

Učebnice JavaScriptu

k videím na YouTube

David Šetek

Odkaz na učebnici

<https://bit.ly/JavaScriptUcebnice>



Seznam videí na kanále YouTube

David Šetek - počítače a IT

https://www.youtube.com/playlist?list=PLQ8x_VWW6Aku2x097IYBmRaQD0PPe8HBH

1. Co je to JavaScript

<https://youtu.be/A3GjzmUoeSs>

PowerPointová prezentace

https://drive.google.com/file/d/1JiTI369G7hD5dTQh911zw-iVkmNZ_Osj/view?usp=sharing

2. Stáhni si učebnici zdarma a pojd' makat

<https://youtu.be/3mXcSOvt494>

Odkaz na stažení učebnice: <http://bit.ly/JavaScriptUcebnice>

3. Brackets neboli kam psát JavaScript

https://youtu.be/KOnuOo_89Rc

JavaScript (stejně jako HTML a CSS) budeme v kurzu psát do IDE Brackets. Brackets si můžete stáhnout zde: <http://brackets.io/>

Doporučuji nainstalovat doplňky s názvem:

- HTML Skeleton
- Emmet

4. Založení nového projektu v Brackets

https://youtu.be/ILAUXP38c_4

Jak jsme postupovali ve videu

Založte si novou složku na ploše (např. s názvem JavascriptStranky, ale název je na vás).

Otevřete ji v Brackets.

V Brackets založte dva soubory: index.html a style.css

Do index.html přidejte pomocí HTML Skeletona strukturu stránky

Propojte index.html a style.css

Vyzkoušejte propojení tak, že obarvíte nadpis h1 na červenou

V index.html budete mít:

```
<!DOCTYPE html>
```

```
<html lang="">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title></title>
```

```
  <link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
  <h1>Toto je hlavní nadpis!!!!!!</h1>
```

```
</body>
```

```
</html>
```

V style.css budete mít:

```
h1 {  
  color: red;  
}
```

5. Náš první JavaScriptový kód a console.log

<https://youtu.be/xTVzjXX-HqA>

Do index.html pod body dáme:

```
<script src="script.js"></script>
```

Do souboru s názvem script.js napíšeme:

```
console.log("David");  
console.log("Martin");  
console.log("Jana");
```

A výpis jmen zkontrolujeme v developerské konzoli v záložce Console.

6. Proměnné v JavaScriptu

<https://youtu.be/1-Qp3xA36XM>

Do script.js napíšeme:

```
var firstName = "David";  
console.log(firstName);
```

```
var lastName = "Šetek";  
console.log(lastName);
```

```
var age = 20;  
console.log(age);
```

```
var job = "Teacher";  
console.log(job);
```

7. Datové typy proměnných - Střih

<https://youtu.be/5BWMuT910AY>

```
// number - číslo  
var age = 20;  
console.log(age);
```

```
// string - text  
var firstName = "David";
```

```
console.log(firstName);
```

```
// boolean - pravda / nepravda
```

```
var adult = true;
```

```
var adult = false;
```

```
console.log(adult);
```

```
// undefined - proměnná nemá hodnotu
```

```
var secondName;
```

```
console.log(secondName);
```

```
// null - proměnná neexistuje
```

```
// console.log(test);
```

```
var age2 = 25;
```

```
    age2 = "David2";
```

```
console.log(age2);
```

8. Komentáře v JavaScriptu

<https://youtu.be/c1-MDxuR9qY>

```
// toto je jednořádkový komentář
```

```
/*
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
*/
```

```
/******
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
    Víceřádkový komentář
```

```
*/
```

9. Výpis více proměnných najednou v JavaScriptu

https://youtu.be/IRcP_Sg5c2U

```
var firstName = "David";  
var secondName = "Šetek";  
var age = 33;
```

```
console.log("Tento člověk se jmenuje " + firstName + " " + secondName + " a je mu " + age +  
" let");
```

10. Alert, confirm a prompt v JavaScriptu

<https://youtu.be/6IAqWnk5iFs>

Postupně jsme v script.js zadávali:

```
alert("Ahoj, já jsem David");
```

```
confirm("Jsi plnoletý?");
```

```
var age = prompt("Kolik je ti let?");  
var adult = age - 18;  
console.log(adult);
```

11. Základní operátory v JavaScriptu

https://youtu.be/w_hkQgPTMDw

```
/******
```

```
* Základní operátory
```

```
*/
```

```
var ageDavid = 15;  
var ageJana = 20;
```

```
var year = 2021;  
var birthDavid = year - 15;  
var birthJana = year - 20;
```

```
console.log(birthJana / 1000);  
console.log(birthJana * 2);  
console.log(birthJana + 5000);  
console.log(birthJana - 500);
```

```
console.log(birthJana * 2 - 500);  
console.log((5+5)*(10+10));
```

12. Logické operátory v JavaScriptu a ještě jednou confirm

<https://youtu.be/XsA6HSW0Ehg>

```
/******  
* Logické operátory  
*/  
//< > == <== >==  
var result = (ageJana == ageDavid);  
console.log(result);  
  
/******  
* Confirm  
*/  
  
var resultConfirm = confirm("Jsi plnoletý/á?");  
console.log(resultConfirm);
```

13. Typeof v JavaScriptu a jak souvisí s datovými typy

<https://youtu.be/72DTqA9JZaU>

```
var ageDavid = 15;  
var ageJana = 20;  
var result = (ageJana == ageDavid);  
  
console.log(typeof(ageDavid));  
console.log(typeof(ageJana));  
console.log(typeof(result));  
console.log(typeof(20));  
console.log(typeof("20"));
```

14. Další operátory v JavaScriptu Střih

https://youtu.be/_bZLqhayVVE

Zvyšování o jedničku (dá se použít pro zvyšování o libovolné číslo):

```
var x = 1;  
x += 1;  
console.log(x);  
x += 1;  
console.log(x);  
x += 1;  
console.log(x);
```

Zápis `x += 1` je stejný, jako zápis `x = x + 1`

Snižování o jedničku (dá se použít pro snižování o libovolné číslo):

```
var x = 1;  
x -= 1;  
console.log(x);  
x -= 1;  
console.log(x);  
x -= 1;  
console.log(x);
```

Zápis `x -= 1` je stejný, jako zápis `x = x - 1`

Stejně se dá použít i pro násobení (*) i dělení (/)

Ještě další zápis při zvyšování o jedničku

`x++;` (stejně jako `x = x + 1` popř. `x += 1`)

Ještě další zápis pro snižování o jedničku

`x--;` (stejně jako `x = x - 1` popř. `x -= 1`)

15. Podmínky if a else v JavaScriptu

https://youtu.be/qhc0-4_hpuE

```
if (3 > 0){  
    // podmínka je true = pravda  
    console.log("Je to pravda.");  
} else {  
    // podmínka je false = nepravda  
    console.log("Není to pravda.");  
}
```

```
var ageDavid = 35;  
var ageJana = 20;
```

```
if (ageDavid > ageJana){  
    console.log("David je starší než Jana.");  
} else {  
    console.log("Jana je starší než David");  
}
```

16. Načtení hodnoty od uživatele a vyhodnocení pomocí podmínky v JavaScriptu

<https://youtu.be/O17Uh4FsJYw>

```
var age = prompt("Kolik je ti let?");

if (age >= 18) {
    console.log("Jsi plnoletý. Můžeš vstoupit.")
} else {
    console.log("Ještě ti nebylo 18. Nemůžeš vstoupit.")
}
```

17. If, else a else if v JavaScriptu

<https://youtu.be/9i6HEr3WNkE>

```
var ageDavid = 20;
var ageJana = 20;

if (ageDavid > ageJana){
    console.log("David je starší než Jana");
} else if (ageDavid < ageJana){
    console.log("David je mladší než Jana.")
} else {
    console.log("David i Jana jsou stejně staří.")
}
```

18. Převod datových typů a jednoduchá kalkulačka v JavaScriptu

<https://youtu.be/gTH0G1jX3i0>

```
var firstNumber = Number(prompt("Zadej první číslo:"));
var secondNumber = Number(prompt("Zadej druhé číslo:"));
var sign = prompt("Zadej plus nebo minus");

if (sign == "+") {
    console.log(firstNumber + secondNumber);
} else if (sign == "-") {
    console.log(firstNumber - secondNumber);
} else {
    console.log("Tutu matematickou operaci neznám.");
}
```

```
console.log(typeof(Number("20"))); // převod textu na číslo
console.log(typeof(String(20))); // převod čísla na text
```


