

Základy JavaScriptu

Všechny příkazy/řádky v JavaScriptu **ukončuj středníkem**. Je to sice až na výjimky nepovinné, ale důrazně se to doporučuje. Zabráníš tím spoustě nechtěných chyb.



Proměnná

Proměnná je pojmenované místo pro uložení libovolné hodnoty. Proměnnou nadefinuješ jednou na začátku programu nebo funkce pomocí klíčového slova **var** a dále už do ní jen vkládáš nebo z ní čteš hodnoty.

```
var jmeno;           // nadefinujeme proměnnou
jmeno = "Lucie";     // vložíme do ní hodnotu

// lze zapsat i dohromady na jednom řádku
var pocet = 42;
```

Podmínka

Podmínka slouží pro rozhodování v programu. Pokud je podmínka splněna, vykoná se kód ve větvi **if**. Není-li podmínka splněna, vykoná se kód ve větvi **else**. Větev **else** je nepovinná a lze ji vynechat.

```
if (pocet === 3) {
  // tento řádek se vykoná, je-li počet roven 3
} else {
  // je-li jiný počet, proved' toto
}
```

Rovnost hodnot porovnáváš pomocí trojitého rovnítká **===**. Použít můžeš samozřejmě i další **srovnávací operátory** **<**, **>**, **<=**, **>=** nebo **!==** (nerovná se).

Výrazy uvnitř podmínek lze spojovat pomocí **logických operátorů** **||** (nebo), **&&** (a zároveň), **!** (ne, neplatí).

```
if (jmeno === "Lucie" && vek >= 18) {
  // kód pouze pro plnoleté Lucie
}
```

Cyklus

Slouží pro opakované provedení sady příkazů. Cyklus **for** se obvykle používá, když dopředu znáš počet opakování. Cykly **while** a **do/while** se typicky používají hlavně, když počet opakování předem neznáš.

```
for (var i = 0; i < 5; i++) {
  // příkazy uvnitř cyklu se provedou celkem 5x
  // i++ na konci každé smyčky cyklu zvětší i o 1
  // proměnná i postupně nabývá hodnot 0, 1, 2, 3 a 4
}
```

```
while (podmínka) {
  // dokud platí podmínka, prováděj příkazy uvnitř cyklu
  // neplatí-li podmínka, příkazy se nemusí provést ani jednou
}
```

```
do {
  // podmínka se testuje až na konci
  // příkazy uvnitř cyklu se vždy provedou alespoň jednou
} while (podmínka);
```

Funkce

Funkce je sada příkazů, které chceš provádět opakovaně. Pomocí klíčového slova **function** je zabalíš do funkce, kterou si nějak pojmenuješ, a pak ji tímto jménem voláš.

Do funkce můžeš (ale nemusíš) předat hodnoty, tzv. **parametry funkce**. Pomocí klíčového slova **return** můžeš vrátit zpět z funkce **výstupní hodnotu** (výsledek).

```
// nadefinujeme funkci
function nazevFunkce() {
  // příkazy uvnitř funkce
}

// funkci zavoláme jménem
nazevFunkce();
```

```
// funkce s parametry
function soucet(a, b) {
  // pomocí return můžeme
  // z funkce vrátit hodnotu
  return a + b;
}
```

```
// zavoláme funkci s parametry
var x = soucet(37, 5);
// x === 42
```

Jak pomocí JavaScriptu ovlivňovat HTML prvky na stránce

Pro práci s HTML prvkem na stránce musíš tento prvek nejprve najít a získat na něj odkaz, který si většinou uložíš do proměnné, aby byla další práce s ním snazší.

Vyhledání HTML prvku na stránce

K vyhledání prvku v dokumentu slouží metoda `querySelector`.

`document.querySelector` najde první prvek na stránce, který odpovídá daným kritériím. Pokud kritériím odpovídá na stránce více prvků, vrátí metoda jen první z nich.

`document.querySelectorAll` nalezne všechny prvky na stránce, které odpovídají daným kritériím, a vrátí jejich seznam. Tento seznam lze potom postupně procházet např. pomocí cyklu `for`.

Jako vyhledávací kritéria lze použít jakýkoliv CSS selektor. Lze tedy hledat podle id, názvu třídy, typu elementu, včetně kombinovaných selektorů i pseudoselektorů.

