



JavaScript

Klára Pešková, Klara.Peskova@mff.cuni.cz
Katedra softwaru a výuky informatiky, MFF UK
Základy tvorby webu, ZS 2024/25

JavaScript

- programovací jazyk
- vznik v r. 1995
- není potřeba ho nijak instalovat, běží v prohlížeči

- HTML – obsah stránky
- CSS – vzhled a layout
- JavaScript – jak se stránka chová

Co JavaScript umí

- měnit obsah HTML tagů
- měnit atributy html tagů
- měnit styly

Co JavaScript umí

- měnit obsah HTML tagů
 - např. přepsat text
- měnit atributy html tagů
 - např. změnit obrázek
- měnit styly
 - např. schovat a zobrazit HTML element

Kdy?

- jako reakce na nějakou událost, např.
 - kliknutí na element
 - přejetí myší před element
 - stisk klávesy
 - načtení stránky

Kam se JavaScript píše

- do tagu `<script></script>`
- do externím souboru, koncovka `.js`
`<script src="muj_skript.js"></script>`
- uvnitř `<head>` nebo `<body>`
- `<script>` může být v HTML vícekrát
- umístění na konci `<body>` může zrychlit načítání stránky
- dříve: `<script type="text/javascript">`
 - typ není potřeba vyplňovat, JavaScript je defaultní skriptovací jazyk pro HTML

JavaScript v externím souboru

- oddělení HTML a JavaScriptu je přehlednější
- stejný .js soubor je možné použít na více stránkách
- nakešování souboru (pozor!)
- k jednomu HTML je možné připojit více .js souborů

Výstup JavaScriptu

- obsah HTML elementu, pomocí innerHTML

```
document.getElementById("demo").innerHTML =  
"Výsledek je " + z + ".";
```

- vyskakovací okno

```
alert("pozdrav")
```



- výstup do konzole

```
console.log("test")
```

- document.write() - zapíše výstup přímo do HTML; nepoužívat po načtení dokumentu, dokument by se tím přepsal

https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_doc_write4

Syntax

- c-čková
- příkazy by měly (ale nemusí) být oddělené středníky
- case sensitive
- dynamicky typovaný
- lowerCamelCase (zvyklost)

Proměnné

- deklarace proměnných
 - `var` – v rámci celého programu / funkce
 - re-deklarace nevynuluje hodnotu
 - `let` – v rámci bloku (od r. 2015)
 - `const` – konstanta v rámci bloku – používat (od r. 2015)
- název může začínat písmenem, podtržítkem nebo dolarem
 - *Using the dollar sign is not very common in JavaScript, but professional programmers often use it as an alias for the main function in a JavaScript library.*
 - *In the JavaScript library jQuery, for instance, the main function `$` is used to select HTML elements. In jQuery `$("p");` means "select all p elements".*

Znakové řetězce

- stringy v uvozovkách nebo apostrofech nebo v `
- skládání řetězců pomocí +
- string + int -> string (od prvního stringu dál)

```
var x = 2 + 3 + "5";    -> x = 55
```

- přístup k jednotlivým znakům přes index (v hranatých závorkách)
- ```
let text = `Welcome ${firstName}, ${lastName}!`;
```

# Komentáře

- `//` jednořádkové
- `/*` blokové `*/`

# Operátory

- && (and), || (or), ! (negace)
- +=, ++, --
- \*\* (mocnina), % (modulo)
- === stejná hodnota a stejný typ
- == stejná hodnota

```
let x = 5; x == "5" -> true
```

# Typy

- je možné zjistit pomocí statementu `typeof`  
`typeof true`
- číslo na string: `číslo.toString()`
- `typ number`
  - vždy 64-bit Floating Point,
  - integery jsou reprezentovány přesně do 15 číslic
  - převod ze stringu `Number("123")`
- `boolean (true, false)`
- `undefined`
- `NaN`

# Pole (array)

- `const auta = ["Tesla", "Ford", "Škoda"];`

- číslované od 0

`auta[0] -> "Tesla"`

- procházení pole `for` cyklem

`for (let i = 0; i < auta.length; i++) {}`

- přidání prvku na konec pole

`auta[auta.length] = "Renault";`

- pokud přidáváme mimo rozsah pole, můžeme si vyrobit "díry"

- `push`, `pop`

- nečíselné indexy -> typ se automaticky změní na `object`

• • •

# Objecty

- klíče jsou stringy
- `const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};`
- přístup k položkám:
  - `objectName.propertyName`
  - `objectName["propertyName"]`

- metody:

```
const person = {
 firstName: "John",
 lastName : "Doe",
 fullName : function() {
 return this.firstName + " " + this.lastName;
 }
};
document.getElementById("demo").innerHTML =
person.fullName();
```

- Typ `Map` - klíče jakéhokoliv typu



# Třídy

- od r. 2015
- třídy jako šablony pro objecty

```
// Constructor Function for Person objects
function Person(first, last, age, eye) {
 this.firstName = first;
 this.lastName = last;
 this.age = age;
 this.fullName = function() {
 return this.firstName + " " + this.lastName
 };
}

class Car {
 constructor(name, year) {
 this.name = name;
 this.year = year;
 }
}
```

# Funkce

```
function vynasob(c1, c2) {
 return c1 * c2;
}
vynasob(10, 3)
```

- příklady volání funkce
  - když uživatel klikne na tlačítko
  - jen tak z JavaScriptového kódu

