Pojem Log

Log (česky záznam, protokol) je **textový soubor nebo výstup**, který zaznamenává události, chyby, varování a další informace o chodu aplikace, systému nebo programu.

Co je log

Log slouží k:

- **Sledování chování** aplikace v reálném čase
- **Debugování** - hledání a oprava chyb
- **Auditování** - kdo, kdy a co provedl
- **Monitorování** výkonu a stavu systému
- **Analýze** problémů, které se staly v minulosti

Kde se s logy setkáme

Webový vývoj

1. Konzole prohlížeče (Browser Console)

V JavaScriptu se logy vypisují pomocí `console` objektu:

```
console.log('Základní zpráva');           // Běžná informace
console.info('Informativní zpráva');      // Info
console.warn('Varování');                 // Varování (žlutě)
console.error('Chyba!');                  // Chyba (červeně)
console.debug('Debug info');              // Debug informace

// Logování proměnných
let jmeno = 'Jan';
let vek = 25;
console.log('Uživatel:', jmeno, 'Věk:', vek);

// Logování objektů
let osoba = {jmeno: 'Jan', vek: 25};
console.log(osoba);                       // Vypíše celý objekt
console.table(osoba);                     // Vypíše jako tabulku
```

Jak otevřít konzoli prohlížeče:

- **Chrome/Edge:** F12 nebo Ctrl+Shift+I → záložka Console
- **Firefox:** F12 nebo Ctrl+Shift+K
- **Safari:** Cmd+Option+C (macOS)

Serverový vývoj

Node.js (JavaScript na serveru):

```
console.log('Server běží na portu 3000');
console.error('Nepodařilo se připojit k databázi');
```

PHP:

```
error_log("Chyba při zpracování požadavku");
echo "Ladící výpis: " . $promenna;
```

Python:

```
import logging

logging.info("Aplikace spuštěna")
logging.warning("Nedostatečná paměť")
logging.error("Kritická chyba!")
```

Systémové logy

Operační systémy zapisují logy do speciálních souborů:

Linux/Unix:

- `/var/log/syslog` - hlavní systémový log
- `/var/log/apache2/error.log` - chyby webového serveru
- `/var/log/auth.log` - přihlášení uživatelů

Windows:

- Event Viewer (Prohlížeč událostí)
- Aplikační logy, systémové logy, bezpečnostní logy

Úrovně logování

Logy mají různé **úrovně závažnosti** (od nejméně po nejvíce závažné):

Úroveň	Popis	Použití
TRACE	Velmi detailní info	Hloubkové ladění
DEBUG	Ladící informace	Vývoj a testování
INFO	Běžné informace	Důležité události
WARN	Varování	Potenciální problémy
ERROR	Chyby	Chyby, které aplikace zvládne
FATAL	Kritické chyby	Pád aplikace

Příklad logu

Typický formát logové zprávy:

```
[2026-01-06 14:23:15] [INFO] Server spuštěn na portu 3000
[2026-01-06 14:23:47] [DEBUG] Načítám data z databáze...
[2026-01-06 14:23:48] [INFO] Uživatel 'jan.novak' se přihlásil
[2026-01-06 14:24:12] [WARN] Pomalý dotaz do databáze (2.3s)
[2026-01-06 14:25:03] [ERROR] Nepodařilo se odeslat email: Connection
timeout
[2026-01-06 14:25:05] [INFO] Pokus o opětovné odeslání emailu...
```

Každý řádek obvykle obsahuje:

- **Časové razítko** - kdy se událost stala
- **Úroveň** - jak závažná je zpráva
- **Zpráva** - popis události
- Volitelně: **zdroj** (modul, funkce), **uživatel**, **IP adresa**

Praktický příklad v JavaScriptu

```
// Funkce pro přihlášení uživatele
function prihlasit(email, heslo) {
  console.log(`[${new Date().toISOString()}] Pokus o přihlášení: ${email}`);

  if (!email || !heslo) {
    console.error('Chyba: Email nebo heslo je prázdné');
    return false;
  }

  try {
    // Simulace ověření
    if (heslo.length < 6) {
      console.warn('Varování: Příliš krátké heslo');
    }

    // Úspěšné přihlášení
    console.log(`✓ Uživatel ${email} úspěšně přihlášen`);
    console.info('Session ID:', Math.random().toString(36));
    return true;

  } catch (error) {
    console.error('Kritická chyba při přihlášení:', error);
    return false;
  }
}

// Použití
```

```
prihlasit('jan@email.cz', 'heslo123');
```

Výstup v konzoli:

```
[2026-01-06T13:23:15.123Z] Pokus o přihlášení: jan@email.cz
Varování: Příliš krátké heslo
✓ Uživatel jan@email.cz úspěšně přihlášen
Session ID: 8k2j9f3h1s
```

Doporučené praktiky

- **Během vývoje:** Používejte hodně logů pro pochopení toku programu
- **V produkci:** Logujte jen důležité události, chyby a varování
- **Nelogujte citlivá data:** hesla, čísla kreditních karet, osobní údaje
- **Používejte správné úrovně:** ERROR pro chyby, INFO pro běžné události
- **Strukturujte zprávy:** Konzistentní formát usnadňuje čtení a zpracování
- **Rotujte logy:** Staré logy archivujte nebo mažte (šetří místo na disku)

Nástroje pro práci s logy

- **Prohlížeč:** DevTools Console
- **Textové editory:** Pro čtení log souborů
- **grep (Linux):** Vyhledávání v logech: `grep ERROR app.log`
- **Logování knihovny:** Winston (Node.js), Log4j (Java), Monolog (PHP)
- **Monitoring služby:** Sentry, LogRocket, Datadog - sledují logy v reálném čase
- **ELK Stack:** Elasticsearch + Logstash + Kibana - pro pokročilou analýzu

Shrnutí

Log je **záznam toho, co se děje v aplikaci**. Je to jako deník programu, který vývojářům pomáhá:

- Pochopit, jak program funguje
- Najít a opravit chyby
- Sledovat, co uživatelé dělají
- Analyzovat problémy, které se staly v minulosti

Každý programátor by měl umět logy používat - jsou to oči a uši do běžící aplikace.