

Interaktivita

SSPŠ
WBA
2025 / 2026
Ing. Matěj Cajthaml



Webové stránky vs aplikace

- Stránky => HTML+CSS, žádná interaktivita
- Aplikace => interaktivita, možnost ovládat stránky
- Nejčastěji:
 - JavaScript
 - Formuláře

Formuláře

- Vstupní pole na webu k zadávání informací
- Můžeme odeslat (viz později), server zpracuje
- Jak reagovat => MVOP, OSE, samostudium
- Vstupní pole => text, e-mail, výběr z prvků, tlačítka, ...

Formulář

- Tag form
- Obsahuje vstupní prvky, které s ním souvisí
- Atributy:
 - action – kam se data odesílají
 - method – jakým způsobem se odešlou
- Prvky jaké znáte odjinud
- Pouze podmnožina z možných
- Validace na klientu (a na serveru)
- Tag input, atribut type
- Některé prvky se stylují dle prohlížeče

Další atributy

- placeholder – hodnota, která se zobrazí jako pozadí, když je vstupní pole prázdné
- value – hodnota, která se do pole přímo nastaví
- name – jméno, pod kterým se data pošlou (klíč–hodnota)

Textový vstup, type=text

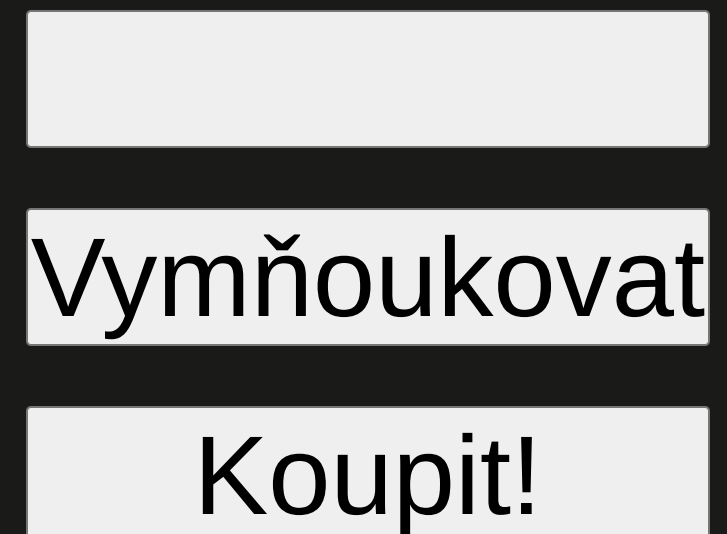
- Vše na jednom řádku, jakýkoliv text, nemá validaci

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formuláře</title>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8   <style>
9     * {margin:0;padding:0;box-sizing:border-box;}
10    body {
11      font-size: 3.5em;
12
```

Vstup, type=button

- Vlastní akce ve formuláři, value je jméno, cos se vykreslí

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8     <style>
9         * {margin:0;padding:0;box-sizing:border-box;}
10        body {
11            font-size: 3.5em;
12
```



Vstup, type=submit

- Odešle data formuláře dle jeho nastavení, value je hodnota, která se vykreslí

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formuláře</title>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8   <style>
9     * {margin:0;padding:0;box-sizing:border-box;}
10    body {
11      font-size: 3.5em;
12
```

Submit

Vymňoukovat

Button

- `input, type=button`
- `button`
- `input, type=submit`
- `input, type=reset`
- Do button jde dávat další obsah – např. ikona

Vstup, type=email

- Textové pole, validuje na klientu e-mail

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formuláře</title>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7
8   <style>
9     * {margin:0;padding:0;box-sizing:border-box;}
10    body {
11      font-size: 3.5em;
12
```

OTÁZKA

Jak vypadá platný e-mail?

