

1. Co je (či není) JavaScript a k jakým účelům se používá?

- JavaScript je **skriptovací jazyk interpretovaný na straně prohlížeče** (on client side)
- Bývá označován také jako **ECMA script**
- Navzdory slovu Java v názvu není totožný s jazykem Java, ale používá podobnou syntax příkazů
- Je využíván pro vytváření dynamických a interaktivních webových aplikací – změna vzhledu i obsahu stránky na základě akcí uživatele, ošetření formulářových vstupů, grafické efekty...

2. Jakými způsoby je možné propojit JavaScript s webovou stránkou?

- Vložit kód do tagu
`<script></script>`
- Připojit externí soubor
`<script type='text/javascript' src='soubor.js'></script>`
- Vložit kód jako reakci na události
`<button onclick="alert('Ahoj');">Klikni</button>`

3. Jaká základní pravidla platí pro psaní javascriptových kódů?

- JavaScript je case sensitive – tj. záleží na velikosti písmen
- Příkazy musí být zakončeny středníkem, pokud je jich více na jednom řádku (zakončení středníkem je vhodné i u jednotlivých příkazů)
- Blok příkazů je uvozen složenými závorkami - {...}
- Komentáře v JS:
`// jednořádkový komentář`
`/* víceřádkový komentář */`

4. Jak je možné v JavaScriptu deklarovat proměnné? Uveďte na příkladech datové typy, které je možné využívat v JavaScript.

```
var x = 0; // s přiřazením hodnoty
var x; // bez přiřazení hodnoty

var x = 8; // celé číslo (typ integer)
var y = 8.53; // reálné číslo (typ float)
var z = 0.1e3; //zápis desetinného čísla pomocí exponentu
var txt = 'Znakový řetězec'; // typ string
var student = false; // logický typ boolean
```

5. Vytvořte a popište jednoduchou funkci zjišťující, zda je zadané číslo sudé, liché, nebo nula. Jak lze otestovat volání této funkce v konzoli?

```
function sudaLicha(num) {
    if (num == 0) {
        return 'Nula';
    } else if (num % 2 == 1) {
        return 'Liché';
    } else {
        return 'Sudé';
    }
}
```

Příklad volání funkce:

```
console.log(sudaLicha(5));
```

6. Jaký význam mají operátory %, ++, --? Co je tzv. **zkrácené přiřazování**?

- % operátor **modulo**, zjištění zbytku po celočíselném dělení
- ++ inkrementace, zvýšení hodnoty proměnné (+ 1)
- -- dekrementace, snížení hodnoty proměnné (- 1)
- zkrácené přiřazování – např. `x += 1`; (stejně jako `x = x + 1`)

7. Jak je možné spojovat znakové řetězce? Jakou roli plní v řetězcích znak obrácené (zpětné) lomítko?

- Spojování řetězců pomocí operátoru +
`var jmeno = 'Karel';`
`var prijmeni = 'Novák';`
`var celejmeno = jmeno + ' ' + prijmeni;`
- Obrácené (zpětné) lomítko používáme:
- pro zápis znaku, který má v programovém kódu speciální význam
`alert('Citát píšeme v '\u0022uvozovkách\u0022');`
- vložení tzv. řídicího znaku
`alert('Po tomto řídicím znaku \n začíná nový řádek');`

8. Jaký je rozdíl mezi porovnávacími operátory `==` a `===`? Co znamená operátor `!=`?

- Porovnávací operátory používáme v podmínkách.
- `==` porovná hodnoty, ale nezohledňuje datový typ (např. `x == 0` je vyhodnoceno stejně jako `x == "0"`)
- `===` zohledňuje i datový typ (např. `x === 0` je vyhodnoceno jinak než `x === "0"`; v prvním případě je nula typu celé číslo - integer, ve druhém znakový řetězec – string)
- `!=` znamená nerovná se

Napište jednoduchý příklad tzv. **podmíněného operátoru** (někdy také ternárního).

```
var vysledek = x == 0 ? 'x je nula' : 'x není nula';
```

9. Napište praktický příklad použití tzv. větvené podmínky **switch**.

```
switch (den) {  
  case 1: alert('pondělí');  
    break;  
  case 2: alert('úterý');  
    break;  
  case 3: alert('středa');  
    break;  
  default: alert('jiný den');  
}
```

10. Jaký je rozdíl mezi cykly **for**, **while** a **do .. while**?

- Cyklus **for** používáme pro předem daný počet opakování
- Cyklus **while** používáme pro předem neznámý počet opakování; podmínka je vyhodnocována na začátku cyklu
- Cyklus **do ... while** používáme pro předem neznámý počet opakování; podmínka je vyhodnocována na konci cyklu

11. Uved'te příklad funkce, která pomocí cyklu **for** vypíše seznam čísel od 20 po 10. Připojte příklad volání funkce tak, aby se výstup z ní objevil na webové stránce.

```
function seznamCisel() {  
  var vypis = '<ul>';  
  for (var i = 20; i >= 10; i--)  
  {  
    vypis += '<li>'+i+'</li>';  
  }  
  vypis += '</ul>';  
  return vypis;  
}  
  
document.getElementById('seznam').innerHTML += seznamCisel();
```

12. Uved'te p'íklad funkce, která pomocí cyklu **while** vypíše čísla od minima po maximum. Připojte p'íklad volání funkce tak, aby se výstup z ní objevil na webové stránce.

```
function vypisCisel(min,max) {
    var i=min;
    var vystup='';
    while(i<=max){
        vystup += '<p>' + i+ '</p>';
        i++;
    }
    return vystup;
}

document.getElementById('cisla').innerHTML += vypisOdmocnin(8,20);
```

13. Popište, co provede následující kód:

```
document.getElementById('btnOK').addEventListener('click',
function(){
    var mujNadpis = prompt('Zadej text nadpisu');
    document.getElementById('nadpis').innerHTML = mujNadpis;
    document.getElementById('nadpis').style.color = 'red';
});
```

14. Popište, co provede následující kód:

```
document.getElementById('foto').addEventListener('mouseenter',
function(){
    this.src = 'smile.png';
    this.style.border = '5px solid black';
});

document.getElementById('foto').addEventListener('mouseleave',
function(){
    this.src = 'cry.png';
    this.style.border = 'none';
});
```

15. Popište, k čemu dojde po vykonání těchto řádků kódu v JS:

- **prompt** – umožní zadat hodnotu do editačního pole „vyskakovacího“ (popup) okna
`var heslo = prompt('Zadej heslo');`
- **alert** – vypíše upozornění do „vyskakovacího“ (popup) okna
`alert('Nezadal jsi heslo');`
- **console.log** – vypíše obsah proměnné do tzv. konzole (zobrazí se po stisku F12 v prohlížeči)
`console.log(heslo);`

16. Popište, k čemu dojde po vykonání těchto řádků kódu v JS:

- **Math.random()** – vygeneruje náhodné desetinné číslo od 0 do 1
- **Math.ceil()** – zaokrouhlí číslo směrem nahoru
- Do proměnné **hod** se vygeneruje náhodné číslo v rozsahu 1 až 6
`var hod = Math.ceil(Math.random()*6);`
- Do proměnné **d** se uloží nový datový objekt, který obsahuje datové a časové informace odpovídající aktuálnímu systémovému času
- **d.getMonth()** – vrací číslo měsíce, který je nastaven v datovém objektu **d** (v rozsahu 0 až 11)
`var d = new Date();`
`console.log(d.getMonth()+1);`