19. And (&&) a or (||) v JavaScriptu

<https://youtu.be/IETfy4Objrl>

// Jmenuji se David a zároveň bydlím v ČB

```
// true = jmenuji se David a jsem z ČB
// false = jmenuji se David a jsem z Prahy
// false = jmenuji se Petr a jsem z ČB
// false = jmenuji se Petr a jsem z Prahy
&& - a zároveň
```

Jmenuji se David a zároveň jsem starší 18 let

```
if (ageDavid > 18 && ageDavid < 30){
    // když obě dvě jsou pravdivé
} else {
    // když je jedna z nich nepravdivá,
    // obě jsou nepravdivé
}
```

// Jmenuji se David nebo bydlím v ČB

```
// true = jmenuji se David a jsem z ČB
// true = jmenuji se David a jsem z Prahy
// true = jmenuji se Petr a jsem z ČB
// false = jmenuji se Petr a jsem z Prahy
|| - nebo
```

Jmenuji se David nebo jsem starší 18 let

```
if(nameDavid == "David" || ageDavid > 18){
    // když jsou obě pravda
    // když je alespoň jedna z nich pravda
} else {
    // když jsou obě nepravda
}
```

20. Doděláme naší kalkučka, aby nešlo dělit nulou v JavaScriptu

<https://youtu.be/n3DQpY6n2eo>

```
var firstNumber = Number(prompt("Zadej první číslo:"));
var secondNumber = Number(prompt("Zadej druhé číslo:"));
var sign = prompt("Zadej znak pro matematickou operaci");
```

```

if (sign == "+") {
    console.log(firstNumber + secondNumber);
} else if (sign == "-") {
    console.log(firstNumber - secondNumber);
} else if (sign == "*") {
    console.log(firstNumber * secondNumber);
} else if (sign == "/" && secondNumber != 0) {
    console.log(firstNumber / secondNumber);
} else if (sign == "/" && secondNumber == 0){
    console.log("Nesmíš dělit nulou. Druhé číslo nesmí být nula při dělení!!!!");
} else {
    console.log("Nedokáži takovouto operaci provést.");
}

```

21. Ternární operátor v JavaScriptu

<https://youtu.be/dZ3Gkq5NeOM>

```

var ageDavid = 10;

```

```

if (ageDavid >= 18){
    console.log("David je dospělý");
} else {
    console.log("David je mladiství");
}

```

```

/*****

```

```

* Ternární operátor

```

```

*/

```

```

ageDavid >= 18 ? console.log("David je dospělý") : console.log("David je mladiství");

```

```

-----

```

```

var ageDavid = 26;

```

```

/*****

```

```

* Ternární operátor

```

```

*/

```

```

//if (ageDavid >= 18){
//    var drink = "pivo";
//} else {
//    var drink = "voda";
//}

```

```

var drink = ageDavid >= 18 ? "pivo" : "voda";

```

```
console.log(drink);
```

22. Switch v JavaScriptu

<https://youtu.be/4WfwJF9l4Co>

```
/******  
* Swich  
*/  
var pet = "pig";  
  
switch(pet){  
  case "cat":  
    console.log("Máme kočku");  
    break;  
  case "dog":  
    console.log("Máme psa");  
    break;  
  case "fish":  
    console.log("Máme rybičku");  
    break;  
  default:  
    console.log("Máme jiné zvíře");  
}
```

```
-----  
  
/******  
* Switch  
*/  
  
var ageDavid = 5;  
  
//if (ageDavid >= 18){  
//  console.log("David je dospělý");  
//} else if (ageDavid <= 18 && ageDavid >= 8) {  
//  console.log("David je mladiství");  
//} else {  
//  console.log("David je dítě");  
//}  
  
switch(true){  
  case ageDavid >= 18:  
    console.log("David je dospělý");  
    break;  
  case ageDavid <= 18 && ageDavid >= 8:  
    console.log("David je mladiství");
```

```
        break;
    default:
        console.log("David je dítě");
}
```

23. Funkce v JavaScriptu a DRY princip

<https://youtu.be/Yjz1E1IUaeQ>

princip DRY - do not repeat yourself (neopakuj se)

```
/******
 *   Function
 */

/* Výpočítáme věk pomocí console.log */
function calculateAge1(birth){
    console.log(2020 - birth);
}

calculateAge1(2000);
calculateAge1(1990);
calculateAge1(1980);

/* Výpočítáme věk pomocí return */
function calculateAge2(birth){
    return 2020 - birth;
}

var ageDavid = calculateAge2(2000);
var ageJohn = calculateAge2(1990);
var ageDiana = calculateAge2(1980);

console.log(ageDavid,ageJohn,ageDiana);
```

24. Složitější funkce v JavaScriptu

<https://youtu.be/gTW0PvDx5Ks>

```
/******
 *   Function
 */

/* Výpočítáme věk */
function calculateAge(birth){
```

```

    return 2020 - birth;
}

function yearsUntilRetirement(year, firstName){
    var age = calculateAge(year);
    var retirement = 65 - age;

    if (retirement >= 0){
        console.log("Počet let do důchodu " + retirement + " let" + " u " + firstName);
    } else {
        retirement = -(retirement);
        console.log(firstName + " je už v důchodu " + retirement + " let.");
    }
}

yearsUntilRetirement(2000, "David");
yearsUntilRetirement(1990, "John");
yearsUntilRetirement(1955, "Diana");

```

25. Function statements a function expressions v JavaScriptu

<https://youtu.be/rYC09u2-zEQ>

```

/*****
*   Function
*/

// Function statements
function calculateAge(birth){
    return 2020 - birth;
}

console.log(calculateAge(2000));

// Function expressions
var calculateAge = function(birth){
    return 2020 - birth;
}

console.log(calculateAge(1920));

```

Další příklad

```

// Function expressions
var myJob = function(job, firstName){
    switch(job){
        case "učitel":

```

```

        return firstName + " je " + job;
    case "manager":
        return firstName + " je " + job;
    default:
        return "Nevím, co má " + firstName + " za práci.";
    }
}

```

```

console.log(myJob("učitel", "David"));
console.log(myJob("manager", "Jane"));
console.log(myJob("truhlář", "John"));

```

26. Falsy a truthy hodnota v JavaScriptu

<https://youtu.be/rYC09u2-zEQ>

```

// falsy = undefined, 0, "", Nan, null
// truthy = to, co není falsy

```

// undefine

```

var salary;

if (salary){
    console.log("Hodnota true");
} else {
    console.log("Hodnota false");
}

```

// ""

```

var salary = "";
if (salary){
    console.log("Hodnota true");
} else {
    console.log("Hodnota false");
}

```

// 0

```

var salary = 0;
if (salary || salary == 0){
    console.log("Hodnota true");
} else {
    console.log("Hodnota false");
}

```

27. Rozdíl mezi == a === v porovnání v JavaScriptu

<https://youtu.be/1Wz9uJ2yfdA>

== neřeší datový typ (nejsou tak přísné)

=== hlídají si datový typ (jsou přísnější)

```
var salary = "0";
```

```
if (salary === 0){  
    console.log(true);  
} else {  
    console.log(false);  
}
```

28. Pole v JavaScriptu

<https://youtu.be/sVy-he03s2o>

```
/******
```

```
*   Array (pole)
```

```
*/
```

```
var employees = ["David", "Jane", "Diana"];  
//var employees = new Array("David", "Jane", "Diana");
```

```
console.log(employees);  
console.log(employees[0]);  
console.log(employees[1]);  
console.log(employees[2]);
```

```
var age = [30, 20, 40];
```

```
console.log(age[0]);  
console.log(age[1]);  
console.log(age[2]);
```

```
var david = ["David", "Šetek", 33, 1980, true];
```

```
console.log(david[3]);
```

