

# Webové technologie (Frontend)

Frontend webových aplikací tvoří tři základní technologie, které společně vytváří to, co uživatel vidí a s čím interaguje v prohlížeči.

## HTML (HyperText Markup Language)

HTML je **strukturální značkovací jazyk** pro tvorbu webových stránek. Definiuje obsah a jeho hierarchii pomocí značek (tagů).

### Co je HTML

HTML není programovací jazyk, ale **markup language** - jazyk pro značkování obsahu. Každý element má svůj sémantický význam a říká prohlížeči, jak má být obsah interpretován.

#### Základní charakteristika:

- Popisuje strukturu dokumentu (nadpisy, odstavce, obrázky, odkazy)
- Používá značky uzavřené do lomených závorek, například `<h1>`, `<p>`, `<img>`
- Je to kostra webové stránky
- Aktuální verze je HTML5 (od roku 2014)

#### Struktura HTML dokumentu

```
<!DOCTYPE html>           <!-- Deklarace typu dokumentu -->
<html lang="cs">          <!-- Kořenový element, jazyk stránky -->
  <head>                  <!-- Hlavička - metadata -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Moje stránka</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>                 <!-- Tělo - viditelný obsah -->
    <h1>Vítejte</h1>
    <p>Toto je odstavec textu.</p>
  </body>
</html>
```

#### Nejdůležitější HTML elementy

##### Strukturální elementy:

- `<header>` - hlavička stránky nebo sekce
- `<nav>` - navigační menu

- `<main>` - hlavní obsah stránky
- `<section>` - tematická sekce
- `<article>` - samostatný článek
- `<aside>` - boční obsah
- `<footer>` - patička stránky

### Textové elementy:

- `<h1>` až `<h6>` - nadpisy (h1 je největší)
- `<p>` - odstavec
- `<strong>` - důležitý text (tučně)
- `<em>` - zdůrazněný text (kurzíva)
- `<br>` - zalomení řádku
- `<hr>` - horizontální čára

### Odkazy a multimédia:

- `<a href=„url“>text</a>` - odkaz
- `<img src=„obr.jpg“ alt=„popis“>` - obrázek
- `<video>` - video
- `<audio>` - audio

### Seznamy:

- `<ul>` - nečíslovaný seznam (odrážky)
- `<ol>` - číslovaný seznam
- `<li>` - položka seznamu

### Formuláře:

- `<form>` - formulář
- `<input>` - vstupní pole
- `<textarea>` - textová oblast
- `<button>` - tlačítko
- `<select>` - rozbalovací seznam

### Příklad komplexnější struktury:

```
<article>
  <header>
    <h2>Název článku</h2>
    <p>Autor: Jan Novák | Datum: 6.1.2026</p>
  </header>

  <section>
    <h3>Úvod</h3>
    <p>Text úvodu s <strong>důležitou informací</strong>.</p>
    
  </section>

  <footer>
    <a href="#">Číst více...</a>
```

```
</footer>
</article>
```

## HTML atributy

Elementy mohou mít **atributy**, které poskytují dodatečné informace:

- `id` - jedinečný identifikátor elementu
- `class` - třída pro stylování (může být více prvků)
- `style` - inline styly (nedoporučuje se)
- `title` - tooltip při najetí myši
- `data-*` - vlastní datové atributy

```
<div id="hlavni-obsah" class="kontejner zvyrazneny" data-
kategorie="novinky">
  Obsah divu
</div>
```

## CSS (Cascading Style Sheets)

CSS je **stylový jazyk** pro vizuální úpravu webových stránek. Určuje, jak mají HTML elementy vypadat a jak jsou rozmístěny na stránce.

### Co je CSS

CSS odděluje **prezentaci** od **struktury**. Zatímco HTML definuje, co je obsah, CSS určuje, jak má obsah vypadat.

