

# Interaktivita SVG + JavaScript

Cvičení 3

Aneta Ryglová

**Z8144 Počítačová grafika v kartografii**

Jaro 2022

	<b>Téma</b>	<b>Zadání</b>	<b>Odevzdání</b>	<b>Body</b>
<b>14.2.</b>	-			
<b>20.2.</b>	základy js a svg	1		
<b>27.2.</b>	js a svg	2	1	10
<b>6.3.</b>	interaktivita svg a js	3	2	10
<b>13.3.</b>	konzultace			
<b>20.3.</b>	tvorba a export svg	4	3	10
<b>27.3.</b>	konzultace			
<b>3.4.</b>	svg definice			
<b>10.4.</b>	html5 canvas	5+6	4	40
<b>17.4.</b>	konzultace		5	0
<b>24.4.</b>	konzultace			
<b>1.5.</b>	svátek			
<b>8.5.</b>	svátek		6	60
<b>15.5.</b>	závěr			

Minimum 60% bodů

# Kahoot!

# Základní tvary - opakování

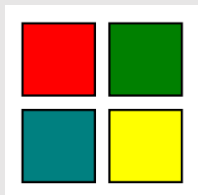
rectangle	rect	obdélník/čtverec
circle	circle	kruh
ellipse	ellipse	elipsa
line	line	úsečka
polygon	polygon	
polyline	polyline	lomená čára
path	path	trasa
text	text	

# 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)

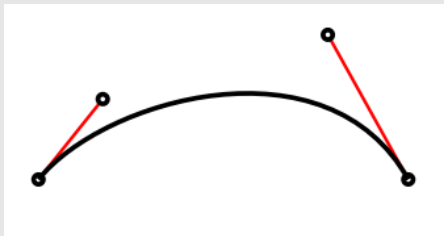
# Trasa – příklad 1

```
<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>
```



# 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>

# Array methods and iteration

## forEach()

```
1  let sum = 0;
2  const numbers = [65, 44, 12, 4];
3  numbers.forEach(myFunction);
4
5  function myFunction(item) {
6      sum += item;
7  }
8
9  console.log(sum) //125
```

# Array methods and iteration forEach()

```
1  let sum = 0;
2  const numbers = [65, 44, 12, 4];
3  numbers.forEach((item) => {
4      sum += item
5  });
6
7  console.log(sum) //125
```

# Array methods and iteration

## forEach()

<i>currentValue</i>	Required. The value of the current element.
<i>index</i>	Optional. The index of the current element.
<i>arr</i>	Optional. The array of the current element.

```
1  const numbers = [65, 44, 12, 4];
2  numbers.forEach((item, index, arr) => {
3      arr[index] = item * 10
4  });
5
6  console.log(numbers) //[ 650, 440, 120, 40 ]
```

# Array methods and iteration

## map()

<i>currentValue</i>	Required. The value of the current element.
<i>index</i>	Optional. The index of the current element.
<i>arr</i>	Optional. The array of the current element.

```
1  const numbers = [65, 44, 12, 4];
2  const newArr = numbers.map(myFunction);
3
4  function myFunction(num) {
5      return num * 10;
6  }
7
8  console.log(newArr) //[ 650, 440, 120, 40 ]
```

# Array methods and iteration

## map()

```
1  const numbers = [65, 44, 12, 4];  
2  
3  const newArr = numbers.map((num) => {  
4    return num * 10;  
5  });  
6  
7  console.log(newArr) //[ 650, 440, 120, 40 ]
```



# Array methods and iteration

## filter()

<i>currentValue</i>	Required. The value of the current element.
<i>index</i>	Optional. The index of the current element.
<i>arr</i>	Optional. The array of the current element.

```
1  const ages = [32, 33, 16, 40];  
2  
3  function checkAdult(age) {  
4    return age >= 18;  
5  }  
6  
7  console.log(ages.filter(checkAdult)); //[ 32, 33, 40 ]
```

# Array methods and iteration

## push()

<i>item1</i>	The item(s) to add to the array.
<i>item2</i>	Minimum one item is required.
..	
<i>itemX</i>	

```
1  const fruits = ["Banana", "Mango"];
2  fruits.push("Kiwi", "Lemon");
3
4  console.log(fruits) //[ 'Banana', 'Orange', 'Kiwi', 'Lemon' ]
```

