

CSS előfeldolgozók

Jeszenszky Péter

Debreceni Egyetem, Informatikai Kar

2024. november 12.

Felhasználási feltételek

This work is licensed under a [Creative Commons](#)
“[Attribution 4.0 International](#)” license.



Mi a CSS előfeldolgozó?

- A CSS előfeldolgozók olyan programok, melyek CSS kódot generálnak a CSS írására szolgáló saját szintaxisukból.
- A CSS-t új lehetőségekkel egészítik ki, mint például változók, egymásba ágyazás, függvények, mixinek.
- Lásd: [MDN Web Docs Glossary – CSS preprocessor](#)

Létező CSS előfeldolgozók

- Less: <https://lesscss.org/>
- PostCSS: <https://postcss.org/>
- Sass: <https://sass-lang.com/>
- Stylus: <https://stylus-lang.com/>

Példák

<https://github.com/jeszy75/css-preprocessor-examples>

Node.js (1)

- A V8 JavaScript motorra épülő JavaScript futtató környezet, melyet skálázható hálózati alkalmazások létrehozásához terveztek.
- Webhely: <https://nodejs.org/>
- Tároló: <https://github.com/nodejs/node>
- Licenc: MIT License
- Programozási nyelv: C++, JavaScript
- Platform: Linux, macOS, Windows

Node.js (2)

- Lehetővé teszi a fejlesztők számára böngészőn kívül futó JavaScript alkalmazások létrehozását.
- Használható kliens oldali és szerver oldali alkalmazások fejlesztéséhez is.
- Csomag ökoszisztémája, az npm, a világ legnagyobb nyílt forrású könyvtár ökoszisztémája.

Node.js (3)

“Helló, világ!” példa:

```
const { createServer } = require('node:http');

const hostname = '127.0.0.1';
const port = 3000;

const server = createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, World!');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Forrás: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>

Node.js telepítés (1)

- A tárgyalt implementációk többségéhez Node.js szükséges.
- Telepítők és binárisok innen tölthetők le:
 - <https://nodejs.org/en/download/prebuilt-installer>
 - <https://nodejs.org/en/download/prebuilt-binaries>
- A Node.js verziókezelők rugalmasabb módon teszik lehetővé a telepítést.

Node.js telepítés (2)

Node.js verziókezelők:

- Parancssori eszközök a Node.js rendszergazdai jogosultságok nélkül történő telepítéséhez, mely lehetővé teszi több különböző Node.js verzió egyidejű kezelését.

Node.js telepítés (3)

Node.js verziókezelők:

- asdf (programozási nyelv: shell; platform: Linux, macOS; licenc: MIT License) <https://asdf-vm.com/> <https://github.com/asdf-vm/asdf>
- Fast Node Manager (fnm) (programozási nyelv: Rust; platform: Linux, macOS, Windows; licenc: GPLv3) <https://github.com/Schniz/fnm>
- Node Version Manager (nvm) (programozási nyelv: shell; platform: Linux, macOS; licenc: MIT License) <https://github.com/nvm-sh/nvm>
- Volta (programozási nyelv: Rust; platform: Linux, macOS, Windows; licenc: Simplified BSD License) <https://volta.sh/>
<https://github.com/volta-cli/volta>

Node.js telepítés (4)

Az fnm telepítése és használata:

```
$ curl -fsSL https://fnm.vercel.app/install | bash
$ fnm list-remote # Az összes távoli verzió listázása
$ fnm list # Az összes lokálisan telepített verzió listázása
$ fnm install 23
$ fnm current
$ node --version
$ fnm install 16
$ fnm use 16
$ fnm current
$ node --version
```

CSS előfeldolgozók közös jellemzői (1)

Egysoros megjegyzések:

- Az egysoros megjegyzések `//` kezdetűek és a sor végéig tartanak.
- A fordítás során nem generálnak CSS-t.
- Lásd:
 - Less: <https://lesscss.org/#comments>
 - Sass: <https://sass-lang.com/documentation/syntax/comments>
 - Stylus: <https://stylus-lang.com/docs/comments.html>

CSS előfeldolgozók közös jellemzői (2)

@import:

- A CSS előfeldolgozók a böngészőktől eltérően kezelik az @import at-szabályokat.
- A sima CSS importokhoz a böngészőnek az oldal renderelésekor kell HTTP kéréseket végrehajtania.
- A CSS előfeldolgozók az importokat a fordítás során kezelik.
- Lásd:
 - Less: <https://lesscss.org/features/#import-atrules-feature>
 - Sass: <https://sass-lang.com/documentation/at-rules/import>
 - Stylus: <https://stylus-lang.com/docs/import.html>

Stílus szabályok egymásba ágyazása (1)

- Az egymásba ágyazással csoportosítani lehet a kapcsolódó stílus szabályokat, melyek kiválasztói ugyanazt a közös komponenst tartalmazzák.
- Az ágyásba ágyazás a stílus szabályokat szervezi, így az olvashatóságot és karbantarthatóságot javítja.
- Lásd:
 - Less: <https://lesscss.org/#nesting>
 - Sass: <https://sass-lang.com/documentation/style-rules/#nesting>
 - Stylus: <https://stylus-lang.com/docs/selectors.html>

Stílus szabályok egymásba ágyazása (2)

Példa:

```
/* Egymásba ágyazás nélkül: */  
article {  
  background-color: aliceblue;  
}  
  
article a {  
  color: green;  
}  
  
article > h2 {  
  text-decoration: underline;  
}  
  
article + p {  
  font-size: smaller;  
}
```

```
/* Egymásba ágyazással: */  
article {  
  background-color: aliceblue;  
  a {  
    color: green;  
  }  
  > h2 {  
    text-decoration: underline;  
  }  
  + p {  
    font-size: smaller;  
  }  
}
```

Stílus szabályok egymásba ágyazása (3)

Példa:

```
/* Egymásba ágyazás nélkül: */  
table.board {  
  border-collapse: collapse;  
}
```

```
table.board td {  
  border: thin solid black;  
}
```

```
table.board td, table.board th {  
  height: 3em;  
  text-align: center;  
  vertical-align: middle;  
  width: 3em;  
}
```

```
/* Egymásba ágyazással: */  
table.board {  
  border-collapse: collapse;  
  td {  
    border: thin solid black;  
  }  
  td, th {  
    height: 3em;  
    text-align: center;  
    vertical-align: middle;  
    width: 3em;  
  }  
}
```

Stílus szabályok egymásba ágyazása (4)

Példa: szülő kiválasztó (&)

```
/* Egymásba ágyazás nélkül: */
p {
  text-indent: 1em;
}
h2 + p {
  text-indent: 0;
}
h2 + p::first-letter {
  font-size: 1.5em;
}
p:hover {
  background-color: gold;
}
```

```
/* Egymásba ágyazással: */
p {
  text-indent: 1em;
  h2 + & {
    text-indent: 0;
    &::first-letter {
      font-size: 1.5em;
    }
  }
  &:hover {
    background-color: gold;
  }
}
```

CSS Nesting Module (1)

- Az alábbi CSS specifikáció bevezeti a stílus szabályok egymásba ágyazását, mely egy olyan lehetőség, melyet széles körben támogatnak a CSS előfeldolgozók:
 - [CSS Nesting Module \(W3C Working Draft, 14 February 2023\)](#)
- Böngésző támogatás: <https://caniuse.com/css-nesting>

CSS Nesting Module (2)

- A specifikáció jelenleg nem engedi meg, hogy a beágyazott szabályok kiválasztója típus kiválasztóval vagy funkcionális jelöléssel kezdődjön.
- Azonban 2024 novemberében a Firefox és a Google Chrome is támogatja a példában szereplő érvénytelen formát.

CSS Nesting Module (3)

Példa:

```
/* Érvénytelen beágyazás: */  
table.board {  
  td, th {  
    height: 3em;  
    width: 3em;  
  }  
}
```

```
/* Érvényes beágyazások: */  
table.board {  
  & td, & th {  
    height: 3em;  
    width: 3em;  
  }  
  :is(td, th) {  
    height: 3em;  
    width: 3em;  
  }  
}
```

Mixinek (1)

- A mixinek lehetővé teszik CSS deklarációk újrafelhasználás céljából történő csoportosítását.
- Akár argumentumokat is kaphatnak.
- Lásd:
 - Less: <https://lesscss.org/features/#mixins-feature>
 - Sass: <https://sass-lang.com/documentation/at-rules/mixin/>
 - Stylus: <https://stylus-lang.com/docs/mixins.html>

Mixinek (2)

Példa:

```
/* Less: */
.rounded-border() {
  border: thin solid black;
  border-radius: 0.25em;
}

nav {
  .rounded-border();
  background-color: azure;
}

p {
  .rounded-border();
  background-color: linen;
}

/* Sass (SCSS): */
@mixin rounded-border {
  border: thin solid black;
  border-radius: 0.25em;
}

nav {
  @include rounded-border;
  background-color: azure;
}

p {
  @include rounded-border;
  background-color: linen;
}

/* Stylus: */
rounded-border()
border thin solid black;
border-radius 0.25em;

nav
  rounded-border()
  background-color azure

p
  rounded-border()
  background-color linen
```

Mixinek (3)

Példa:

```
/* Less: */  
.set-size(@size) {  
  height: @size;  
  width: @size;  
}
```

```
table.chessboard td {  
  .set-size(2em);  
}
```

```
/* Sass (SCSS): */  
@mixin set-size($size) {  
  height: $size;  
  width: $size;  
}
```

```
table.chessboard td {  
  @include set-size(2em);  
}
```

```
/* Stylus: */  
set-size(size)  
  height size  
  width size
```

```
table.chessboard td  
  set-size 2em
```

Less

- A Less szintaxisa a CSS szintaxis egy kiterjesztése.
- Webhely: <https://lesscss.org/>
- Tároló: <https://github.com/less/less.js>
- Implementáció:
 - Less.js (programozási nyelv: JavaScript; licenc: Apache License 2.0)
<https://github.com/less/less.js>
- Állomány kiterjesztés: `.less`

Less: lehetőségek

- Változók
- Aritmetikai műveletek
- Beépített függvények (például logikai, matematikai, sztring- és színkezelő függvények)
- Asszociatív tömbök
- CSS őrfeltételek
- ...

Less: szintaxis

Példa:

```
@size: 2em;
```

```
table.chessboard {  
  td {  
    height: @size;  
    width: @size;  
  }  
}
```

Less: telepítés

A telepítéshez Node.js és az npm csomagkezelő szükséges:

```
npm install -g less
```

Less: eszköztámogatás

Lásd: <https://lesscss.org/tools/>

Less: szerkesztők

Szabad és nyílt forrású szoftverek:

- Visual Studio Code (platform: Linux, macOS, Windows; licenc: MIT License) <https://code.visualstudio.com/>
<https://github.com/Microsoft/vscode>
 - Kiterjesztések:
 - Easy LESS <https://marketplace.visualstudio.com/items?itemName=mrcrowl.easy-less>
<https://github.com/mrcrowl/vscode-easy-less>

Lásd: <https://code.visualstudio.com/docs/languages/css>

Less: online eszközök

- [less-preview](#)
- [CodePen](#)

Less: használat (1)

Parancssori használat:

```
lessc input.less output.css
```

Less: használat (2)

Less stíluslap társítása HTML dokumentumhoz:

```
<link rel="stylesheet/less" type="text/css"  
  href="style.less"/>
```

```
<script src="//cdn.jsdelivr.net/npm/less"/>
```

Sass

- Webhely: <https://sass-lang.com/>
- Tároló: <https://github.com/sass/sass>
- Állomány kiterjesztés: `.sass/.scss`

Sass: lehetőségek

Lásd: <https://sass-lang.com/guide>

- Változók
- Aritmetikai műveletek
- Beépített függvények (például logikai, matematikai, sztring- és színkezelő függvények)
- Asszociatív tömbök
- Vezérlési szerkezetek (@if, @each, @for, @while)
- Kiterjesztés/öröklés
- Modulok (@use)
- ...

Sass: szintaxis (1)

A Sass az alábbi két szintaxist támogatja:

- SCSS: a CSS szintaxis egy kiterjesztése.
 - Állomány kiterjesztés: `.scss`
- Behúzott szintaxis (*indented syntax*): a Sass eredeti szintaxisa, mely behúzást használ kapcsos zárójelek és pontosvesszők helyett.
 - Állomány kiterjesztés: `.sass`

Lásd: <https://sass-lang.com/documentation/syntax>

Sass: szintaxis (2)

Példa:

```
/* SCSS: */  
$size: 2em;
```

```
table.chessboard {  
  td {  
    height: $size;  
    width: $size;  
  }  
}
```