29. Práce s polem v JavaScriptu

https://youtu.be/lnjV_3ag7ho

```
var employees = ["David", "Jane", "Diana"];
```

```
employees[0] = "Martin";  
employees[1] = "Anna";  
employees[5] = "Eduard";
```

```
console.log(employees);
```

```
-----  
var employees = ["David", "Jane", "Diana"];
```

```
/*  
    push - přidá na konec,  
    unshift - přidá na začátek,  
  
    pop - odebere poslední,  
    shift - odebere první.  
*/
```

```
employees.push("John");  
employees.unshift("Albert");
```

```
employees.pop();  
employees.shift();
```

```
console.log(employees);
```

30. Pole a indexOf a length v JavaScriptu

<https://youtu.be/98yYsA-MUh4>

```
var employees = ["David", "Jane", "Diana", "Martin", "Eduard"];
```

```
console.log(employees.indexOf("David"));  
console.log(employees.indexOf("Eduard"));  
console.log(employees.indexOf("Natalia"));  
console.log(employees.indexOf("Petr"));
```

```
console.log(employees.length);  
console.log(employees[employees.length - 1]);
```


31. Objekty v JavaScriptu

<https://youtu.be/Y3zAXFey3YQ>

```
/******  
*   Objects (objekty)  
*/  
  
// pole = hodnotám jsou přiřazené index; záleží na pořadí  
var employees = ["David", "Jane", "Diana"];  
  
// objekt = hodnotám jsou přiřazené jiné hodnoty; nezáleží na pořadí  
  
var david = {  
    firstName: "David",  
    lastName: "Setek",  
    birthYear: 1980,  
    driveLicence: true,  
    family: ["John", "Anna", "Diana"]  
}  
  
console.log(david.firstName);  
console.log(david.lastName);  
console.log(david.driveLicence);  
  
console.log(david["firstName"]);  
console.log(david["lastName"]);  
console.log(david["driveLicence"]);  
  
// Druhá možnost, jak zapsat objekt  
var diana = new Object();  
diana.firstName = "Diana";  
diana.lastName = "Novotna";  
diana.birthYear = 1999;  
diana.driveLicence = false;  
  
console.log(diana);  
  
-----  
var david = {  
    firstName: "David",  
    lastName: "Setek",  
    birthYear: 1980,  
    driveLicence: true,  
    family: ["John", "Anna", "Diana"]  
}
```

```
david.firstName = "Martin";  
david.birthYear = 2000;  
david.driveLicence = false;
```

```
console.log(david);
```

32. Metody u objektů v JavaScriptu

<https://youtu.be/x-i7r2KzUP0>

```
/******
```

```
*   Objects (objekty)
```

```
*/
```

```
// Metody u objektu = funkce
```

```
var david = {  
    firstName: "David",  
    lastName: "Setek",  
    birthYear: 1980,  
    driveLicence: true,  
    family: ["John", "Anna", "Diana"],  
    calculateAge: function(birthYear){  
        return 2020 - birthYear;  
    }  
}
```

```
console.log(david.calculateAge(1990));
```

```
var david = {  
    firstName: "David",  
    lastName: "Setek",  
    birthYear: 1980,  
    driveLicence: true,  
    family: ["John", "Anna", "Diana"],  
    calculateAge: function(){  
        return 2020 - this.birthYear;  
    }  
}
```

```
var diana = {  
    firstName: "David",  
    lastName: "Setek",  
    birthYear: 2000,  
    driveLicence: true,  
    family: ["John", "Anna", "Diana"],
```

```
    calculateAge: function(){
        return 2020 - this.birthYear;
    }
}
```

```
var eduard = {
    firstName: "David",
    lastName: "Setek",
    birthYear: 1990,
    driveLicence: true,
    family: ["John", "Anna", "Diana"],
    calculateAge: function(){
        return 2020 - this.birthYear;
    }
}
```

```
var anna = {
    firstName: "David",
    lastName: "Setek",
    birthYear: 1950,
    driveLicence: true,
    family: ["John", "Anna", "Diana"],
    calculateAge: function(){
        return 2020 - this.birthYear;
    }
}
```

```
console.log(david.calculateAge());
console.log(diana.calculateAge());
console.log(eduard.calculateAge());
console.log(anna.calculateAge());
```

```
var david = {
    firstName: "David",
    lastName: "Setek",
    birthYear: 1980,
    driveLicence: true,
    family: ["John", "Anna", "Diana"],
    calculateAge: function(){
        this.age = 2020 - this.birthYear;
    }
}
```

```
david.calculateAge();
```

```
console.log(david);

var diana = {
  firstName: "David",
  lastName: "Setek",
  birthYear: 2000,
  driveLicence: true,
  family: ["John", "Anna", "Diana"],
  calculateAge: function(){
    this.age = 2020 - this.birthYear;
  }
}

diana.calculateAge();
console.log(diana);
```

33. Cykly: cyklus for

<https://youtu.be/eIYB1L1zetM>

```
/******
 *   Cykly
 */

// výpis bez cyklu
console.log(0);
console.log(1);
console.log(2);
console.log(3);
console.log(4);
console.log(5);
console.log(6);
console.log(7);
console.log(8);
console.log(9);

// výpis s cyklem
for(var i = 0; i < 10; i++){
  console.log(i);
}

for(var i = 1; i <= 5; i++){
  console.log(i + ": Já jsem první řádek cyklu");
  console.log(i + ": Já jsem druhý řádek cyklu");
}
```

34. Cykly: procházení pole cyklem for

<https://youtu.be/rFL5MQw5ERs>

```
var employees = ["David", "Jane", "Diana"];
```

```
console.log(employees[0]);  
console.log(employees[1]);  
console.log(employees[2]);
```

```
for(var i = 0; i < 3; i++){  
    console.log(employees[i]);  
}
```

```
-----  
var employees = ["David", "Jane", "Diana", "Peter", "Eduard", "Anna"];
```

```
for(var i = 0; i < employees.length; i++){  
    console.log(employees[i]);  
}
```

35. Cykly: příklady na cyklus for

<https://youtu.be/EGxD4NNChNs>

```
/******
```

```
*   Příklady na cyklus for
```

```
*/
```

```
// Vypište 12x své jméno a příjmení
```

```
for(var i = 0; i < 12; i++){  
    console.log("David Šetek");  
}
```

```
// Vypište čísla od 0 do 10
```

```
for(var i = 0; i < 10; i++){  
    console.log(i);  
}
```

```
// Vypište čísla od 5 do 50 včetně, ale jen násobky pěti
```

```
for(var i = 5; i <= 50; i = i + 5){  
    console.log(i);  
}
```

```
// Vypište čísla od 15 do 0 (tedy v obráceném pořadí)
```

```
for(var i = 15; i >= 0; i--){  
    console.log(i);  
}
```

36. Cykly: While cyklus

<https://youtu.be/uxsblNAK5pQ>

```
/******  
*   While cyklus  
*/
```

```
for(var i = 0; i < 10; i++){  
    console.log(i);  
}
```

```
var i = 0;  
while(i < 10){  
    console.log(i);  
    i++;  
}
```

```
/*
```

For cyklus - když známe přesný počet opakování,

While - neznáme přesný počet opakování

Kolikrát můžete dané číslo dělit 2 tak, aby výsledek byl větší nebo rovno 1.

12 : 2 = 6
6 : 2 = 3
3 : 2 = 1,5
1,5 : 2 = 0,75

40 : 2 = 20
20 : 2 = 10
10 : 2 = 5
5 : 2 = 2,5
2,5 : 2 = 1,25
1,25 : 2 = 0,625
*/

37. Cykly: Do while cyklus a porovnání všech cyklů

<https://youtu.be/9zGFmYv3PPY>

```
/******
```

* Do while cyklus

*/

// for - známe počet opakování

```
for(var i = 0; i < 0; i++){  
    console.log("David Setek");  
}
```

// while - neznáme počet opakování, nemusí se provést ani jednou

```
var i = 0;  
while(i < 0){  
    console.log("David Setek");  
    i++;  
}
```

// do while - neznáme počet opakování, provede se minimálně jednou

```
var i = 0;  
do {  
    console.log("David Setek");  
    i++;  
}while(i < 0);
```

38. Cykly: break a continue v cyklu

<https://youtu.be/qvIAPTwd26Q>

Break

/*

break - způsobí ukončení cyklus

continue - zastaví průchod cyklem, ale cyklus pokračuje dál (např. zastaví první průchod cyklem, ale pokračuje druhým opakováním)

