

```
7 =^.^=
8 =^.^=
9 =^.^=
```

### 1.3 SYNTAXE

Náš software zahrnuje různé symboly, včetně závorek (), středníků ;, složených závorek {}, plusových znamének + a několika slov, která mohou na první pohled působit záhadně (například var a console.log). Všechny tyto prvky tvoří syntaxi programovacího jazyka JavaScript, což jsou pravidla určující, jak kombinovat symboly a slova k vytváření funkčních programů.

### 1.4 KOMENTÁŘE

První řádek našeho programu obsahuje:

```
// Nakresli kočičí obličejíky!
```

Toto je komentář a program jej nečte. Komentáře slouží nám k přehlednosti v kódu. Jednořádkový komentář vytvoříme pomocí dvou lomených závorek, jak je v ukázce, víceřádkové komentáře vytvoříme pomocí lomítka s hvězdičkou:

```
/*
Nakresli kolik
Koček chceš!
*/
```

### 1.5 CVIČENÍ

#### 1.5.1 Pozdrav mě

Vytvořte program, který napíše x-krát za sebou „Dobrý den!“. Hodnotu x nastav na 5.

## 2 DATOVÉ TYPY A PROMĚNNÉ

### 2.1 DATOVÉ TYPY

**Number** (číslo)

```
5;
```

**String** (řetězec)

```
"Dobrý den.";
```

**Boolean** (logický typ)

```
true;
```

```
//Násobení
99 * 123;
//Výsledek
12177
//Žádáme o první 4 znaky řetězce
"This is a long string".slice(0, 4);
//Výsledek
"This"
//Kontrola pravdivosti dvou operací
true && false;
//Výsledek
false
```

### 2.2 PROMĚNNÉ

Abychom dali hodnotám názvy, používáme proměnné. Pro vytvoření nové proměnné používáme klíčové slovo `var`.

```
//Nadefinovaná proměnná, která nemá danou hodnotu
var nick;
//Nadefinovaná proměnná věk, kterou jsme nastavili na číslo 12
var vek = 12;
//Když chceme proměnnou volat zpět a získat její hodnotu, pouze napíšeme její
název
console.log (age);
//Pokud již nadefinované proměnné chceme hodnotu změnit
vek = 13;
//Možnost výpočtu
var pocet_sourozencu = 1 + 3;
var pocet_tycinek = 8;
console.log (numberOfCandies / numberOfSiblings);
```

#### 2.2.1 Pojmenování proměnných

Názvy jsou v JavaScriptu citlivé na velikosti písmen. Nadefinujeme-li si proměnnou:

```
var pocetSourozencu = 5;
```

Nemůžeme volat proměnnou `PocetSourozencu` a očekávat výstup číslo 5, proměnná `PocetSourozenců` totiž nadefinovaná není.

#### 2.2.2 Cvičení

##### 2.2.2.1 Sekundy v hodině

Napiš program, který si uloží do proměnných počet sekund v minutě a počet minut v hodině a v proměnné počet sekund v hodině vypočítá hodnotu. Následně nechej výsledek vypsát do konzole.

##### 2.2.2.2 Sekundy za den

Přidej další proměnné a vypočítej počet sekund za den, výsledek nechej vypsát do konzole.

### 2.2.2.3 Sekundy za rok

Přidej další proměnné a vypočítej počet sekund za rok, výsledek nechej vypsát do konzole.

### 2.2.2.4 Věk v sekundách

Vytvoř program, který vypočítá tvůj věk v sekundách. Výsledek nechej vypsát do konzole.

## 2.3 INCREMENTING (ZVĚTŠENÍ) A DECREMENTING (ZMENŠENÍ)

„Incrementing“ a „decrementing“ jsou operace v programování, které se používají k zvětšení nebo zmenšení hodnoty proměnné o jednotku.

### 2.3.1 Incrementing (zvětšení o jedno)

Přidání jedné jednotky k aktuální hodnotě proměnné. V JavaScriptu se to může provést pomocí operátoru ++.

```
let x = 5;
x++;
// x nyní obsahuje hodnotu 6
```

### 2.3.2 Decrementing (zmenšení o jedno)

Odečítání jedné jednotky od aktuální hodnoty proměnné. V JavaScriptu se to může provést pomocí operátoru --.

```
let y = 8;
y--;
// y nyní obsahuje hodnotu 7
```

Tyto operace jsou užitečné při opakování nebo v různých scénářích, kde je potřeba dynamicky měnit hodnoty proměnných o jednotku.