# Array methods and iteration

## pop()

```
1  const fruits = ["Banana", "Orange", "Apple", "Mango"];
2  fruits.pop();
3
4  console.log(fruits) //[ 'Banana', 'Orange', 'Apple' ]
```

## toString()

```
1  const fruits = ["Banana", "Orange", "Apple", "Mango"];
2  let text = fruits.toString();
3
4  console.log(text)//Banana,Orange,Apple,Mango
```

# Array methods and iteration

## includes()

```
1 const fruits = ["Banana", "Orange", "Apple", "Mango"];
2 console.log(fruits.includes("Mango")); //true
```

## some()

```
1 const ages = [3, 10, 19, 2];
2
3 console.log(ages.some(age => {
4     return age > 18;
5 }))); //true
```

# Array methods and iteration

## every()

```
1  const ages = [3, 10, 19, 2];  
2  
3  console.log(ages.every(age => {  
4    return age > 18;  
5  })); //false
```

## findIndex()

```
1  const ages = [3, 10, 18, 20];  
2  
3  console.log(ages.findIndex(age => {  
4    return age === 18;  
5  })); //2
```

# Array methods and iteration

## join()

```
1  const fruits = ["Banana", "Orange", "Apple", "Mango"];
2  let text = fruits.join(" and ");
3
4  console.log(text); //Banana and Orange and Apple and Mango
```

## splice()

```
1  const fruits = ["Banana", "Orange"];
2  fruits.splice(1, 0, "Lemon", "Kiwi");
3
4  console.log(fruits);//[ 'Banana', 'Lemon', 'Kiwi', 'Orange' ]
```

# Array methods and iteration

## length()

```
1  const fruits = ["Banana", "Orange", "Apple", "Mango"];
2  let length = fruits.length;
3
4  console.log(length); //4
```

## concat()

```
1  const arr1 = ["Cecilie", "Lone"];
2  const arr2 = ["Emil", "Tobias"];
3  const arr3 = arr1.concat(arr2);
4
5  console.log(arr3);//[ 'Cecilie', 'Lone', 'Emil', 'Tobias' ]
```

# 1. úkol

[https://github.com/anetryg/pocitacova\\_grafika\\_2023/blob/main/cv3/arrayMethods.js](https://github.com/anetryg/pocitacova_grafika_2023/blob/main/cv3/arrayMethods.js)



## 2. úkol

V SVG:

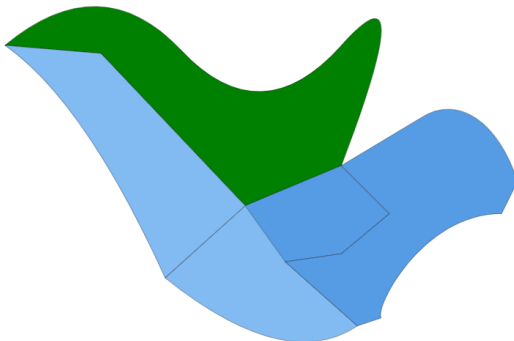
- Vytvořte mapové pole, které bude obsahovat smyšlený stát a jeho kraje (alespoň 5 krajů).
- Toto mapové pole (obrysy jednotlivých krajů) vytvořte pomocí tras.
- Použijte i křivky nebo kruhové výseče.
- Jednotlivým „krajům“ přidejte atribut „data-population“, který bude obsahovat fiktivní data o populaci.
- Výsledek si uložte někam na disk (budeme s ním dále pracovat).

### 3. úkol

- Přidejte interaktivitu této „mapě“. Pomocí JavaScriptu (mimo SVG) nastavte „mouseover“ a „mouseout“ event.
- Po najetí myši na libovolný „kraj“ by se tento kraj měl zabarvit definovanou barvou (vyberte si vlastní). Zároveň by se někde na stránce měl zobrazit údaj o populaci daného kraje.
- Pokud jste správně nastavili atribut data-population, dostanete se k němu v JS pomocí `element.dataset.population`.
- Po odjetí myši by se měl obnovit původní vzhled kraje a informace o populaci zmizí.
- Úkol odevzdávejte jako jeden HTML soubor nebo archiv .zip
- Termín odevzdání 20.3 4:00, max. 10 b.

# Stát a jeho SVG kraje

Počet obyvatel daného kraje je 80.



Díky za pozornost!