Vstup, type=password

- Nelze zkopírovat

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width
7
8     <style>
9         * {margin:0;padding:0;box-sizing
10         body {
11             font-size: 3.5em;
12
```

Vaše heslo od Roblox účtu

●●●●●●●●●●

Vstup, type=date

- Výběr data, různá validace, další možnosti je datetime a datetime-local

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formuláře</title>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8   <style>
9     * {margin:0;padding:0;box-sizing:border-box;}
10    body {
11      font-size: 3.5em;
12
```

mm/dd/yyyy 

mm/dd/yyyy 

mm/dd/yyyy 

Odeslat

Vstup, type=file

- Výběr lokálních souborů

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8     <style>
9         * {margin:0;padding:0;box-sizing:border-box;}
10        body {
11            font-size: 2.5em;
12            color: white;
```

Choose File No file chosen

Choose Files No file chosen

Choose Files No file chosen

Odeslat

Vstup, type=checkbox

- Zaškrtávací políčko, jde odškrtnout

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8     <style>
9         * {margin:0;padding:0;box-sizing:border-box;}
10        body {
11            font-size: 2.5em;
12            color: white;
```

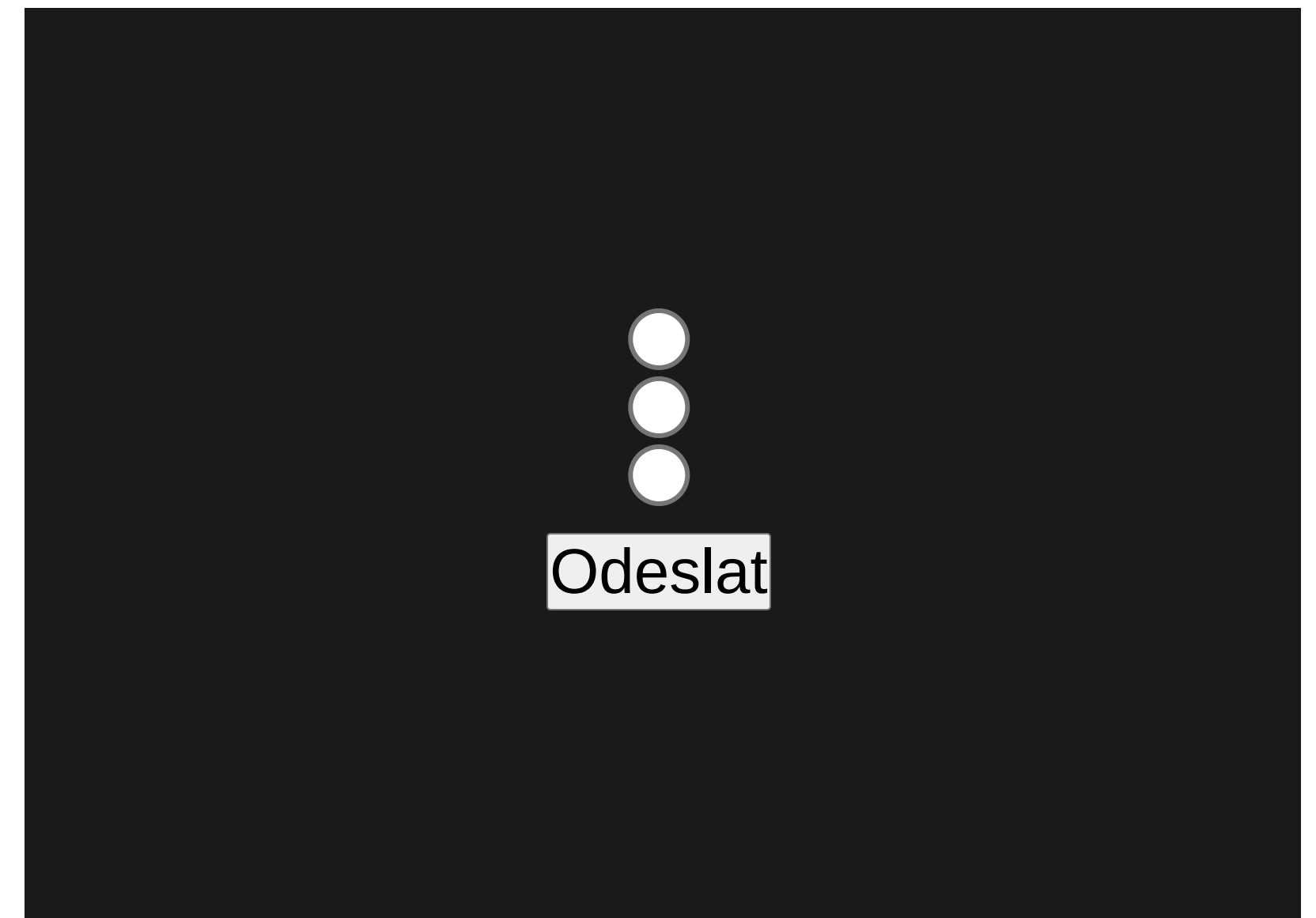


Odeslat

Vstup, type=radio

- Zaškrtávací políčko, jde zaškrtnout jen jiné v tomto "jméně"

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8     <style>
9         * {margin:0;padding:0;box-sizing:border-box;}
10        body {
11            font-size: 2.5em;
12            color: white;
```



Dlouhý vstup

- Lze zalomit řádek, základní hodnota jako obsah tagu

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7
8     <style>
9         * {margin:0;padding:0;box-sizing:border-box;}
10        body {
11            font-size: 3.5em;
12            color: white;
```

:)

Sem vyplňte všechny
soukromé údaje

Popisky

- Tag label
- Jde ke každému prvku přidat
- Odkazujeme pomocí atribut for, ten ukazuje na id vstupu
- Některé prvky v podstatě nutné - type=checkbox, type=radio

Popisek

- Při správném spojení se při kliku daný vstup "zaměří"

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8     <style>
9         * {margin:0;padding:0;box-sizing:border-box;}
10        body {
11            font-size: 3.5em;
12            color: white;
```

Popis něčeho

Jméno

- ☐ Yoko
- ☐ Bambi
- ☐ Yenda

Výběr z možností

- Tag select & option

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formuláře</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8     <style>
9         * {margin:0;padding:0;box-sizing:border-box;}
10        body {
11            font-size: 3.5em;
12            color: white;
```