*/

```
var wanted = ["Killer", "Bloody", "Hacker"];
```

```
var traveler = prompt();
```

```
for(var i = 0; i < 3; i++){
```

```
    console.log(i);
```

```
    if(wanted[i] === traveler){
```

```
        console.log("Zadržte ho");
        break;
    }
}
```

Continue

```
var wanted = ["Killer", "Bloody", "Hacker"];
```

```
var traveler = prompt();
```

```
for(var i = 0; i < 3; i++){
```

```
    console.log(i);
```

```
    if(wanted[i] === traveler){
        console.log("Zadržte ho");
        continue;
    }
```

```
    console.log("Toto se neprovede poté, co se narazí na continue")
}
```

39. Co je to DOM v JavaScriptu

<https://youtu.be/g58f9lHN3Bs>

Vybíráme tag h1

```
<!DOCTYPE html>
```

```
<html lang="">
```

```
<head>
```

```
    <title>Moje první stránky</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Ahoj, já jsem David</h1>
```

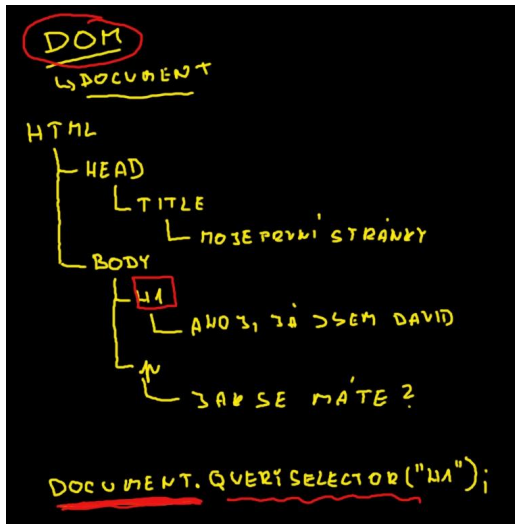
```
    <p>Jak se máte?</p>
```

```
</body>
```

```
<script src="script.js"></script>
```

```
</html>
```


Takto vypadá DOM výše uvedeného HTML kódu:



JavaScript:

```
var text = document.querySelector("h1").textContent;
console.log(text);
```

Vybíráme třídu

HTML:

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <p class="pozdrav">Jak se máte?</p>
</body>
```

JavaScript:

```
var text = document.querySelector(".pozdrav").textContent;
console.log(text);
```

Vybíráme idéčko

HTML:

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <p id="pozdrav">Jak se máte?</p>
</body>
```

JavaScript:

```
var text = document.querySelector("#pozdrav").textContent;
console.log(text);
```

40. Query Selector a změna textu v HTML pomocí JavaScriptu

<https://youtu.be/fQgQ-nySsMw>

Vybíráme tagy

HTML:

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <p>Jak se máte?</p>
</body>
```

JavaScript:

```
var headline = document.querySelector("h1").textContent;
console.log(headline);
```

```
var paragraph = document.querySelector("p").textContent;
console.log(paragraph);
```

Vybíráme třídu

HTML:

```
<body>
  <h1 class="heading1">Ahoj, já jsem David</h1>
  <p class="paragraph1">Jak se máte?</p>
</body>
```

JavaScript:

```
var headline = document.querySelector(".heading1").textContent;
console.log(headline);
```

```
var paragraph = document.querySelector(".paragraph1").textContent;
console.log(paragraph);
```

Vybíráme idéčko

HTML:

```
<body>
  <h1 id="heading1">Ahoj, já jsem David</h1>
  <p id="paragraph1">Jak se máte?</p>
</body>
```

JavaScript:

```
var headline = document.querySelector("#heading1").textContent;
console.log(headline);
```

```
var paragraph = document.querySelector("#paragraph1").textContent;
console.log(paragraph);
```

Vkládáme nový text

HTML:

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <p>Jak se máte?</p>
</body>
```

JavaScript:

```
document.querySelector("h1").textContent = "Já jsem nová H1";
```

```
document.querySelector("p").textContent = "Já jsem nový odstavec";
```

41. Query Selector All a výběr více stejných tagů v JavaScriptu

<https://youtu.be/XtmwcTvGGfQ>

HTML

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <p>První odstavec</p>
  <p>Druhý odstavec</p>
  <p>Třetí odstavec</p>
</body>
```

Výběr více stejných tagů z HTML a výpis pole bez cyklu

```
var text =
document.querySelectorAll("p");
```

```
console.log(text[0].textContent);
console.log(text[1].textContent);
console.log(text[2].textContent);
```

Výpis pole cyklem

```
var text =
document.querySelectorAll("p");

for(var i = 0; i < 3; i++){
  console.log(text[i].textContent);
}
```

Výpis jakkoli velkého pole

HTML

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <p>První odstavec</p>
```

```
<p>Druhý odstavec</p>
<p>Třetí odstavec</p>
<p>Čtvrtý odstavec</p>
<p>Pátý odstavec</p>
</body>
```

JavaScript:

```
var text =
document.querySelectorAll("p");

for(var i = 0; i < text.length; i++){
    console.log(text[i].textContent);
}
```

42. Výběr a změna CSS pomocí query selectoru v JavaScriptu

<https://youtu.be/0SiAGs1pTmc>

```
document.querySelector("h1").style.color = "blue";

document.querySelector("h1").style.display = "none";

document.querySelector("h1").style.fontSize = "100px";

document.querySelector("h1").style.fontWeight = 400;
```

43. GetElementsByClassName a getElementById v JavaScriptu

<https://youtu.be/OboqzqAulDg>

HTML

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <p class="prvni">První odstavec</p>
  <p>Druhý odstavec</p>
  <p class="prvni">Třetí odstavec</p>
  <p>Čtvrtý odstavec</p>
  <p class="prvni">Pátý odstavec</p>
</body>
```

JavaScript - výpis pole tříd bez cyklu

```
var mojeTridy =
document.getElementsByClassName("prvni");

console.log(mojeTridy[0].textContent);
```

```
console.log(mojeTridy[1].textContent);
console.log(mojeTridy[2].textContent);
```

JavaScript - výpis pole tříd pomocí cyklu

```
var mojeTridy =
document.getElementsByClassName("prvni");

for(var i = 0; i < 3; i++){
    console.log(mojeTridy[i].textContent);
}
```

JavaScript - výpis jakkoli velkého pole tříd pomocí cyklu

```
var mojeTridy =
document.getElementsByClassName("prvni");

for(var i = 0; i < mojeTridy.length; i++){
    console.log(mojeTridy[i].textContent);
}
```

JavaScript - výpis pomocí idéčka

```
var text1 =
document.getElementById("firstId").textContent;

console.log(text1);
```

44. Výběr podle názvu tagu pomocí getElementByTagName v JavaScriptu

<https://youtu.be/dOt6ANrL-a0>

JavaScript - výpis pole odstavců bez cyklu

```
var mojeOdstavce =
document.getElementsByTagName("p");

console.log(mojeOdstavce[0].textContent);
console.log(mojeOdstavce[1].textContent);
console.log(mojeOdstavce[2].textContent);
```

JavaScript - výpis pole odstavců pomocí cyklu

```
var mojeOdstavce =
document.getElementsByTagName("p");

for(var i = 0; i < 3; i++){
```

```
    console.log(mojeOdstavce[i].textContent);  
}
```

JavaScript - výpis jakkoli velkého pole odstavců pomocí cyklu

```
var mojeOdstavce =  
document.getElementsByTagName("p");  
  
for(var i = 0; i < mojeOdstavce.length; i++){  
    console.log(mojeOdstavce[i].textContent);  
}
```

45. textContent a innerHTML v JavaScriptu

<https://youtu.be/dN78xAaGcPI>

```
<body>  
  <h1>Ahoj, já jsem David</h1>  
  <p><a href="#">První odstavec</a></p>  
  <p>Druhý odstavec</p>  
  <p>Třetí odstavec</p>  
  <p>Čtvrtý odstavec</p>  
  <p>Pátý odstavec</p>  
  <p>Šestý odstavec</p>  
  <p>Sedmý odstavec</p>  
</body>
```

JavaScript

```
var text =  
document.querySelector("p").textContent;  
console.log(text);
```

```
var html =  
document.querySelector("p").innerHTML;  
console.log(html);
```

46. Zjišťujeme atributy jako src, alt, href pomocí JavaScriptu

<https://youtu.be/TRdQ-k1JpF0>

JavaScript:

```
var text1 =  
document.querySelector("a").hasAttribute("href");  
  
console.log(text1);
```

Do HTML přidat:

```

```

JavaScript:

```
console.log(document.querySelector("img").src);  
console.log(document.querySelector("img").alt);
```

47. Vybíráme prvního a posledního potomka pomocí JavaScriptu

<https://youtu.be/qj7Li0VBYSw>

HTML:

```
<body>  
  <h1>Ahoj, já jsem David</h1>  
  <ul id="list">  
    <li>Jedna</li>  
    <li>Dva</li>  
    <li>Tři</li>  
  </ul>  
</body>
```

JavaScript:

```
var text1 =  
document.getElementById("list").firstElementChild.textCon;
```

```
console.log(text1);
```

JavaScript:

```
var text1 =  
document.getElementById("list").lastElementChild.textContent;
```

```
console.log(text1);
```

48. Vytváříme nový tag a přidáváme ho do HTML pomocí**JavaScriptu**

<https://youtu.be/HIJ4gbbhT90>

```
<body>  
  <h1>Ahoj, já jsem David</h1>  
  <ul id="list">  
    <li>Jedna</li>  
    <li>Dva</li>
```

```
</li>Tři</li>
</ul>
</body>
```

JavaScript - přidání za tag

```
var newSpan = document.createElement("span");
var myH1 = document.querySelector("h1");

myH1.appendChild(newSpan,myH1);
```

JavaScript - přidání před tag

```
var newSpan = document.createElement("span");
var myH1 = document.querySelector("h1");

document.body.insertBefore(newSpan,myH1);
```

49. Co všechno umíme a úvod do eventů v JavaScriptu

<https://youtu.be/MdK1pbmwGmU>

- // 1. přijmeme informace od návštěvníka stránky (např. prompt, event)
- // 2. zpracujeme je (podmínky, cykly, funkce, proměnné atd.)
- // 3. vypíšeme je zpět do stránky (umístíme je do HTML nebo změníme CSS)

```
var name = prompt("Jak se jmenuješ?");

if(name === "David"){
    document.querySelector("h1").textContent = "Ahoj," + name;
} else {
    document.querySelector("h1").textContent = "Tebe neznám";
}
```

50. Event kliknutí a jeho zachycení v JavaScriptu

<https://youtu.be/1K2LEXYkFk0>

// Event - zpráva, která se posílá kódu, že se na webu něco stalo. Např. bylo na něco kliknutí, scrollovalo se, změněna velikost okna, kliknuto na určitou klávesu atd.

// Event listener - funkce, která čeká na určitý event a pak něco udělá

```
function welcome(){
    console.log("Vítejte u nás");
}
```



```
// Callback funkce
document.querySelector("h1").addEventListener("click",welcome);

// Anonymní funkce
document.querySelector("h1").addEventListener("click",function(){
    console.log("Vítejte u nás");
});
```

51. Další eventy v JavaScriptu

<https://youtu.be/ZIbudyzQIII>

```
function myEvent(){
    console.log("Event byl spuštěn");
}

document.querySelector("h1").addEventListener("mouseover",myEvent);

// click - kliknutí
// dblclick - dvojité kliknutí
// wheel - kolečko myši
// mousedown - kliknete na tlačítko myši směrem dolů
// mouseenter - najedeme myší
// mouseleave - odejetí myši
// mousemove - jakékoli pohnutí myši
// mouseover - myší nad elementem
```

52. Tlačítkem přidáme novou položku do seznamu

https://youtu.be/im_HbmCW9SM

HTML

```
<body>
  <h1>Ahoj, já jsem David</h1>
  <a href="#" class="btn">Přidat novou položku seznamu</a>
  <ul id="list"></ul>
</body>
```

CSS - nastýlování tlačítka

```
.btn {
    text-decoration: none;
    border: 1px solid black;
    padding: 5px 15px;
    border-radius: 15px;
    background: orange;
```

```
    color: #000;  
}
```

JavaScript

```
var button = document.getElementsByClassName("btn");  
  
button[0].addEventListener("click",clickMe);  
  
function clickMe(){  
  
    var myList = document.getElementById("list");  
  
    var newItem = document.createElement("li");  
  
    newItem.textContent = "Jsem tu nový";  
  
    myList.appendChild(newItem);  
  
}
```

53. Vypisujeme kliknutí na klávesu v JavaScriptu

<https://youtu.be/k4naAHUI8OY>

Seznam eventů:

<https://developer.mozilla.org/en-US/docs/Web/Events>

JavaScript - keydown

```
document.body.addEventListener("keydown",myFunction);  
  
function myFunction(){  
    console.log("Klávesa byla stisknuta");  
}
```

JavaScript - keyup

```
document.body.addEventListener("keyup",myFunction);  
  
function myFunction(){  
    console.log("Klávesa byla stisknuta");  
}
```

JavaScript - keypress

```
document.body.addEventListener("keypress",myFunction);  
  
function myFunction(){
```

```
    console.log("Klávesa byla stisknuta");  
}
```

JavaScript - vypsání konkrétní klávesy

```
document.body.addEventListener("keydown",myFunction);
```

```
function myFunction(event){  
    console.log(event.key);  
}
```

54. Vypisujeme vlastnosti CSS v JavaScriptu pomocí

getComputedStyle

<https://youtu.be/2oyDcOFeMDY>

HTML

```
<body>  
  <h1>Ahoj, já jsem David</h1>  
  <div id="cube"></div>  
</body>
```

CSS

```
#cube {  
  width: 100px;  
  height: 100px;  
  border: 1px solid black;  
  background-color: blue;  
  position: relative;  
  top: 10px;  
  left: 20px;  
}
```

Vypsání vlastnosti left (jednoslovné CSS)

```
var myVar = document.querySelector("#cube");
```

```
var style = getComputedStyle(myVar);
```

```
console.log(parseInt(style.left));
```

Vypsání barvy pozadí (víceslovné CSS)

```
var myVar = document.querySelector("#cube");
```

```
var style = getComputedStyle(myVar);
```

```
console.log(style.backgroundColor);
```

55. JavaScriptem ovlivňujeme HTML i CSS - kostku posouváme všemi směry

https://youtu.be/bR_32r8GKmY

JavaScript

```
var cube = document.querySelector("#cube");

var style = getComputedStyle(cube);

document.body.addEventListener("keydown",myFunction);

function myFunction(event){
    var oldLeft = parseInt(style.left);
    var oldRight = parseInt(style.top);

    if(event.key === "ArrowRight"){
        var newLeft = oldLeft + 10;
        cube.style.left = newLeft + "px";
    } else if(event.key === "ArrowLeft"){
        var newLeft = oldLeft - 10;
        cube.style.left = newLeft + "px";
    } else if(event.key === "ArrowUp"){
        var newRight = oldRight - 10;
        cube.style.top = newRight + "px";
    } else if(event.key === "ArrowDown"){
        var newRight = oldRight + 10;
        cube.style.top = newRight + "px";
    }
}
```

56. Pomocí random generujeme náhodná čísla v JavaScriptu

<https://youtu.be/9El2WJwS4gQ>

```
Math.random();
0.7770256707984451
```

```
Math.random();
0.04536381950235757
```

```
Math.floor(5.3)
5
```

```
Math.floor(5.9)
5
```

Math.floor(5.1)

5

Math.round(5.3)

5

Math.round(5.5)

6

Math.round(5.7)

6

Math.ceil(5.1)

6

Math.ceil(5.3)

6

Math.ceil(5.8)

6

Vygeneruje celé číslo od 1 do 6 (včetně 1 i 6)

Math.ceil(Math.random()*6)

57. Hra v JavaScriptu - jak bude vypadat a pravidla

<https://youtu.be/ZILctoxsIVI>

Ukázka, jak hra bude vypadat a jaká bude mít pravidla. U hry si pak můžete vytvořit jiný design a dodělat responzivitu pro mobily a tablety.

58. Hra v JavaScriptu - nový projekt a stažení obrázků kostky

<https://youtu.be/m8cJpQqUKPg>

Obrázky kostky ke stažení: <http://bit.ly/jskostka>

Podrobný postup:

1. na ploše vytvoříte novou složku s názvem JShra
2. otevřete Brackets a dáte nahoře na liště Soubor -> Otevřít složku
3. najdete složku v počítači a pouze ji označíte (neproklíkáváte se dovnitř) a dáte Otevřít složku
4. V levém panelu v Brackets dáte pravé tlačítko a vyberete z nabídky Nový soubor. Takto založíte index.html, styly.css a script.js
5. Z této adresy stáhnete obrázky kostky: <http://bit.ly/jskostka>
6. Všech 6 obrázků dáte do složky img, kterou umístíte do složky JShra

HTML

