


# Výběr prvku na stránce

```
let nadpis;
```

```
nadpis = document.querySelector("h1");
```



Nalezený prvek si uložíme do proměnné **nadpis**

Vyhledáváme v celém dokumentu

Hledáme jeden prvek

Popis prvku, který hledáme. Zapisuje se stejně, jako kdybychom psali selektor pro prvek v CSS

# Selektory

- Aneb to, co píšeme do uvozovek dovnitř  
**document.querySelector("...");**

**h1** HTML prvek

**.foto** Prvek s CSS třídou class="foto"

**#cedule** Prvek s id="cedule"

**.menu a** Odkaz uvnitř prvku s třídou menu

# Změna vlastností prvku

- Když jsem získal objekt prvku na stránce pomocí **document.querySelector("...");** mohu v JavaScriptu měnit jeho vlastnosti
- Přímou nastavovat CSS styly
- Přidávat / odebírat CSS třídy
- Měnit atributy prvky
- Měnit textový obsah prvku

## Přímá změna CSS stylu prvku

```
let nadpis;
```

```
nadpis = document.querySelector("h1");
```

```
nadpis.style.color = "red";
```

## Přímá změna CSS stylu prvku

- CSS vlastnosti se v JavaScriptu zapisují pomocí tzv. camel case. Vlastnosti se zapisují dohromady bez mezer a pomlček, první písmeno je malé, další slova v názvu začínají velkým písmenem.

background-color	→	backgroundColor
font-size	→	fontSize

# Změna textového obsahu prvku

```
let nadpis;
```

```
nadpis = document.querySelector("h1");
```

```
nadpis.textContent = "Ahoj";
```

Měníme pouze textový obsah prvku, nemůžeme vložit jiné značky nebo formátování

## Změna HTML obsahu prvku

```
let odstavec;
```

```
odstavec = document.querySelector("p");
```

```
odstavec.innerHTML =
```

```
    "Umíme <strong>JavaScript</strong>";
```

Pomocí innerHTML můžeme do obsahu prvku přidávat i jiné HTML značky (zde např. tučný text)

# Přidání/odebrání CSS třídy pomocí classList

Jako vždy si nejprve najdeme v dokumentu příslušný prvek:

```
let nadpis;  
nadpis = document.querySelector("h1");
```

```
nadpis.classList.add("csstrida");
```

↑  
Našemu vybraný  
nadpis

↑  
Seznam CSS tříd  
aplikovaných na  
objektu

↑  
Do seznamu  
přidáme

↑  
CSS třídu s tímto názvem

Výsledek je stejný, jako bychom v HTML napsali `<h1 class="csstrida">Můj nadpis</h1>`



# Přidání/odebrání CSS třídy pomocí `classList`

Stejným způsobem můžeme CSS třídy z objektu odebírat:

```
nadpis.classList.remove("csstrida");
```

Můžeme také třídu na objektu přepínat:

- pokud třída na objektu neexistuje, tak se přidá
- pokud třída na objektu existuje, tak se naopak odebere

```
nadpis.classList.toggle("csstrida");
```

# Události

- Událost nastává na prvku na stránce ve chvíli, kdy se s prvkem něco děje
- Může se jednat např. o kliknutí na prvek, najetí kurzoru myši, stisk klávesy, změna hodnoty formulářového pole, apod.
- Událostem nasloucháme a ve chvíli, kdy k nim dojde, na ně můžeme reagovat

# Nejpoužívanější události

<b>click</b>	kliknutí na prvek
<b>mouseover</b>	kurzor myši najel nad prvek
<b>mouseout</b>	kurzor myši odjel z prvku pryč
<b>keydown</b>	stisk klávesy
<b>change</b>	změna hodnoty pole ve formuláři
<b>load</b>	kompletní načtení prvku (typicky celé stránky)

A spousta dalších, viz např. <https://developer.mozilla.org/en-US/docs/Web/Events>

# Nasloucháme událostem

Jako vždy si nejprve najdeme v dokumentu příslušný prvek:

```
const tlacitko = document.querySelector("a");
```

```
tlacitko.addEventListener("click", priKliknuti);
```

↑  
Na našem  
tlačítku

↑  
Nasloucháme  
události

↑  
Kliknutí myši

↑  
Když k události dojde,  
zavoláme tuto funkci

# Reagujeme na kliknutí

```
const tlacitko = document.querySelector("a")
```

```
const priKliknuti = () => {  
    // zde je náš kód, který se provede  
    // při kliknutí na tlačítko  
}
```

```
tlacitko.addEventListener("click", priKliknuti)
```