

# HTML: A böngésző nyelve

Webprogramozás – 2. előadás

Sulyok Csaba

[csaba.sulyok@gmail.com](mailto:csaba.sulyok@gmail.com)



# 1. rész

HTML: bevezető

- ▶ *HyperText Markup Language* – hiperszöveges jelölőnyelv
- ▶ XML-alapú leíró nyelv
- ▶ Engedélyes
- ▶ Eredeti adottságai:
  - ▶ szövegek formázása
  - ▶ hiperlinkek (hyperlinks) összekötik az oldalakat
  - ▶ hipermédia megjelenítése: képek, hang- és videóanyag
  - ▶ táblázatok
  - ▶ űrlapok (formok) – távoli számítógépek által nyújtott szolgáltatások igénybevétele
- ▶ Komplexebb irányt vesz a legutóbbi, **HTML5** standard bevezetésével

- ▶ Tim Berners-Lee alkotta 1990-ben
- ▶ az SGML (Standard Generalized Markup Language, ISO 8879) sablonjára íródott
- ▶ kezdeti tartalom: bekezdések, listák, címek és hiperlinkek
- ▶ **IETF** (Internet Engineering Task Force) – létrehozott egy **HTML** munkacsoportot (HTML Working Group), mely 1995-ben kiadja a HTML 2.0 specifikációt (az első kísérlet a szabványosítás irányában)
- ▶ 1996-tól a **World Wide Web Consortium** (W3C) vállalja át a felelősséget az újabb HTML specifikációkért
- ▶ utoljára kiadott W3C szabvány: **HTML 5.2** – 2017. december
- ▶ 2000-ben nemzetközi standarddá válik (az **ISO HTML** a **4.01 Strict** verzióra alapoz)

- ▶ 2000-ben készült specifikáció, a HTML 4.01 egy szigorúbb, XML-alapú újrafogalmazása
- ▶ kötelező az XML (eXtensible Markup Language) standardoknak való megfelelés
- ▶ különbségek az engedékenyebb HTML-hez képest:
  - ▶ minden megnyitott elemet kötelező bezárni
  - ▶ attribútumok értékeinél az idézőjelek használata kötelező
  - ▶ tagek nem keresztezhetik egymást (pl. `<tag1><tag2></tag1>`)
  - ▶ kizárólag kisbetűs tagek használhatóak

- ▶ A HTML egyre szigorúbb és komplexebb szintaxis felé halad minden újabb verzióval.
- ▶ A **HTML 4.01** egyes tageket *elavult*nak minősülnek – használatuk alapján 3 dokumentumtípust különböztet meg:
  - ▶ **Strict** (szigorú specifikáció, elavult tageket nem enged használni)
  - ▶ **Transitional** (átmenet a régebbi használt stílus és az újabb, szigorúbb között)
  - ▶ **Frameset** (frame-eken alapuló oldalak)
- ▶ **HTML5** – a legfrissebb főverzió (major version) a következő új adottságokkal:
  - ▶ geolocations
  - ▶ hipermédia elemek: **canvas**, **audio**, **video**
  - ▶ local storage – offline adatok tárolása webalkalmazásokban
  - ▶ web workers – többszálú futtatás háttér munkás szálakkal
  - ▶ új szemantikus és strukturális elemek: **header**, **footer**, **sidebar**, **navbar**, **article**, **figure**

- ▶ A böngészők továbbra is “elnezőek” – azaz a *maguk módján* próbálják megjeleníteni a hibás dokumentumot
- ▶ Ezért szükséges leellenőrizni a dokumentum helyességét – megfelel-e egy adott specifikációnak
- ▶ <https://validator.w3.org/>
- ▶ Validálás: HTML5 és XHTML közti különbségek

## 2. rész

XML



# XML $\Rightarrow$ HTML: elemek



```
<cars>
  <car brand="Volkswagen" year="2010" kms="150000">
    <revision>2016-01-12</revision>
    <revision>2018-08-14</revision>
  </car>
  <car brand="BMW" year="2016" kms="70000" />
</cars>
```

▶ A HTML kölcsönveszi a *tag*, *elem* és *attribútum* fogalmait az XML-től

▶ **elem:**

- ▶ a dokumentum egy részét leíró egységes struktúra
- ▶ részei: kezdő tag, törzs (*content*), befejező tag - pl.  
`<revision>2018-08-14</revision>`
- ▶ egy üres elem (nincs tartalma) rövidre zárható - pl.  
`<car ...></car>` helyett `<car ... />`
- ▶ az elemek egymásba ágyazhatóak
- ▶ egy dokumentumnak csak egy gyökéreleme lehet (itt `cars`)