```
<!DOCTYPE html>
<html lang="">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title></title>
  <link rel="stylesheet" href="styly.css">
</head>

<body>

</body>
  <script src="script.js"></script>
</html>
```

59. Hra v JavaScriptu - pozadí linear-gradient a hrací plocha

<https://youtu.be/wrpDXBFcdOk>

HTML

```
<div class="allPage">
  <div class="playground">

  </div>
</div>
```

CSS

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* All page */
.allPage {
  background: linear-gradient(#b34e4b, #471f1e);
  height: 100vh;
}

/* Playground */
.playground {
  width: 80%;
  height: 80%;
  background: #d1ccb4;
  position: absolute;
  top: 10%;
```

```
    left: 10%;  
}
```

60. Hra v JavaScriptu - přidání celkového skóre

<https://youtu.be/fGN4wrhKT8g>

HTML

```
<body>  
  <div class="allPage">  
    <div class="playground">  
  
      <div class="totalScore totalScore0">  
        <p id="name-0">Skóre 1. hráče</p>  
        <p id="totalScorePlayer-0">30</p>  
      </div>  
  
      <div class="totalScore totalScore1">  
        <p id="name-1">Skóre 2. hráče</p>  
        <p id="totalScorePlayer-1">40</p>  
      </div>  
  
    </div>  
  </div>  
</body>
```

CSS

```
/* Total score */  
.totalScore {  
  font-size: 30px;  
  color: #471f1e;  
  font-weight: bold;  
}  
  
.totalScore p:nth-of-type(2){  
  font-size: 40px;  
  text-align: center;  
}  
  
.totalScore0 {  
  position: absolute;  
  top: 15%;  
  left: 10%;  
}  
  
.totalScore1 {
```

```
position: absolute;
top: 15%;
left: 70%;
}
```

61. Hra v JavaScriptu - přidání současného skóre

<https://youtu.be/BYkdXk4-Z34>

HTML (do div s třídou playground přidáme tento kód)

```
<div class="currentScore currentScore0">
  <p>Součátné skóre</p>
  <p id="currentScore-0">3</p>
</div>

<div class="currentScore currentScore1">
  <p>Součátné skóre</p>
  <p id="currentScore-1">3</p>
</div>
```

CSS

```
/* Current score */
.currentScore {
  width: 150px;
  height: 150px;
  background-color: #b34e4b;
}

.currentScore p {
  font-size: 20px;
  color: #fff;
  text-align: center;
  padding-top: 20px;
}

.currentScore p:nth-child(2){
  font-size: 40px;
}

.currentScore0 {
  position: absolute;
  top: 40%;
  left: 11%;
}
```



```
.currentScore1 {  
    position: absolute;  
    top: 40%;  
    right: 16%;  
}
```

62. Hra v JavaScriptu - tlačítka k ovládání hry

<https://youtu.be/lnBreNdHVeM>

HTML

```
<div class="middleText">  
    <p class="btn">Hodit kostkou</p>  
    <p class="btn">Podržet skóre</p>  
    <p class="btn">Nová hra</p>  
</div>
```

CSS

```
/* Middle text */
```

```
.middleText {  
    position: absolute;  
    top: 70%;  
    left: 40%;  
    text-align: center;  
    font-size: 35px;  
    color: #fff;  
}
```

```
.btn {  
    border: 2px solid #b34e4b;  
    padding: 5px 10px;  
    border-radius: 10px;  
    margin-bottom: 10px;  
    background-color: #b34e4b;  
}
```

```
.btn:hover {  
    border: 2px solid #D2263D;  
    background-color: #D2263D;  
}
```

63. Hra v JavaScriptu - obrázek kostky

<https://youtu.be/oye70HvRgZE>

HTML

```
<div class="dice">
  
</div>
```

CSS

```
/* Dice */
.dice {
  position: absolute;
  top: 30%;
  left: 42%;
}
```

64. Hra v JavaScriptu - Vynulování hodnot a základní proměnné

<https://youtu.be/Re2pVDwwGRY>

JavaScript

```
// základní proměnné
var totalScore, roundScore, activePlayer, dice;

totalScore = [0,0];
roundScore = 0;
activePlayer = 0;

// vynulování a odstranění kostky
document.getElementById("totalScorePlayer-0").textContent = 0;
document.getElementById("totalScorePlayer-1").textContent = 0;
document.getElementById("currentScore-0").textContent = 0;
document.getElementById("currentScore-1").textContent = 0;

document.querySelector(".diceImage").style.display = "none";
```

65. Hra v JavaScriptu - házíme kostkou pomocí náhodného čísla

<https://youtu.be/WdnpLeNh10Q>

HTML (přidejte třídu k tlačítku Hodit kostkou)

```
<div class="middleText">
  <p class="btn rollDice">Hodit kostkou</p>
  <p class="btn">Podržet skóre</p>
  <p class="btn">Nová hra</p>
</div>
```

JavaScript

```
// měníme obrázek kostky podle náhodného čísla
document.querySelector(".rollDice").addEventListener("click", function(){
    // 1. generujeme náhodné číslo mezi 1 a 6
    var dice = Math.ceil(Math.random()*6);

    // 2. zobrazit správný obrázek
    var diceElement = document.querySelector(".diceImage");
    diceElement.style.display = "block";
    console.log(diceElement.src = "img/" + dice + ".jpg");
});
```

66. Hra v JavaScriptu - Nasčítáváme čísla z kostky

https://youtu.be/-N80_wHIZY0

JavaScript

```
// 3. Nasčítáme čísla z kostky
if (dice !== 1){
    roundScore = roundScore + dice;
    document.getElementById("currentScore-0").textContent = roundScore;
} else {
    // bude hrát další hráč
}
```

67. Hra v JavaScriptu - přepínání hráčů a zvýraznění aktuálního hráče

<https://youtu.be/UJ4MWji5b64>

HTML (přidejte třídu active)

```
<div class="totalScore totalScore0 active">
    <p id="name-0">Skóre 1. hráče</p>
    <p id="totalScorePlayer-0">30</p>
</div>
```

CSS

```
/* Active player */
.active {
    color: #fff;
    border: 2px solid #AD4F34;
    padding: 5px 10px;
    border-radius: 10px;
    background-color: #AD4F34;
}
```

JavaScript

```
function nextPlayer(){
    if(activePlayer === 0){
        activePlayer = 1;
    } else {
        activePlayer = 0;
    }

    roundScore = 0;

    document.getElementById("currentScore-0").textContent = 0;
    document.getElementById("currentScore-1").textContent = 0;

    document.querySelector(".diceImage").style.display = "none";

    document.querySelector(".totalScore0").classList.toggle("active");
    document.querySelector(".totalScore1").classList.toggle("active");
}
```

68. Hra v JavaScriptu - hráči si mohou podržet své skóre

<https://youtu.be/qlfmWkvmNjg>

HTML

```
<div class="middleText">
    <p class="btn rollDice">Hodit kostkou</p>
    <p class="btn holdScore">Podržet skóre</p>
    <p class="btn">Nová hra</p>
</div>
```

JavaScript

```
document.querySelector(".holdScore").addEventListener("click", function(){
    // celkové skóre se vyplní současným skóre
    totalScore[activePlayer] = totalScore[activePlayer] + roundScore;

    // propsání celkového skóre
    document.querySelector("#totalScorePlayer-" + activePlayer).textContent =
totalScore[activePlayer];

    if(totalScore[activePlayer] >= 20){
        document.querySelector("#name-" + activePlayer).textContent = "Vítěz! Vítěz!";
    } else {
        nextPlayer();
    }
});
```

69. Hra v JavaScriptu - Načítání skóre i u druhého hráče

<https://youtu.be/65MYMkYax90>

JavaScript

1. změna

```
if (dice !== 1){
    roundScore = roundScore + dice;
    document.getElementById("currentScore-" + activePlayer).textContent = roundScore;
} else {
    nextPlayer();
}
```