#### Základní charakteristika:

- Definuje barvy, fonty, rozložení, rozměry prvků
- Odděluje vzhled od struktury (HTML)
- Umožňuje responzivní design pro různá zařízení
- Kaskádové - styly se dědí a mohou se přepisovat
- Aktuální verze je CSS3

### Způsoby připojení CSS

#### 1. Externí CSS soubor (doporučeno):

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

#### 2. Interní CSS (v hlavičce HTML):

```
<style>
  h1 { color: blue; }
```

```
</style>
```

### 3. Inline CSS (přímo v elementu):

```
<h1 style="color: blue;">Nadpis</h1>
```

## Anatomie CSS pravidla

```
selektor {  
  vlastnost: hodnota;  
  další-vlastnost: hodnota;  
}
```

### Příklad:

```
h1 {  
  color: blue;           /* Barva textu */  
  font-size: 24px;      /* Velikost písma */  
  text-align: center;   /* Zarovnání textu */  
  margin-bottom: 20px;  /* Vnější okraj zespodu */  
}
```

## Typy selektorů

### Elementový selektor:

```
p { color: black; } /* Všechny <p> elementy */
```

### Třídní selektor (class):

```
.zvyrazneny { background: yellow; } /* class="zvyrazneny" */
```

### ID selektor:

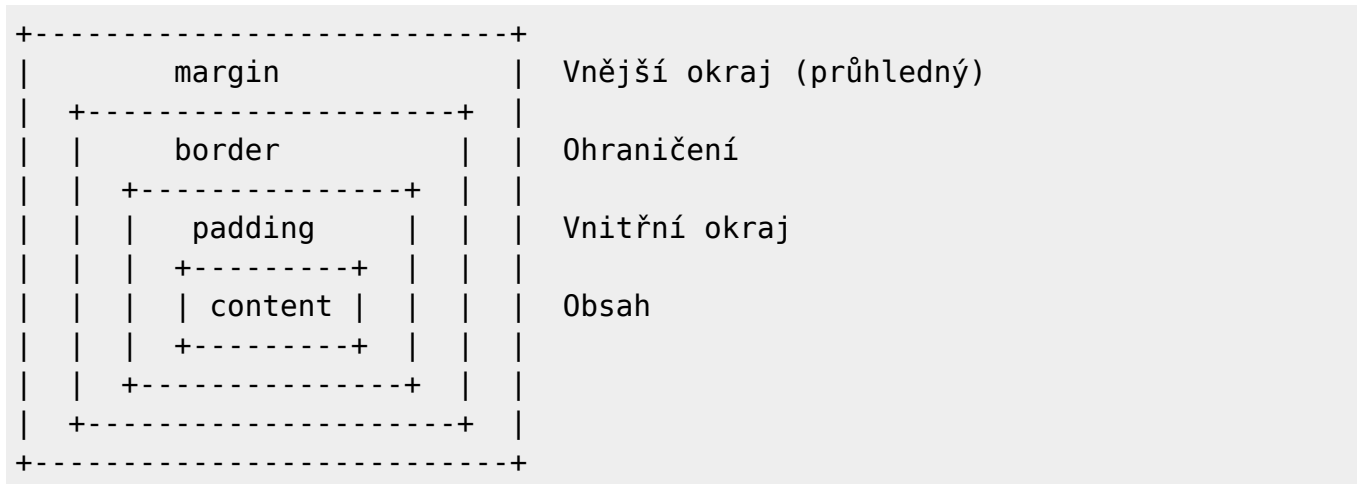
```
#hlavicka { font-size: 32px; } /* id="hlavicka" */
```

### Kombinované selektory:

```
div p { color: red; } /* <p> uvnitř <div> */  
.box h2 { margin: 0; } /* <h2> uvnitř elementu s class="box" */  
a:hover { color: green; } /* Odkaz při najetí myší */  
input:focus { border: 2px solid blue; } /* Vstup při zaměření */
```

## Box Model

Každý HTML element je obdélník s následující strukturou:



```
.box {
  width: 300px;           /* Šířka obsahu */
  height: 200px;         /* Výška obsahu */
  padding: 20px;         /* Vnitřní okraj */
  border: 2px solid black; /* Ohraničení */
  margin: 10px;          /* Vnější okraj */
}
```

## Důležité CSS vlastnosti

### Barvy:

```
color: red;               /* Pojmenovaná barva */
color: #ff0000;           /* Hexadecimální */
color: rgb(255, 0, 0);    /* RGB */
color: rgba(255, 0, 0, 0.5); /* RGB s průhledností */
background-color: #f0f0f0; /* Barva pozadí */
```

