**Javascript**

Proměnné v Javascriptu se definují pomocí klíčových slov let a const.

Typy dat v javascriptu zahrnuji:

1. String
2. Number
3. Boolean
4. Array
5. Object

Operátory v Javascriptu dělíme na:

1. Aritmetické (+,-,/,\*,%)
2. Prorovnávající

* == rovnost – nekontroluje datový typ
* === přísná rovnost a kontroluje datový typ
* != nerovnost
* !== přísná nerovnost
* > větší než
* < menší než
* >= větší nebo rovno
* <= menší nebo rovno

1. Logické

* && a zároven
* || nebo
* ! ne

**Podmíněné výrazy (if/else)**

Podmínky jsou základní způsob, jak řídit tok programu. Na základě určitých podmínek můžeme vykonat specifický kód.

**Cyklus (loops)**

cyklus ti umožní opakovat určité úkony.

**for** cyklus:

Nejčastěji používaná cyklus, která se používá, když víme, kolikrát chceme něco opakovat.

for (let i = 0; i < 5; i++) {

console.log(i); // Vypíše čísla 0, 1, 2, 3, 4

}

**while** cyklus:

cyklus **while** běží, dokud je podmínka pravdivá.

let i = 0;

while (i < 5) {

console.log(i); // Vypíše čísla 0, 1, 2, 3, 4

i++;

}

**Úkol:**

1. Zkus si napsat podmíněný výraz, který vypíše, jestli je číslo sudé nebo liché pomocí **if/else** a operátoru **%** (modulo).
2. Napiš smyčku, která vypíše všechna čísla od 1 do 10.
3. let a = [];
4. for (let i=1;i<11;i++){
5. //console.log(i);
6. a.push(i);
7. }
8. console.log(a);
9. for (let cislo of a){
10. if (cislo % 2 ==0 ){
11. console.log(cislo);
12. console.log("Cislo je sudé...");
13. }
14. else{
15. console.log(cislo);
16. console.log("Čislo je liché");
17. }
18. }

**Funkce**

Umožnuji seskupit část kodu, kterou můžu znovu a znovu používat.

     function pozdrav(jmeno){

        console.log("Čau",jmeno);

      }

      pozdrav("kamil");

      // funkce s návratovou hodnotou

      function secti(a,b){

        return a+b;

      }

      // z funkce si výsledek vrátím a uložím do proměnné

      let vysledek = secti(5,3);

      console.log(vysledek);

funkce může vracet hodnotu pomocí klíčového slova return.

Existují také anonymní funkce, takže funkci mohu uložit do proměnné.

 const pozdrav = function(){

            console.log("Ahoj světe");

     }

     pozdrav();

Arrow funkce

Arrow funkce jsou novější způsob, jak psát funkce v Javascriptu. Byly zavedeny ES6 a jsou především kratší a jednodušší variantou tradičních funkci. Jejich největší výhodou je však to, jak zachází s klíčovým slovem this, což může být v klasických funkcí matoucí.

Arrow funkce

Const pozdrav = (jmeno) **=>**{

console.log(“Ahoj“+jmeno);

}

Tradiční funkce

**function** pozdrav(jmeno){

console.log(“Ahoj“+jmeno);

}

Šipka => nahrazuje klíčové slovo function

Ještě kratší verze funkce může vypadat následovně.

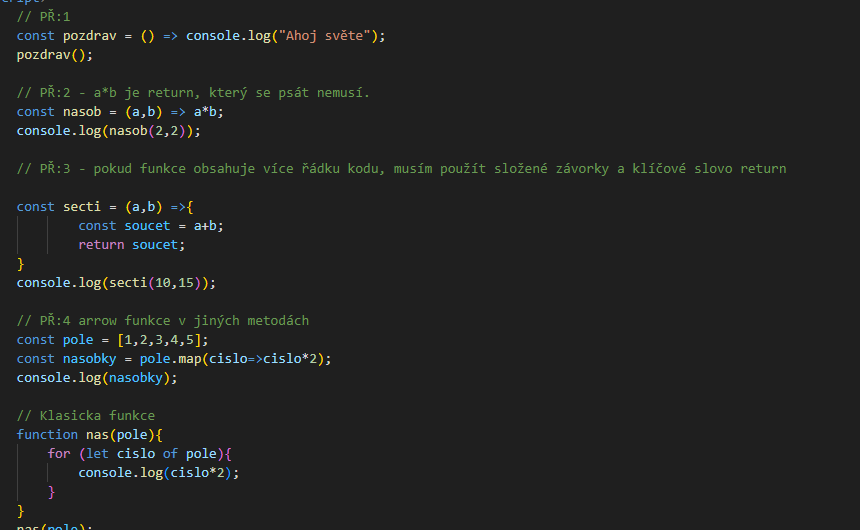
const pozdrav = jmeno => “Ahoj” + jmeno; -> pokud v těle funkce je pouze jedna instrukce.

Arrow funkce nemají svoje this, takže odkazuje na hodnotu z okolního nadřazeného kontextu.

### Úkoly

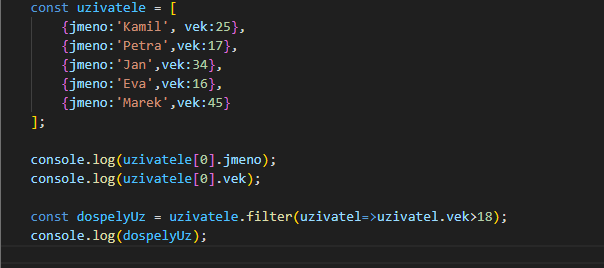
1. **Úkol 1**: Napiš **arrow funkci** s názvem **nasob**, která přijme dvě čísla jako parametry a vrátí jejich součin.
2. **Úkol 2**: Máš pole čísel [1, 2, 3, 4, 5]. Použij **arrow funkci** v kombinaci s metodou **map**, abys vytvořil nové pole, kde každé číslo bude zvýšeno o 3.





**Pole objektu**

V Javascriptu je pole objektů takové, kde každý prvek pole je object, který má své vlasnosti. V tomhle případě má každý object vlasnosti jmeno a vek



**Úkoly:**

1. **Cykly a podmínky**:
   * Napiš for cyklus, který vypíše čísla od 1 do 20. Pokud je číslo sudé, vypíše "sudé", a pokud je liché, vypíše "liché". Použij if/else.
2. **Funkce**:
   * Vytvoř funkci s názvem pozdrav, která přijme parametr jmeno a vrátí zprávu ve formátu "Ahoj, jméno!" (kde jméno bude nahrazeno hodnotou parametru).
3. **Arrow funkce**:
   * Napiš arrow funkci s názvem nasob, která přijme dvě čísla jako parametry a vrátí jejich součin.
4. **Pole a mapování**:
   * Máš pole čísel [2, 4, 6, 8, 10]. Použij arrow funkci v kombinaci s metodou map, abys vytvořil nové pole, kde každé číslo bude vyděleno dvěma.