2. změna

```
if(totalScore[activePlayer] >= 30){
    document.querySelector("#name-" + activePlayer).textContent = "Vítez! Vítěz!";
    document.querySelector(".diceImage").style.display = "none";
} else {
    nextPlayer();
}
```

70. Hra v JavaScriptu - funkce pro resetování hry

<https://youtu.be/pcXr1fH3PsI>

JavaScript

newStart();

```
function newStart(){
    totalScore = [0,0];
    roundScore = 0;
    activePlayer = 0;
```

// vynulování a odstranění kostky

```
document.getElementById("totalScorePlayer-0").textContent = 0;
document.getElementById("totalScorePlayer-1").textContent = 0;
document.getElementById("currentScore-0").textContent = 0;
document.getElementById("currentScore-1").textContent = 0;
```

//skrytí kostky

```
document.querySelector(".diceImage").style.display = "none";
```

// texty do původního stavu

```
document.querySelector("#name-0").textContent = "Skóre 1. hráče";
document.querySelector("#name-1").textContent = "Skóre 2. hráče";
```

// vrátíme zvýraznění aktivního hráče k prvnímu a u druhého odstraníme

```

document.querySelector(".totalScore0").classList.add("active");
document.querySelector(".totalScore1").classList.remove("active");
}

```

Zavolá funkci newStart a vše vrátí do původního stavu, když se klikne na tlačítko

Nová hra:

```
document.querySelector(".newGame").addEventListener("click", newStart);
```

71. Hra v JavaScriptu - máme vítěze, tak zastavíme hru

<https://youtu.be/OWJ10Vqq1P8>

JavaScript

1. změna

// základní proměnné

```
var totalScore, roundScore, activePlayer, dice, playGame;
```

2. změna

```
function newStart(){
  totalScore = [0,0];
  roundScore = 0;
  activePlayer = 0;
  playGame = true;
  ...
}
```

3. změna

// měníme obrázek kostky podle náhodného čísla

```
document.querySelector(".rollDice").addEventListener("click", function(){
```

```
  if(playGame){
```

```
    // 1. generujeme náhodné číslo mezi 1 a 6
```

```
    var dice = Math.ceil(Math.random()*6);
```

```
    // 2. zobrazit správný obrázek
```

```
    var diceElement = document.querySelector(".diceImage");
```

```
    diceElement.style.display = "block";
```

```
    console.log(diceElement.src = "img/" + dice + ".jpg");
```

```
    // 3. Nasčítáme čísla z kostky
```

```
    if (dice !== 1){
```

```
      roundScore = roundScore + dice;
```

```
      document.getElementById("currentScore-" + activePlayer).textContent = roundScore;
```

```
    } else {
```

```
      nextPlayer();
```

```
    }
```

```
  }
```

```
});
```

4. změna

```
document.querySelector(".holdScore").addEventListener("click", function(){
    if(playGame){
        // celkové skóre se vyplní současným skóre
        totalScore[activePlayer] = totalScore[activePlayer] + roundScore;

        // propsání celkového skóre
        document.querySelector("#totalScorePlayer-" + activePlayer).textContent =
        totalScore[activePlayer];

        if(totalScore[activePlayer] >= 30){
            document.querySelector("#name-" + activePlayer).textContent = "Vítěz! Vítěz!";
            document.querySelector(".diceImage").style.display = "none";
            playGame = false;
        } else {
            nextPlayer();
        }
    }
});
```

72. Závěrečné video

<https://youtu.be/WhQkkRy1eVc>

73. Objektově orientovaný javascript (OOP) - konstruktor a objekty

<https://youtu.be/74K6y3Hmplo>

// OOP - objektově orientované programování

// Tvorba objektů bez konstruktor

```
var david = {
    name: "David",
    yearBirth: 1980,
    job: "programátor"
}
```

```
var anna = {
    name: "Anna",
    yearBirth: 1990,
    job: "designerka"
}
```

```
var markus = {
    name: "Markus",
```

```
    yearBirth: 1981,  
    job: "učitel"  
}
```

// Tvorba objektů s konstruktorem

```
var Person = function(name, yearBirth, job){  
    this.name = name;  
    this.yearBirth = yearBirth;  
    this.job = job;  
}
```

```
var david = new Person("David", 1980, "programátor");  
var anna = new Person("Anna", 1990, "designerka");  
var markus = new Person("Markus", 1981, "učitel");
```

```
console.log(david.name);  
console.log(david.yearBirth);  
console.log(david.job);
```

```
console.log(anna.name);  
console.log(anna.yearBirth);  
console.log(anna.job);
```

```
console.log(markus.name);  
console.log(markus.yearBirth);  
console.log(markus.job);
```

74. Objektově orientovaný javascript (OOP) - konstruktor a funkce, return

<https://youtu.be/X5TLbdAJzL0>

// OOP - objektově orientované programování

// Atribut - vlastnosti nebo data

// Metody - schopnosti

// Tvorba objektů s konstruktorem

```
var Person = function(name, yearBirth, job){  
    this.name = name;  
    this.yearBirth = yearBirth;  
    this.job = job;  
    this.calcAge = function(){  
        return 2020 - this.yearBirth;  
    };  
    this.adult = function(){
```



```
        console.log("Je on/ona dospělý? " + (2020-this.yearBirth > 18));
    }
}
```

```
var david = new Person("David", 1980, "programátor");
var anna = new Person("Anna", 1990, "designerka");
var markus = new Person("Markus", 1981, "učitel");
```

```
console.log(david.calcAge());
console.log(anna.calcAge());
console.log(markus.calcAge());
```

```
david.adult();
anna.adult();
markus.adult();
```

75. Objektově orientovaný javascript (OOP) - konstruktor a prototype

<https://youtu.be/n5SkxELNHZs>

```
// OOP - objektově orientované programování
// Atribut - vlastnosti nebo data
// Metody - schopnosti
```

```
// Tvorba objektů s konstruktorem
var Person = function(name, yearBirth, job){
    this.name = name;
    this.yearBirth = yearBirth;
    this.job = job;
}
```

```
Person.prototype.calcAge = function(){
    return 2020 - this.yearBirth;
};
```

```
var david = new Person("David", 1980, "programátor");
console.log(david.calcAge());
```

Vypište si pouze objekt david a rozklikejte si jednotlivé prototypy, ať vidíte, že skutečně existují

```
console.log(david)
```

76. Objektově orientovaný javascript (OOP) - procvičování konstruktoru, prototype a objekty

<https://youtu.be/wNP26PgByO8>

```
var Car = function(wheels, color, brand, tank, distance){
  this.wheels = wheels;
  this.color = color;
  this.brand = brand;
  this.tank = tank;
  this.distance = distance;
  this.calculateDistance = function(){
    return this.tank * this.distance;
  };
}
```

```
Car.prototype.type = "řízeno lidmi";
```

```
var car1 = new Car(4, "red", "toyota", 40, 6);
var car2 = new Car(4, "blue", "hyundai", 30, 5);
var car3 = new Car(4, "green", "skoda", 35, 5);
```

```
console.log(car1.wheels);
console.log(car2.color);
console.log(car3.brand);
```

```
console.log(car1.type);
console.log(car2.type);
console.log(car3.type);
```

```
console.log(car1.calculateDistance());
```

77. Objektově orientovaný javascript (OOP) - pojmy a prototype v praxi

<https://youtu.be/QOj8pZAcjJY>

```
// Atribut
// Metoda
// Instance, objekt
// Třída, konstruktor
// Prototype
```

```
// Třída, konstruktor
var Car = function(wheels, color, brand, tank, distance){
  this.wheels = wheels; // Atribut
  this.color = color; // Atribut
```

```

    this.brand = brand; // Atribut
    this.tank = tank; // Atribut
    this.distance = distance; // Atribut
    this.calculateDistance = function() { // Metoda
        return this.tank * this.distance;
    };
}

```

// **Prototype**

```
Car.prototype.type = "řízeno lidmi";
```

// **Instance, objekt**

```

var car1 = new Car(4, "red", "toyota", 40, 6);
var car2 = new Car(4, "blue", "hyundai", 30, 5);
var car3 = new Car(4, "green", "skoda", 35, 5);