### Texty:

```
font-family: Arial, sans-serif; /* Rodina písma */
font-size: 16px;                 /* Velikost */
font-weight: bold;               /* Tloušťka */
text-align: center;              /* Zarovnání */
text-decoration: underline;      /* Podtržení */
line-height: 1.6;                /* Výška řádku */
```

### Rozložení (Layout):

```
display: block;                /* Blokovaný element */
display: inline;               /* Řádkový element */
display: flex;                 /* Flexbox layout */
display: grid;                 /* Grid layout */
position: relative;           /* Relativní pozicování */
position: absolute;           /* Absolutní pozicování */
float: left;                   /* Obtékání */
```

## Flexbox (moderní layout):

```
.kontejner {
  display: flex;
  justify-content: center;      /* Vodorovné zarovnání */
  align-items: center;         /* Svislé zarovnání */
  gap: 20px;                   /* Mezery mezi prvky */
}
```

## Responzivní design

**Media queries** umožňují různé styly pro různé velikosti obrazovek:

```
/* Základní styly pro mobily */
.kontejner {
  width: 100%;
  padding: 10px;
}

/* Styly pro tablety a větší */
@media (min-width: 768px) {
  .kontejner {
    width: 750px;
    padding: 20px;
  }
}

/* Styly pro desktop */
@media (min-width: 1200px) {
  .kontejner {
    width: 1140px;
  }
}
```

## Příklad kompletnějšího CSS

```
/* Reset základních stylů */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: 'Segoe UI', Tahoma, sans-serif;
  line-height: 1.6;
  color: #333;
  background-color: #f4f4f4;
}
```

```
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}  
  
.card {  
  background: white;  
  border-radius: 8px;  
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
  padding: 20px;  
  margin-bottom: 20px;  
  transition: transform 0.3s ease;  
}  
  
.card:hover {  
  transform: translateY(-5px);  
  box-shadow: 0 4px 8px rgba(0,0,0,0.2);  
}  
  
.btn {  
  display: inline-block;  
  padding: 10px 20px;  
  background-color: #007bff;  
  color: white;  
  text-decoration: none;  
  border-radius: 5px;  
  transition: background-color 0.3s;  
}  
  
.btn:hover {  
  background-color: #0056b3;  
}
```

## JavaScript

JavaScript je **interpretovaný programovací jazyk**, který umožňuje interaktivitu a dynamické chování webových stránek. Je to jediný programovací jazyk, který běží nativně v prohlížeči.

### Co je JavaScript

JavaScript (zkráceně JS) je plnohodnotný programovací jazyk s širokou škálou využití od jednoduchých interakcí až po komplexní webové aplikace.

#### Základní charakteristika:

- Reaguje na události (kliknutí, psaní, pohyb myši)
- Mění obsah stránky bez nutnosti jejího znovunačtení
- Komunikuje se servery (AJAX, Fetch API)
- Validuje formuláře, vytváří animace
- Je objektivě orientovaný a funkcionální
- Běží v prohlížeči (frontend) i na serveru (Node.js - backend)

## Způsoby připojení JavaScriptu

### 1. Externí JS soubor (doporučeno):

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

### 2. Interní JavaScript:

```
<script>
  console.log('Ahoj světe!');
</script>
```

### 3. Inline JavaScript (nedoporučuje se):

```
<button onclick="alert('Kliknuto!')">Klikni</button>
```

## Základní syntaxe

### Proměnné:

```
// Moderní způsob deklarace proměnných
let jmeno = 'Jan';           // Proměnná (lze měnit)
const vek = 25;             // Konstanta (nelze měnit)
var stary = 'zastaralé';    // Starý způsob (nepoužívat)

// Datové typy
let cislo = 42;              // Number
let text = "Ahoj";          // String
let pravda = true;          // Boolean
let pole = [1, 2, 3];        // Array (pole)
let objekt = {jmeno: 'Jan'}; // Object (objekt)
let nic = null;              // Null
let nedefinovano = undefined; // Undefined
```