Známka

Výborná ▾

Oblíbení pejsci

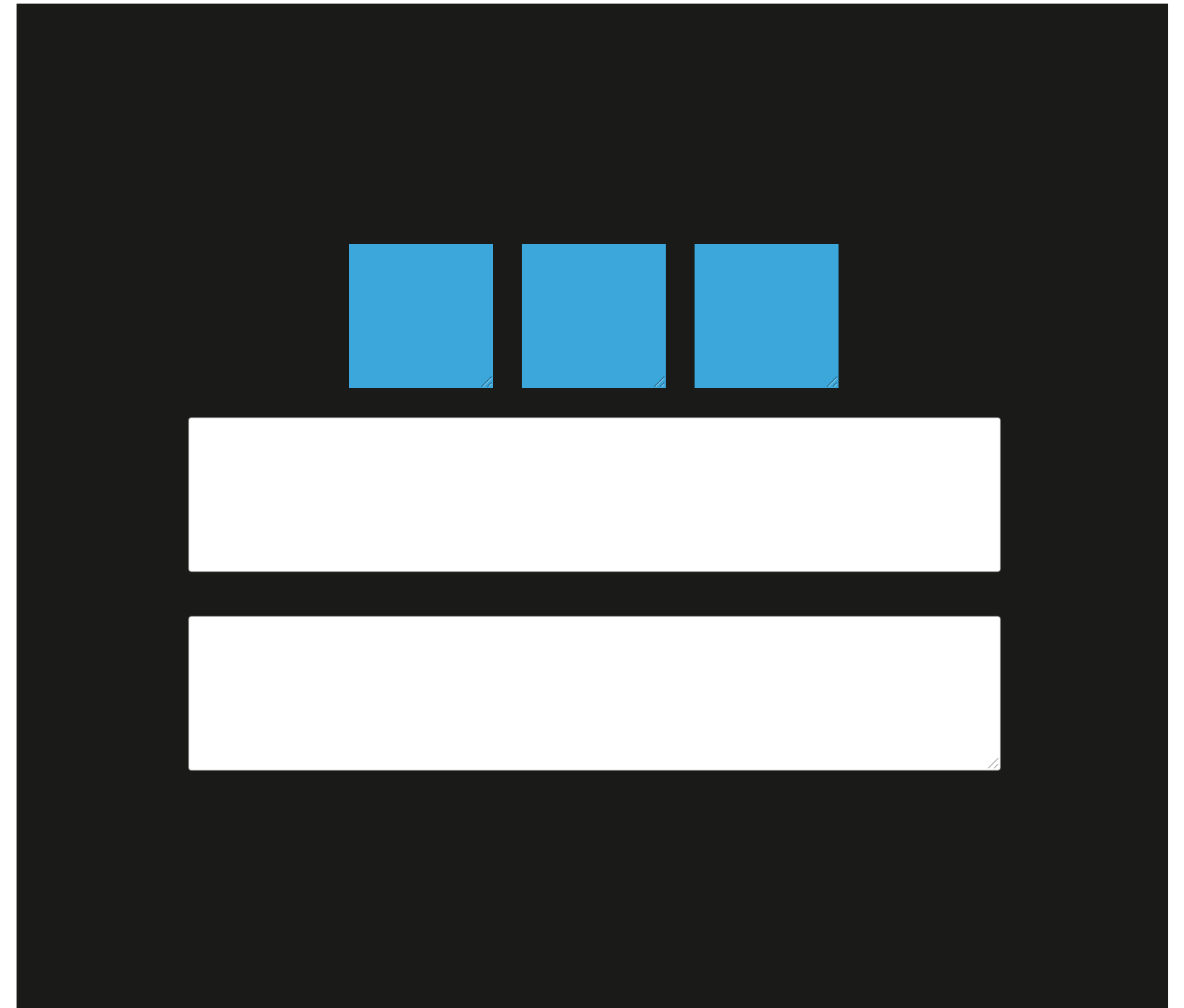
Hafan 1
Hafan 2
Rost'a určitě ne

Další vstupní pole

- `input, type=color`
- `input, type=month`
- `input, type=number`
- `input, type=range`
- `input, type=search`
- `input, type=tel`
- `input, type=url`
- `input, type=week`
- `input, type=hidden`
- `output`
- `datalist`

Vlastnost resize

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Formuláře</title>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="wid
7
8      <style>
9          * {padding:0;margin:0;box-sizi
10
11      body {
12          height: 100vh;
13          display: flex;
14          flex-direction: column;
15          justify-content: center;
16          align-items: center;
```



Nové selektory

- `:checked`
- Když je radio/checkbox zaškrtnut
- `:has(sel)`
- Jestli daný prvek obsahuje daný selektor

PRÁCE

Zkusme si nadesignovat vlastní checkbox.

Odesílání dat

- Data se posílají přes protokol HTTP
- HTTP má metody – jak se data předávají
- Pro nás dva typy – GET a POST uvnitř atributu method
- GET = data se předávají uvnitř URL v parametrech
- POST = data se předávají uvnitř požadavku
- Mimo to můžeme nastavovat i formát dat – atribut enctype:
- multipart/form-data
- application/x-www-form-urlencoded
- Pro odesílání souborů nutné form-data

OTÁZKA

Jaký je limit URL adresy na délku?

Metody

GET

- Data v URL
- Parametr – klíč & value
- Klíč dle atributu name
- Některé prvky nelze předávat

POST

- Uvnitř požadavku
- Vidět např. v DevTools
- "Více skrytá"

OTÁZKA

Co je lepší na předáváníí? GET či POST?
Co je bezpečnější?

GET

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formuláře</title>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="wid
7
8   <style>
9     * {padding:0;margin:0;box-sizi
10    body {
11      font-size: 3.5em;
12      min-height: 100vh;
13      display: flex;
14      justify-content: center;
15      align-items: center;
```