```
/* Behúzott szintaxis: */  
$size: 2em
```

```
table.chessboard  
  td  
    height: $size  
    width: $size
```

Sass: szintaxis (3)

- A behúzott szintaxis előnyei:
 - Az SCSS szintaxishoz képest tömörebb és könnyebben írható.
 - Kikényszeríti a tiszta kód írását.
- A behúzott szintaxis a Haml jelölőnyelvből jön, melyből a Sass származik.
 - Haml: <https://haml.info/>

Sass: érdekességek

- Történeti okokból a Sass azonosítóiban, mint például a változó-, függvény-, és mixin-nevekben, a kötőjelek és aláhúzójelek azonosként vannak kezelve.
 - Ez azt jelenti, hogy például `$font-size` és `$font_size` is ugyanarra a változóra hivatkozik.
 - Lásd: <https://sass-lang.com/documentation/variables/>

Sass: kifejezések

- A Sass támogatja kifejezések használatát, melyek mindegyike egy értéket eredményez.
- A Sass kifejezések szintaxisát SassScript-nek nevezik.
 - További információk:
<https://sass-lang.com/documentation/syntax/structure#expressions>
- A CSS-ben rendelkezésre álló adattípusokon túl a Sass továbbiakat is támogat (például logikai, null).

Sass: implementációk

- Dart Sass (programozási nyelv: Dart; licenc: MIT License)
<https://sass-lang.com/dart-sass> <https://github.com/sass/dart-sass>
 - A Sass elsődleges implementációja.
 - JavaScript-re fordítva npm csomagként is rendelkezésre áll (sass).
- Ruby Sass (programozási nyelv: Ruby; licenc: MIT License)
<https://sass-lang.com/ruby-sass> <https://github.com/sass/ruby-sass>
 - A Sass eredeti, ma már elavult Ruby implementációja.

Sass: telepítés

Lásd: <https://sass-lang.com/install>

- Az önálló Dart Sass letöltése:
<https://github.com/sass/dart-sass/releases/>
- A Sass telepítése az npm-mel:

```
# A Sass JavaScript implementációjának telepítése:  
npm install -g sass
```

Sass: szerkesztők

Szabad és nyílt forrású szoftverek:

- Visual Studio Code (platform: Linux, macOS, Windows; licenc: MIT License) <https://code.visualstudio.com/>
<https://github.com/Microsoft/vscode>
 - Kiterjesztések:
 - Live Sass Compiler <https://marketplace.visualstudio.com/items?itemName=glenn2223.live-sass>
<https://glenn2223.github.io/vscode-live-sass-compiler/>
<https://github.com/glenn2223/vscode-live-sass-compiler>

Lásd: <https://code.visualstudio.com/docs/languages/css>

Sass: online eszközök

- [CodePen](#)
- [Sass.js](#)

Sass: használat

Parancssori használat (Dart Sass):

```
sass input.scss output.css
```

```
sass input1.scss:output1.css input2.scss:output2.css
```

```
sass ./style/
```

```
sass --watch ./style/
```

Sass: kapcsolódó eszközök

SassDoc (licenc: MIT License) <http://sassdoc.com/>
<https://github.com/SassDoc/sassdoc>

- Egy dokumentációs eszköz a Sass-hoz.
- A Sass-ban `///` karakterekkel adható meg dokumentációs megjegyzés.
- A dokumentációs megjegyzések szövegének értelmezése Markdown-ként történik.
- A dokumentációs megjegyzésekben számos annotáció (például `@author`, `@parameter`, `@return`) áll rendelkezésre a részletek megadásához.

Sass: további információk

- [The Sass Way](#)
- [Sass Guidelines](#)

Stylus

- Webhely: <https://stylus-lang.com/>
- Tároló: <https://github.com/stylus/stylus>
- Implementáció:
 - Programozási nyelv: Node.js
 - Licenc: MIT License
- Állomány kiterjesztés: `.styl`

Stylus: lehetőségek

Lásd: <https://stylus-lang.com/docs/>

- Változók
- Aritmetikai műveletek
- Beépített függvények (például logikai, matematikai, sztring- és színkezelő függvények)
- Felhasználó által definiált függvények
- Asszociatív tömbök (hashes)
- Vezérlési szerkezetek (if, unless, for)
- Kiterjesztés
- ...

Stylus: szintaxis (1)

A Stylus az alábbi két szintaxist támogatja:

- CSS-stílusú szintaxis: a CSS szintaxis egy kiterjesztése.
- Behúzott szintaxis (*indented syntax*): behúzást használ kapcsos zárójelek és pontosvesszők helyett.

Lásd: <https://stylus-lang.com/docs/css-style.html>

Stylus: szintaxis (2)

Példa:

```
/* CSS-stílusú szintaxis: */ /* Behúzott szintaxis: */  
size = 2em;                size = 2em
```

```
table.chessboard {  
  td {  
    height: size;  
    width: size;  
  }  
}
```

```
table.chessboard  
  td  
    height size  
    width size
```

Stylus: kifejezések

- A Stylus támogatja kifejezések használatát, melyek mindegyike egy értéket eredményez.
 - Lásd: <https://stylus-lang.com/docs/operators.html>
- A CSS-ben rendelkezésre álló adattípusokon túl a Stylus továbbiakat is támogat (például logikai, null).

Stylus: telepítés

A telepítéshez Node.js és az npm csomagkezelő szükséges:

```
npm install -g stylus
```

Stylus: szerkesztők

Szabad és nyílt forrású szoftverek:

- Visual Studio Code (platform: Linux, macOS, Windows; licenc: MIT License) <https://code.visualstudio.com/>
<https://github.com/Microsoft/vscode>
 - Kiterjesztés: language-stylus <https://marketplace.visualstudio.com/items?itemName=sysoev.language-stylus>
<https://github.com/d4rkr00t/language-stylus>

Stylus: online eszközök

- [CodePen](#)

Stylus: használat

Parancssori használat:

```
stylus input.styl # kimenet írása az input.css-be
```

```
stylus -w input.styl
```

```
stylus ./style/
```

```
stylus -w ./style/
```

Mixin könyvtárak

- Less, Sass:
 - Hover (licenc: MIT License) <https://github.com/ianLunn/Hover>
<https://ianlunn.github.io/Hover/>
- Less, Sass, Stylus:
 - Open Color (licenc: MIT License) <https://yeun.github.io/open-color/>
<https://github.com/yeun/open-color>

CSS előfeldolgozók összehasonlítása

- Brian Jackson. *CSS Preprocessors – Sass vs LESS*. January 13, 2023.
<https://www.keycdn.com/blog/sass-vs-less>
- Jordan Irabor. *Less vs Sass vs Stylus*. February 20, 2022.
<https://stackshare.io/stackups/less-vs-sass-vs-stylus>

Forrástérképek (1)

- A forrástérképek (*source maps*) olyan állományok, melyek megmondják a böngészőknek és más CSS fogyasztó eszközöknek, hogy miként felel meg generált CSS kód azoknak az állományoknak, melyekből előállításra került.
- A forrástérképek segítségével a böngésző fejlesztőeszközök a felhasználó számára mutatni tudják az eredeti forrást, melyből CSS kód előállításra került.
 - Ez a lehetőség nyomkövetéshez hasznos.
- Vonatkozó specifikáció:
 - John Lenz, Nick Fitzgerald. *Source Map Revision 3 Proposal*. February 11, 2011. <https://sourcemaps.info/spec.html>

Forrástérképek (2)

CSS előfeldolgozó támogatás:

- Less: <https://lesscss.org/usage/#less-options-source-map-options>
 - Lásd a `--source-map` parancssori opciót. A forrástérképek nem állnak rendelkezésre a Less.js a böngészőben történő használatánál.
- Sass:
<https://sass-lang.com/documentation/cli/dart-sass#source-maps>
 - A Dart Sass alapértelmezésben forrástérképet generál minden általa létrehozott CSS állományhoz.
- Stylus: <https://stylus-lang.com/docs/sourcemaps.html>
 - Lásd a `--sourcemap/-m` parancssori opciót.

Forrástérképek (3)

Böngésző támogatás:

- Firefox:
 - [Style Editor – Source map support](#)
- Chromium, Google Chrome:
 - Meggin Kearney, Paul Bakaus, Sofia Emelianova. *Chrome DevTools – Debug your original code instead of deployed with source maps.*
<https://developer.chrome.com/docs/devtools/javascript/source-maps/>