### Operátory:

```
// Aritmetické
let soucet = 5 + 3;          // 8
let rozdil = 5 - 3;          // 2
let soucin = 5 * 3;          // 15
let podil = 6 / 3;           // 2
```

```
let zbytek = 7 % 3;           // 1 (modulo)

// Porovnávací
5 == '5';                   // true (porovnává hodnotu)
5 === '5';                  // false (porovnává hodnotu i typ)
5 != 3;                     // true
5 > 3;                      // true
5 <= 5;                     // true

// Logické
true && false;              // false (AND)
true || false;             // true (OR)
!true;                     // false (NOT)
```

## Podmínky a cykly

### Podmínky:

```
let vek = 20;

if (vek >= 18) {
  console.log('Plnoletý');
} else if (vek >= 15) {
  console.log('Teenager');
} else {
  console.log('Dítě');
}

// Ternární operátor
let status = vek >= 18 ? 'dospělý' : 'dítě';

// Switch
switch (den) {
  case 'pondělí':
    console.log('Začátek týdne');
    break;
  case 'pátek':
    console.log('Konec týdne');
    break;
  default:
    console.log('Běžný den');
}
```

### Cykly:

```
// For cyklus
for (let i = 0; i < 5; i++) {
  console.log(i); // 0, 1, 2, 3, 4
}
```

```
// While cyklus
let i = 0;
while (i < 5) {
  console.log(i);
  i++;
}

// For...of pro pole
let ovoce = ['jablko', 'hruška', 'banán'];
for (let kus of ovoce) {
  console.log(kus);
}

// ForEach
ovoce.forEach(function(kus) {
  console.log(kus);
});
```

## Funkce

```
// Klasická funkce
function pozdrav(jmeno) {
  return 'Ahoj ' + jmeno + '!';
}

// Arrow funkce (moderní ES6)
const secti = (a, b) => {
  return a + b;
};

// Zkrácená arrow funkce
const nasobVDva = x => x * 2;

// Volání funkce
console.log(pozdrav('Petr')); // "Ahoj Petr!"
console.log(secti(5, 3)); // 8
console.log(nasobVDva(4)); // 8
```

## Pole (Arrays)

```
let cisla = [1, 2, 3, 4, 5];

// Přidání prvku
cisla.push(6); // [1, 2, 3, 4, 5, 6]
cisla.unshift(0); // [0, 1, 2, 3, 4, 5, 6]

// Odebrání prvku
```

```
cisla.pop();           // Odebere poslední
cisla.shift();        // Odebere první

// Užitečné metody
cisla.length;         // Počet prvků
cisla.indexOf(3);     // Index prvku
cisla.includes(3);   // true/false
cisla.slice(1, 3);   // Část pole [2, 3]

// Pokročilé metody
let dvojnásobek = cisla.map(x => x * 2); // [2, 4, 6, 8, 10]
let sude = cisla.filter(x => x % 2 === 0); // [2, 4]
let soucet = cisla.reduce((a, b) => a + b, 0); // 15
```

## Objekty

```
// Vytvoření objektu
let osoba = {
  jmeno: 'Jan',
  prijmeni: 'Novák',
  vek: 30,
  email: 'jan@email.cz',
  pozdrav: function() {
    return 'Ahoj, jsem ' + this.jmeno;
  }
};

// Přístup k vlastnostem
console.log(osoba.jmeno); // "Jan"
console.log(osoba['email']); // "jan@email.cz"
console.log(osoba.pozdrav()); // "Ahoj, jsem Jan"

// Přidání/změna vlastnosti
osoba.telefon = '123456789';
osoba.vek = 31;

// Odstranění vlastnosti
delete osoba.email;
```

## DOM Manipulace

DOM (Document Object Model) je rozhraní pro práci s HTML dokumentem.

