```
1 let a = 2;  
2 let b = 'hello';  
3 console.log(a += 3); // Addition  
4 // expected output: 5
```

Nullish coalescing operator (??)

```
1 let foo = null ?? 'default string';  
2 console.log(foo)  
3 // expected output: 'default string'
```

[a další...](#)

Jak si to mám zapamatovat?

Nijak ...stačí mozek!
...a internet!

<http://www.w3schools.com/jsref/>
<https://javascript.info/>
<https://google.com/>

1. úkol

Pracovní HTML soubor

Pomocí konzole JavaScriptu:

- změňte text prvního odstavce na "Hello world"
- změňte id druhého odstavce na "druhy"
- změňte velikost fontu (font-size: 7px) u třetího odstavce na 7px

2. úkol

Pracovní HTML soubor

Pomocí konzole JS:

1. přidejte 7 nových položek do seznamu (li)
2. změňte barvu všech hypertextových odkazů v odstavci s id "lipsum" na červenou (color: red;) a podtrhněte je (text-decoration: underline;)
3. zvětšete velikost písma (font-size: 2em;) všech odstavců (p), pokud obsahují třídu "big" (if) o 50 %

SVG – základní pojmy

- Scalable Vector Graphics
- vektor / rastr
- XML (eXtensible Markup Language)

tree	strom
element	prvek
attribute	atribut
parent	rodič
child	potomek
sibling	sourozenec

Tvorba grafiky

Grafický editor – Inkscape, Adobe Illustrator

Textový editor

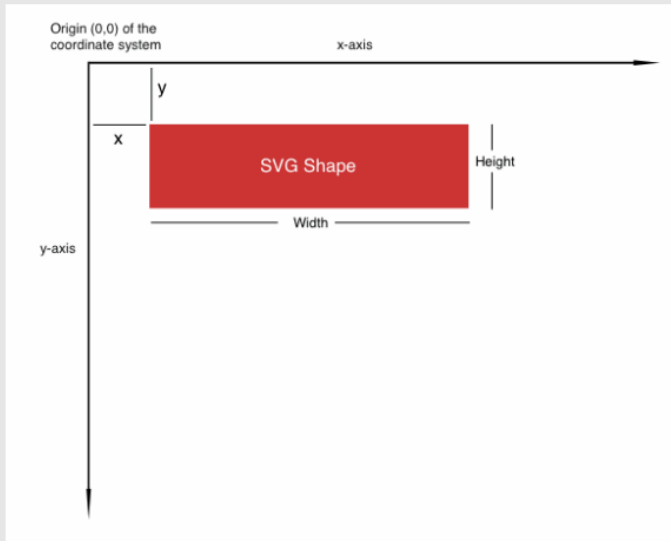
SVG v HTML

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Page Title</title>
5      </head>
6      <body>
7
8          <svg>...</svg>
9
10     </body>
11 </html>
```

Příklad SVG

```
1 <svg width="200" height="200">
2 <circle cx="100" cy="100" r="90"
3     fill="blue" stroke="black" stroke-width="8"/>
4 </svg>
```

SVG souřadnice



Základní tvary

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

Použití CSS

Presentation attributes