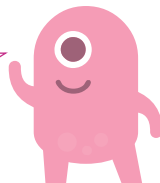
```
// najdeme značku (tag) pro hlavní nadpis H1
var element = document.querySelector("h1");

// najdeme div s přiřazenou css třídou "menu"
var element = document.querySelector("div.menu");

// najdeme div s id="nadpis"
var element = document.querySelector("#nadpis");

// najdeme první odstavec v patičce stránky
var element = document.querySelector("footer p:first-child");
```

JavaScript je nejlepší :) Všichni mimozemšťani prostě milují JavaScript. Můžeme se po něm utlouct. Mňam!



Přímé nastavení CSS vlastností

Nejčastějším úkonem prováděným nad HTML prvky je změna jejich CSS vlastností (zobrazení/skrytí, barva, poloha, ...). K CSS vlastnostem přistupuješ pomocí **`prvek.style.cssVlastnost`**.

Všechny víceslovné názvy CSS vlastností, které mají v názvu pomlčku, v JavaScriptu zapíšeš pomocí tzv. camel case - pomlčky se odstraní, první písmeno je malé, každé další slovo začíná velkým písmenem.

v CSS	v JavaScriptu	
color	color	barva textu
background-color	backgroundColor	barva pozadí
font-size	fontSize	velikost písma

```
// najdeme na stránce prvek s id="nadpis"
// a nastavíme jeho barvu na červenou a zvětšíme písmo
var nadpis = document.querySelector("#nadpis");
nadpis.style.color = "red";
nadpis.style.fontSize = "80px";
```

Práce s CSS třídami

Často není vhodné nastavovat CSS vlastnosti prvku přímo, ale raději bychom k prvku přidali nebo z něj odebrali CSS třídu. Přidáním CSS třídy (ta je nadefinovaná v CSS souboru) tak můžeš na prvku změnit spoustu vlastností najednou.

S třídami pracuješ pomocí **`prvek.classList`** a používáš metody **`add`**, **`remove`** nebo **`toggle`** (přidá třídu, pokud na prvku ještě není; odebere třídu, pokud už na prvku je).

```
var nadpis = document.querySelector("#nadpis");
nadpis.classList.add("velkyCerveny");
nadpis.classList.remove("velkyCerveny");
nadpis.classList.toggle("velkyCerveny");
```

Události

V HTML stránce dochází k různým událostem - stisk klávesy, pohyb nebo kliknutí myši, dokončení načtení stránky, apod. V JavaScriptu můžeš na tyto události reagovat. Kompletní seznam událostí je mnohem delší, toto jsou ty nejpoužívanější, se kterými si zatím vystačíš.

Událost	Popis
onLoad	k události dojde, když se do prohlížeč dokončil načítání celá stránka; hodí se, potřebujeme-li provést něco hned při "startu" stránky
onClick	při kliknutí na prvek myši
onMouseOver	při najetí myši nad prvek
onMouseOut	při odjetí myši pryč z prvku
onKeyDown	při stisknutí klávesy
onKeyUp	při uvolnění klávesy

Poloha prvků na stránce

HTML prvky na stránce se standardně řadí na stránce automaticky. Chceš-li přesně nastavovat polohu prvku na stránce, musíš mu nejprve nastavit CSS vlastnost **position: absolute;**

Potom můžeš polohu prvku určovat pomocí vlastností **left** a **top** (levý horní roh) nebo **right** a **bottom** (pravý dolní roh). Hodnoty lze nastavovat v různých jednotkách, nejčastěji se ale používají pixely (obrazové body).

```
var mimoszemstan = document.querySelector("#mimoszemstan");
mimoszemstan.style.position = "absolute";
mimoszemstan.style.left = "70px";
mimoszemstan.style.top = "50px";
```

index.html

```
<html>
<body onLoad="priStartu();"
      onKeyDown="priStiskuKlavesy(event);">
  <h1 onClick="priKliknuti();">Nadpis</h1>
</body>
</html>
```

program.js

```
function priStartu() {
  // tato funkce se zavolá při načtení stránky
}

function priKliknuti() {
  // tato funkce se zavolá při kliknutí na nadpis
}

function priStiskuKlavesy(event) {
  // tato funkce se zavolá při stisku klávesy
  // z parametru event můžeš zjistit, o jakou klávesu šlo
  if (event.key === "A" ) {
    // stisknuto velké A
  }
}
```