Jméno

Věk

Nejoblíbenější kočka

- ☐ Yoko
- ☐ Bambi
- ☐ Yenda

POST

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formuláře</title>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="wid
7
8   <style>
9     * {padding:0;margin:0;box-sizi
10    body {
11      font-size: 3.5em;
12      min-height: 100vh;
13      display: flex;
14      justify-content: center;
15      align-items: center;
```

Jméno

Věk

Nejoblíbenější kočka

- ☐ Yoko
- ☐ Bambi
- ☐ Yenda



JavaScript

Aneb jak skriptovat stránky

JavaScript

- Dále často již jen JS
- Skriptovací jazyk
- Tvořen pro interakci na webových stránkách
- Např. klikneme a otevře se jiný prvek
- Každý rok nový standard, čistě teoreticky již jen tzv. ECMAScript
- Poslední roky velmi stabilní a použitelný na cokoliv
- V rámci mimo web programovací jazyk
- Pomocí runtime – např. NodeJS
- A i v jiných SW mimo web

Historie JS a rozdíly

- 30+let
- Vytvořen za pár dní => rychlý vývoj
- Špatný začátek, "táhne se to s ním"
- Zejména kvůli zpětné kompatibilitě => webové prohlížeče NECHTĚJÍ rozbité stránky
- Java jen z důvodu marketingového tahu
- Nutné jiné přemýšlení
- Spousta možností a nutných kontrol

Opakování základu programování

- Typy jazyků
- kompilované vs interpretované
- Typování
- statické vs dynamické
- Datové typy:
- čísla, typy čísla
- char, string
- pole, listy
- objekty
- Proměnné a konstanty

Způsoby zápisu

- Přípona .js
- Nebo přímo v HTML
- Tag skript
- Doporučuji dávat na konec body (NE ZA BODY), nebo v head

Zápisy

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>A</title>
5
6      <script src="link/to/script.js"></script>
7  </head>
8  <body>
9      <p>Ahoj</p>
10     <script>
11         let a = "...";
12     </script>
13 </body>
14
15 </html>
```

Komentáře

```
1 // Komentář
2
3 /*
4 komentář
5 */
```

Konzole

- Prozatím píšeme jako obyčejný program
- Skriptování ignorujeme
- A pracujeme pouze s konzolí v DevTools
- Globální objekt, funkce

```
1 console.log("Ahoj!");  
2 console.log("Yoko", "neví, já nevím");
```

Fakta

- JS je jedno-vláknový jazyk, běží sekvenčně postupně, jak se načte skript v tagu script.
- V JS je možné čekat, např. na událost, a potom zavolat část kódu.
- V konzoli jsou vidět i chyby, které se stanou až při interpretaci kódu.
- V konzoli je možné psát kód jako v tagu script.
- (Skoro) každý příkaz v JS končí středníkem, ale nejsou často vyžadovány, ale mohou způsobit chyby.
- Klíčová slova jsou slova, která jsou registrována jazykem a nemohou být použity pro jména uživatelsky definovaných věcí.
- Když v kódu nastane chyba, kód v daném bloku již nepokračuje.

Proměnné

- Tři typy: var, const a let
- Var je zastaralý a nepoužíváme ho
- Určujeme jméno, datový typ je vždy dle dat a může obsahovat cokoliv
- Deklarace je unikátní pro blok, const je neměnný (ale hodnota se může změnit)
- Viz referenční vs hodnotový typ.

```
1 let name = "Yoko";  
2 console.log(name);  
3 name = "Bambi";  
4 console.log(name);  
5 name = "Yoko";  
6 console.log(name);  
7  
8 let age;
```

OTÁZKA

Vždy píšeme názvy konzistentně:
camelCase/PascalCase/snake_case
Anglicky.
Proč?

Typy

- Číslo – celé, s čárkou – Number
- Text – string, píšeme do ", ' či `
- Pole – jedná se o list
- Funkce
- Objekt

- Hodnota undefined – není definováno
- Hodnota null – je definováno, ale záměrně prázdné

Typy

```
1 let number = 1;
2 let numberFloating = 3.213;
3 let string = "Hello";
4 let boolean = true;
5 let nullValue = null;
6 let undefinedValue = undefined;
7
8
9 console.log(number);
10 console.log(numberFloating);
11 console.log(string);
12 console.log(boolean);
13 console.log(nullValue);
14 console.log(undefinedValue);
```