```

```

console.log(car1);
console.log(car1.hasOwnProperty("wheels"));
console.log(car1.hasOwnProperty("calculateDistance"));

```

Takto to vypadá v konsoli, když si vypíšete car1

```

Car {wheels: 4, color: "red", brand: "toyota", tank: 40, distance: 6, ...}
  brand: "toyota"
  calculateDistance: f ()
  color: "red"
  distance: 6
  tank: 40
  wheels: 4
  __proto__:
    type: "řízeno lidmi"
    constructor: f (wheels, color, brand, tank, distance)
    __proto__:
      constructor: f Object()
      hasOwnProperty: f hasOwnProperty()
      isPrototypeOf: f isPrototypeOf()
      propertyIsEnumerable: f propertyIsEnumerable()
      toLocaleString: f toLocaleString()
      toString: f toString()
      valueOf: f valueOf()
      __defineGetter__: f __defineGetter__()
      __defineSetter__: f __defineSetter__()
      __lookupGetter__: f __lookupGetter__()
      __lookupSetter__: f __lookupSetter__()
      get __proto__: f __proto__()
      set __proto__: f __proto__()

```

78. Objektově orientovaný javascript (OOP) - další možnost, jak vytvořit objekt

<https://youtu.be/-EMXgK0Sk9A>

```
var personSchema = {
  ageCalculator: function(){
    console.log(2020 - this.yearBirth);
  }
};

var david = Object.create(personSchema);
david.name = "David";
david.yearBirth = 2000;
david.job = "hacker";

var anna = Object.create(personSchema,{
  name: {value: "Anna"},
  yearBirth: {value: 2010},
  job: {value: "žádná práce"}
});

console.log(david.name);
david.ageCalculator();
console.log(anna.name);
anna.ageCalculator();
```

79. Objektově orientovaný javascript (OOP) - primitivní datové typy a objekty

https://youtu.be/_oComtP7YOE

// Primitivní datové typy = string, number, boolean, undefined, null

```
var name = "David";
var age = 40;
var adult = false;
var eyeColor = null;
```

// Objekty = vše ostatní

```
var a = 10; //a je 10
var b = a; //b je 10
a = 20; //a je 20
console.log(a);
console.log(b);
```

```
var object1 = {
  name: "David",
  age: 40,
  job: "hacker"
}

var object2 = object1;
object1.name = "Anna";
object1.age = 25;
object1.job = "učitelka";

console.log(object1);
console.log(object2);
```

```
// funkce
var age = 30;
var object = {
  name: "David",
  city: "České Budějovice"
};
```

```
function change(a,b){
  a = 60;
  b.city = "Praha";
}
```

```
change(age,object);

console.log(age);
console.log(object.city);
```

80. Objektově orientovaný javascript (OOP) - funkce jako argument

<https://youtu.be/wUGuRPQfzyQ>

```
var roky = [1950,1975,1920,2010,2015];

function poleVypocet(pole,funk){
  var vyslednePole = [];
  for(var i = 0; i < pole.length; i++){
    vyslednePole.push(funk(pole[i]));
  }
  return vyslednePole;
};
```

// 1. funkce

```
function vypoctiVek(neco){  
    return 2020 - neco;  
}
```

```
var vek = poleVypocet(roky,vypoctiVek);  
console.log(vek);
```

// 2. funkce

```
function jePlnolety(neco2){  
    return neco2 > 18;  
}
```

```
var vysledek2 = poleVypocet(vek, jePlnolety);  
console.log(vysledek2);
```

81. Objektově orientovaný javascript (OOP) - Funkce vracející funkci

<https://youtu.be/E9AspfppQpl>

```
function pohovorOtazky(prace){  
    if(prace === "hacker"){  
        return function(jmeno){  
            console.log(jmeno + ", uveďte základní typy hackerů.");  
        }  
    } else if (prace === "programator"){  
        return function(jmeno){  
            console.log(jmeno + ", jaké znáte programovací jazyky?");  
        }  
    } else {  
        return function(jmeno){  
            console.log(jmeno + ", co děláte za práci?")  
        }  
    }  
}
```

```
var otazka1 = pohovorOtazky("hacker");  
otazka1("David");  
var otazka2 = pohovorOtazky("programator");  
otazka2("Anna");  
var otazka3 = pohovorOtazky("ucitel");  
otazka3("Jana");
```

// další zápis

```
pohovorOtazky("hacker")("Markus");
```

82. Objektově orientovaný javascript (OOP) - IIFE = funkce, která se spustí při načtení kódu

<https://youtu.be/yzUXSgL8ySk>

// IIFE = Immediately Invoked Function Expression = funkce, která se okamžitě spustí při načtení kódu;

// běžná funkce

```
function hra(){  
    var skore = Math.random() * 10;  
    console.log(skore);  
}
```

//hra();

//hra();

//hra();

//IIFE funkce

```
(function hra2(){  
    var skore = Math.random() * 10;  
    console.log(skore);  
})();
```

//IIFE funkce s parametrem

```
(function hra2(jmeno){  
    var skore = Math.random() * 10;  
    console.log(jmeno + ", vaše skóre je: " + skore);  
})("Anna");
```

83. Objektově orientovaný javascript (OOP) - Lexical scoping

<https://youtu.be/dz3SIGY0Pso>

// Lexical scoping

```
function mojeFuncce(){  
  
    var jmeno = "David";  
  
    function zobraz(){  
  
        console.log(jmeno);  
    }  
  
    zobraz();  
};
```

```
mojeFunce();
```

// další příklad - vyhodí chybu

```
function test(){  
    var x = 10;  
  
}  
console.log(x);
```

84. Objektově orientovaný javascript (OOP) - Closure

<https://youtu.be/cWRRgTjmTno>

// Closure

```
var test = function mojeFunkce(){  
    var x = 66;  
    return x;  
}();
```

```
console.log(test); // vypíše 66  
console.log(x); // vyhodí chybu
```

// další příklad Closure

```
var test = function mojeFunkce(){  
  
    var x = 66;  
  
    return {  
        verejne: function(){  
            return x;  
        }  
    }  
}();
```

```
console.log(test.x);  
console.log(test.verejne());  
console.log(x);
```


85. Objektově orientovaný javascript (OOP) - Closure pokračování

<https://youtu.be/QpdC8Gj9f0o>

// Closure - toto funguje

```
var test = function mojeFunkce(){  
    var x = 66;  
    return x;  
}();
```

```
console.log(test);
```

// Closure - toto také funguje

```
function mojeFunkce(){  
    var x = 66;  
    return x;  
};
```

```
var test = mojeFunkce();  
console.log(test);
```

86. Objektově orientovaný javascript (OOP) - Call, Apply a Bind

<https://youtu.be/-KHkGK8EuJI>

// Call - umožňuje půjčit si metodu, kterou má jiný objekt

```
var david = {  
    name: "David",  
    yearBirth: 1980,  
    job: "hacker",  
    calculateAge: function(){  
        console.log(2020 - this.yearBirth);  
    }  
}
```

```
var anna = {  
    name: "Anna",  
    yearBirth: 2000,  
    job: "programmer"  
}
```

```
david.calculateAge.call(anna); // vypíše 20
```

// Call s parametrem

```
var david = {  
  name: "David",  
  job: "hacker",  
  calculateAge: function(yearBirth){  
    console.log(2020 - yearBirth);  
  }  
}
```

```
var anna = {  
  name: "Anna",  
  job: "programmer"  
}
```

david.calculateAge.call(anna, 1990); **// vypíše 30**

// Apply - stejné jako Call, jen argumenty potřebuje jako pole

```
var david = {  
  name: "David",  
  job: "hacker",  
  calculateAge: function(yearBirth){  
    console.log(2020 - yearBirth);  
  }  
}
```

```
var anna = {  
  name: "Anna",  
  job: "programmer"  
}
```

david.calculateAge.apply(anna, [2000]); **// vypíše 20**

// Bind - umožňuje zadat jeden argument a až poté zadat další argument

```
var david = {  
  calculateAge: function(name, yearBirth){  
    return name + " " + (2020 - yearBirth);  
  }  
}
```

```
var anna = {  
  name: "Anna",  
  job: "programmer"  
}
```

var test = david.calculateAge.bind(anna, "Anna"); **//zadáme argument name**
console.log(test(2000)); **//zadáme argument yearBirth**