```
1 <svg width="200" height="200">
2 <rect x="25" y="25" width="150" height="150"
3     style="fill:blue; stroke:black; stroke-width:15;" />
4 </svg>
```

3. úkol

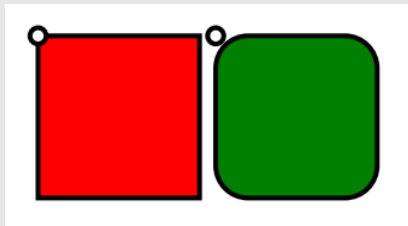
- Vytvořte HTML soubor (lokálně), se všemi náležitostmi (hlava, tělo, ...).
- Tento soubor bude obsahovat odstavec textu, který bude popisovat přiložený svg obrázek.
- Dále bude tento soubor obsahovat svg obrázek se žlutě ohraničenou modrou elipsou (atributy pro elipsu jsou: cx, cy, rx, ry).

1. bodovaný úkol

- Stáhněte si [šablonu pro 1. bodovaný úkol](#)
- Úkol vypracovávejte sami.
- Smíte měnit pouze část souboru mezi tagy `<script>a </script>`, který se nachází na úplném konci souboru. Zbytek souboru žádným způsobem neměňte.
- Odevzdaný soubor bude tedy vypadat shodně jako vzorový až na část s javascriptem.
- Vypracování úkolu zabere cca 20 řádků.
- V případě jakýchkoli dotazů či nejasností napište na Discord nebo e-mail.
- Každý z následujících bodů vypracujte pomocí **JavaScriptu**:
 - Pozměňte barvu všech hypertextových odkazů, které mají atribut třídy nastaven na red (`class='red'`), na červenou.
 - Změňte velikost písma všech nadpisů (h3) na 24px.
 - Pomocí javascriptu přidejte do druhého odstavce přes **for** cyklus 5 hypertextových odkazů, které odkazují na **is.muni.cz**.

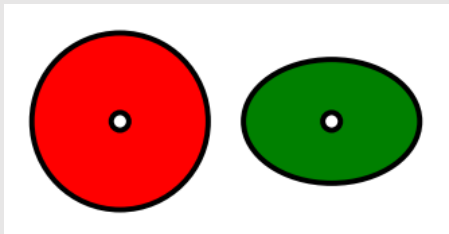
Obdélník / čtverec

```
<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>|
```



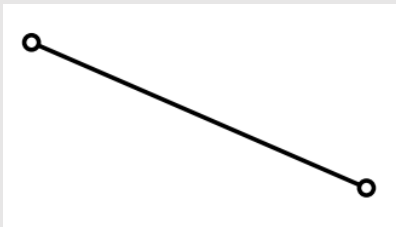
Kruh / elipsa

```
<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>
```



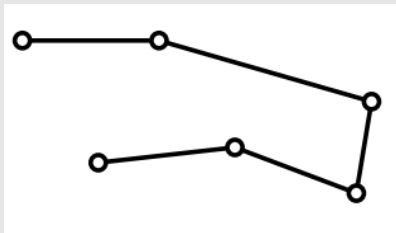
Úsečka

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



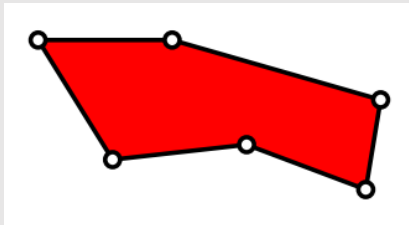
Lomená čára

```
<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>
```



Text

```
<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>
```

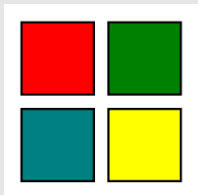
Počítačová grafika
v kartografii

Trasa Path

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)

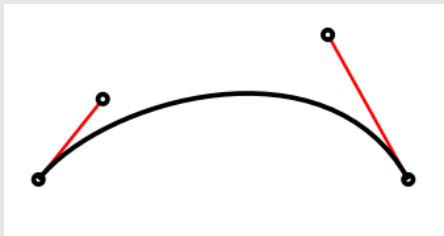
Trasa – příklad 1

```
<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 l0,100 l-100,0 l0,-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 - řešení

```
<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"fill="white"/>
    <circle
cx="100"cy="220"r="80"stroke="black"stroke-width="0"fill="white"/>
  </g>
</svg>
```

1. úkol - řešení

```
<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 myší
<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 myší na prvek
<code>onmouseout</code>	<code>mouseout</code>	odjetí myší z prvku
<code>onkeydown</code>	<code>keydown</code>	stisk klávesy
<code>onkeyup</code>	<code>keyup</code>	konec stisku klávesy

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

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

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

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 text (`fontWeight:normal`).

2. úkol - řešení

```
<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);
```

2. bodovaný 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
- Maximum je 10 bodů

Díky za pozornost!