- funkce v proměnné

```
hello = function () {
 return "Hello World!";
}
```

- defaultní hodnoty parametrů

```
function (a=1, b=1) { ... }
```

# static object Math

- matematické konstanty a funkce
  - `Math.PI`
  - `Math.round(x)`

# Syntax

- `if, else, else if`
- ```
switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}
```

Cykly - while, for

- while, do - while
- ```
for (let i = 0; i < cars.length; i++) {
 text += cars[i] + "
";
}
```

# for in, for of

- **for in** - iterace přes vlastnosti (klíče) objectu / indexy v poli

```
const person = {jm:"John", pr:"Doe", age:25};
```

```
let text = "";
for (let x in person) {
 text += person[x];
}
```

- **for of** - iterace přes hodnoty sekvence

```
const cars = ["BMW", "Volvo", "Mini"];
```

```
let text = "";
for (let x of cars) {
 text += x;
}
```

# JSON

- formát pro předávání a uchovávání dat (nejen pro JavaScript)
  - např. předávání mezi prohlížečem a serverem

- JavaScript Object Notation

```
"employees": [
 {"firstName": "John", "lastName": "Doe"},
 {"firstName": "Anna", "lastName": "Smith"},
 {"firstName": "Peter", "lastName": "Jones"}
]
```

- JSON je pouze text
- v JavaScriptu - snadná práce
  - `const obj = JSON.parse(text);`

# Debuggování

- nástroje pro vývojáře v prohlížeči – debugger
- klíčové slovo `debugger`; v kódu = breakpoint, pokud není zapnutý debugger, nedělá nic
- `console.log(x) ;`

*The first known computer bug was a real bug (an insect) stuck in the electronics.*



# HTML DOM

- DOM - Document Object Model, W3C standard
- definuje
  - HTML elementy jako objekty
  - vlastnosti HTML elementů
  - metody, jak se k HTML elementům dostat
  - události HTML elementů
- *The HTML DOM is a standard for how to get, change, add, or delete HTML elements.*

# Příklad

- `document.getElementById("demo").innerHTML = "Hello World!";`
- `document` je celá webová stránka

# Výběr HTML elementu

- `document.getElementById(id)`
- `document.getElementsByTagName(name)`
  - vrátí prvky (HTML collection)
- `document.getElementsByClassName(name)`
  - vrátí prvky (HTML collection)
- `document.querySelectorAll("p.intro");`
  - vrátí prvky (HTML collection)
- př. odstavec uvnitř main:

```
const x = document.getElementById("main");
const y = x.getElementsByTagName("p");
```

# Změna elementu

- `element.innerHTML` = nový obsah HTML elementu
- `element.attribute` = nová hodnota HTML atributu
- `element.style.property` = new style
  - `document.getElementById("p2").style.color = "blue";`

# Přidání / odebrání elementu

- `document.createElement(element)` - vytvoření HTML elementu
- `document.removeChild(element)` - odebrání HTML elementu
- `document.appendChild(element)`
- `document.replaceChild(new, old)`
  
- `document.write(text)` - zapsání (čehokoliv) přímo do HTML

# Události

- JavaScript může reagovat na události na HTML stránce, např. na
  - načtení stránky
  - kliknutí na tlačítko
  - změnu pole ve formuláři
- JavaScript může být přiřazen události pomocí speciálního atributu HTML elementu
  - `<element event="some JavaScript">`
  - `<button onclick="document.getElementById('demo').innerHTML = Date()">The time is?</button>`
  - `<button onclick="this.innerHTML = Date()">The time is?</button>`
  - `<button onclick="displayDate()">The time is?</button>`

# Validace formulářů

- bude později

# Události

- `onclick`
- `onchange`
- `onmouseover`
- `onmouseout`
- `onkeydown`
- `onload` - dokončení načtení stránky prohlížečem



# Události

```
<h1 onclick="this.innerHTML = 'kliknuto'">
klikni</h1>
```

- přidání události v JavaScriptu
  - `document.getElementById(id).onclick = function(){code}`
- Event Handler – může jich být přidáno libovolné množství
  - `document.getElementById("tlacitko").addEventListener("click", displayDate);`

# BOM – Browser Object Model

- interakce JavaScriptu s prohlížečem
  - `window`
    - `window.innerWidth`; `window.innerHeight`;
    - `window.open()` - otevření nového okna
    - `window.location.href` - URL současné stránky
    - `window.location.protocol`
    - `window.location.assign(url)`
      - načtení nového dokumentu
    - `window.history.back()`,  
`window.history.forward()`
    - `window.navigator.onLine` – je browser online?

# BOM - Browser Object Model

- `window.alert` [https://www.w3schools.com/js/js\\_popup.asp](https://www.w3schools.com/js/js_popup.asp)
- Časovač
  - `setTimeout(function, milliseconds)`
    - provede funkci po určeném čase
  - `clearTimeout()`
  - `setInterval(function, milliseconds)`
    - opakovaně provádí funkci po určeném časovém intervalu
  - `clearInterval()`
- Cookies
  - [https://www.w3schools.com/js/js\\_cookies.asp](https://www.w3schools.com/js/js_cookies.asp)

# JavaScript a grafika

- canvas
- knihovny na kreslení grafů
- [https://www.w3schools.com/js/js\\_graphics.asp](https://www.w3schools.com/js/js_graphics.asp)