### Výběr elementů:

```
// Výběr jednoho elementu
let nadpis = document.getElementById('nadpis');
let prvni0dstavec = document.querySelector('p');
```

```
let tlacitko = document.querySelector('.btn-primary');

// Výběr více elementů
let vsechnyOdstavce = document.querySelectorAll('p');
let divySTridou = document.getElementsByClassName('box');
```

## Změna obsahu:

```
// Změna textu
nadpis.textContent = 'Nový text';
nadpis.innerHTML = '<strong>Tučný text</strong>';

// Změna atributů
let obrazek = document.querySelector('img');
obrazek.src = 'novy-obrazek.jpg';
obrazek.alt = 'Popisek';

// Změna stylů
nadpis.style.color = 'blue';
nadpis.style.fontSize = '24px';
nadpis.style.display = 'none'; // Skrytí

// Práce s třídami
nadpis.classList.add('zvyrazneny');
nadpis.classList.remove('skryty');
nadpis.classList.toggle('aktivni');
```

## Vytváření elementů:

```
// Vytvoření nového elementu
let novyDiv = document.createElement('div');
novyDiv.textContent = 'Nový obsah';
novyDiv.classList.add('box');

// Přidání do stránky
document.body.appendChild(novyDiv);

// Odebrání elementu
novyDiv.remove();
```

## Události (Events)

```
// Kliknutí
let tlacitko = document.querySelector('button');
tlacitko.addEventListener('click', function() {
  alert('Tlačítko bylo kliknuto!');
});

// Arrow funkce
```

```
tlacitko.addEventListener('click', () => {
  console.log('Kliknutí');
});

// Různé typy událostí
document.addEventListener('DOMContentLoaded', () => {
  console.log('Stránka načtena');
});

let input = document.querySelector('input');
input.addEventListener('input', (e) => {
  console.log('Napsáno:', e.target.value);
});

input.addEventListener('focus', () => {
  console.log('Input zaměřen');
});

input.addEventListener('blur', () => {
  console.log('Input opuštěn');
});

// Pohyb myši
document.addEventListener('mousemove', (e) => {
  console.log('X:', e.clientX, 'Y:', e.clientY);
});
```

## Asynchronní JavaScript

### Časovače:

```
// Spuštění po 2 sekundách
setTimeout(() => {
  console.log('Uplynuly 2 sekundy');
}, 2000);

// Opakování každou sekundu
let interval = setInterval(() => {
  console.log('Tikání');
}, 1000);

// Zastavení intervalu
clearInterval(interval);
```

### Fetch API (komunikace se serverem):

```
// Získání dat ze serveru
fetch('https://api.example.com/data')
  .then(response => response.json())
```

```
.then(data => {
  console.log(data);
})
.catch(error => {
  console.error('Chyba:', error);
});

// Moderní async/await syntaxe
async function nactiData() {
  try {
    let response = await fetch('https://api.example.com/data');
    let data = await response.json();
    console.log(data);
  } catch (error) {
    console.error('Chyba:', error);
  }
}
```

## Praktický příklad

```
// Interaktivní todo list
document.addEventListener('DOMContentLoaded', () => {
  const form = document.querySelector('#todo-form');
  const input = document.querySelector('#todo-input');
  const list = document.querySelector('#todo-list');

  // Přidání nového úkolu
  form.addEventListener('submit', (e) => {
    e.preventDefault(); // Zabránění odeslání formuláře

    const text = input.value.trim();
    if (text === '') return;

    // Vytvoření nového prvku
    const li = document.createElement('li');
    li.textContent = text;

    // Přidání tlačítka pro smazání
    const btn = document.createElement('button');
    btn.textContent = 'Smazat';
    btn.addEventListener('click', () => {
      li.remove();
    });

    li.appendChild(btn);
    list.appendChild(li);

    // Vyčištění inputu
    input.value = '';
  });
});
```

```
});
```

## Vzájemná spolupráce

Tyto tři technologie fungují jako **stavební bloky moderního webu**:

- **HTML** = struktura (co je na stránce)
- **CSS** = vzhled (jak to vypadá)
- **JavaScript** = chování (co to dělá)

**Analogie:** Představte si dům - HTML je konstrukce a zdivo, CSS je malování a dekorace interiéru, JavaScript jsou elektrické rozvody a funkční systémy, které dům oživují.

From:

<https://serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:

<https://serviceit.cz/doku.php?id=it:sw:web>

Last update: **2026/01/06 17:29**