### 2.3.3 Cvičení

#### 2.3.3.1 Highfive

Vytvořte proměnnou, která má na začátku počet placáků nula, následně jí pomocí navýšení zvyšte o dva a poté opět snižte o jeden.

## 2.4 += (PLUS-ROVNÁ SE) A -= (MÍNUS-ROVNÁ SE)

Abychom zvýšili hodnotu proměnné, musíme provést například následující příkaz.

```
var x = 10;
x = x + 5;
//Následně je x rovno 15
```

Díky použití += a -= ovšem můžeme tento zápis zjednodušit.

```
var y = 10;
y += 7;
//V tuto chvíli jsme y zvětšili na hodnotu 17
```

### 2.4.1 Cvičení

#### 2.4.1.1 Vyzkoušej si opsat následující zápisy a zjist, co dělají:

První zápis:

```
var balonky = 100;
balonky *= 2;
console.log (balonky);
```

Druhý zápis:

```
var cukratka = 100;
```

```
cukratka /= 4;
console.log (cukrátka);
```

## 2.5 STRINGS

String jsou textové řetězce. To ovšem neznamená, že se skládají pouze z textu, textový řetězec může být i číslo 15, když jej zapišeme ale jako datový typ string, nejsme schopni s tímto zápisem počítat, protože program jej chápe jako text, nikoli jako hodnotu čísla. String zapisujeme do úvozovek.

```
var string_pozdrav = "Dobrý den!";
console.log (string_pozdrav);
```

Pojďme se tedy podívat na rozdíl v zápisu čísla jako číslo a jako string.

```
var cislo_jako_cislo = 2;
var cislo_jako_string = "2";
console.log (cislo_jako_cislo + cislo_jako_cislo);
console.log (cislo_jako_string + cislo_jako_string);
```

V případě, že je 2 zapsáno jako číslo, program s ní umí počítat. V případě, že je zapsáno jako string, považuje jej program za znak a v případě sčítání jej napíše jen za sebou.

### 2.5.1 Spojování řetězců

Pro spojování řetězců můžeme použít operátor plus +.

```
var pozdrav = "Ahoj";
var jmeno = "Franto";
console.log (pozdrav + jmeno);
```

Tento zápis ovšem napíše AhojFranto, tedy bez mezery, jelikož žádnou mezeru jsme programu nezadali. Můžeme tedy pro console.log použít zápis pozdrav + „ „ + jmeno.

### 2.5.2 Zjišťování délky řetězce

Abychom zjistili délku řetězce, přidáme za příkaz .length

```
console.log ("ZjistimiJakJeTotoDlouhé".length);
```

### 2.5.3 Získání znaku z řetězce

Pro získání znaku z řetězce použijeme následující zápis.

```
var prezdivka = "Nick";
console.log (prezdivka[0]);
```

Tento zápis nám vypíše první (nulté) písmeno, tedy N.

#### 2.5.3.1 Cvičení

Jaké je tajné heslo? První si sám vyzkoušej, poté pro ověření správnosti své odpovědi přepiš kód a spusť.

```
var heslo1 = "informatika";
var heslo2 = "tlačítko";
var heslo3 = "pátek";
console.log (heslo1[0] + heslo2[6] + heslo3[2]);
```

### 2.5.4 Dělení řetězců

Abychom oddělili od sebe části řetězců, použijeme příkaz .slice a za něj do závorky vložíme dvě čísla, první určuje kterým znakem začneme a druhý, kterým skončíme.

```
var dlouhy_retezec = "Toto je můj dlouhý řetězec. ";
console.log (dlouhy_retezec.slice (12, 26));
```

Můžeme ovšem zadat i pouze začáteční znak a následně mi příkaz vypíše vše od daného znaku.

```
console.log (dlouhy_retezec.slice (12));
```

### 2.5.5 Změna řetězce na všechna velká nebo malá písmena

```
var na_velka = "Ahoj, jak se máš?".toUpperCase();
var na_mala = "AHOJ, jak SE MÁŠ?".toLowerCase();
```

## 2.6 BOOLEANS

Boolean může obsahovat pouze dvě hodnoty a to true nebo false.

### 2.6.1 Logické operátory

<b>a</b>	<b>b</b>	<b>and &amp;&amp;</b>	<b>or !!</b>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

#### 2.6.1.1 Not

$a = 0 \rightarrow y = 0$     $a = 1 \rightarrow y = 0$