# XML $\Rightarrow$ HTML: tagek



```
<cars>
  <car brand="Volkswagen" year="2010" kms="150000">
    <revision>2016-01-12</revision>
    <revision>2018-08-14</revision>
  </car>
  <car brand="BMW" year="2016" kms="70000" />
</cars>
```

## ▶ tag:

- ▶ egy szövegrész *megjelölését* szolgálja
- ▶ nyitó tag: `<...>`, pl. `<cars>`
- ▶ záró tag: `</...>`, pl. `</cars>`
- ▶ üres elem nyitó tagje: `<.../>`, pl. `<car/>`
- ▶ kis- és nagybetű nem számít HTML esetén – XML esetén konvencionális kisbetűket használunk, ezért itt is ajánlott

# XML $\Rightarrow$ HTML: attribútumok



```
<cars>
  <car brand="Volkswagen" year="2010" kms="150000">
    <revision>2016-01-12</revision>
    <revision>2018-08-14</revision>
  </car>
  <car brand="BMW" year="2016" kms="70000" />
</cars>
```

## ▶ attribútum:

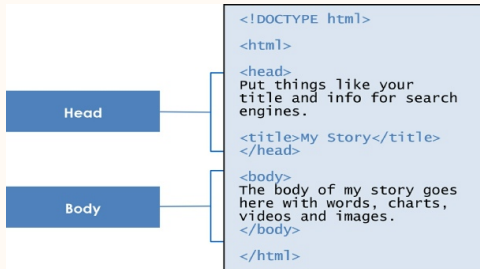
- ▶ egy elem különböző tulajdonságait határozza meg – pl. `year="2010"`
- ▶ kulcs-érték páros, melyet a nyitó tagben adunk meg
- ▶ idézőjelek használata XML-ben kötelező, HTML-ben csak ajánlott
- ▶ az attribútum nevében a kis-/nagybetűs szabály megegyezik a tagekkel, az értékben viszont számíthat.

## 3. rész

### HTML struktúra és elemek

# HTML állományok struktúrája

1. opcionális: egyetlen **BOM** (U+FEFF) karakter. (Byte Order Mark)
  2. **DOCTYPE** – jelzi a HTML verziót, amelynek a dokumentum megfelel
  3. fő **html** elem a következő alelemekkel:
    - ▶ **head** – metainformáció mely nem jelenik meg az oldalon, pl. cím (**title** – kötelező), encoding, külső állományok betöltése, stb.
    - ▶ **body** – a HTML fő teste, melyet a böngésző motra értelmez és kivetít.
- ▶ további részletek a [W3C oldalán](#)



## HTML 4.01 Strict:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

## HTML 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
http://www.w3.org/TR/html4/loose.dtd>
```

## XHTML Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

## HTML5

```
<!DOCTYPE html>
```

## HTML5:

```
<!DOCTYPE html>
<html>
<head>
  <title>Cim</title>
</head>
<body>
  <!-- tartalom ide -->
</body>
</html>
```

## XHTML:

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD Xhtml 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Cim</title>
</head>
<body>
  <!-- tartalom ide -->
</body>
</html>
```

- ▶ a jelölő nyelv számára fenntartott speciális karaktereket **karakterentitásokkal** szűrhetjük be:
  - ▶ `&entity_name;` - gyakoribb szimbólumok
    - ▶ “<” - `&lt;`;
    - ▶ “>” - `&gt;`;
    - ▶ szóköz mely nem töri meg a sort - `&nbsp;` (non-breaking space)
    - ▶ “&” - `&amp;`;
    - ▶ ” - `&quot;`;
  - ▶ `&#entity_number;` - Unicode decimális kód:
    - ▶ “<” - `&#60;`;
  - ▶ `&#xentity_number;` - Unicode hexadecimális kód:
    - ▶ “<” - `&#x3C;`;
- ▶ **kommentek** kezdete `<!--`, vége `-->` (ne legyen benne `--`)



# (Fontosabb) HTML elemek: head

**head**-en belül a dokumentummal kapcsolatos metaadatokat adunk meg:

- ▶ **title** - a dokumentum címe (kötelező)
- ▶ **meta** - különböző (meg nem jelenített) információ a böngésző illetve keresőmotrok számára
- ▶ **link** - jelzi a dokumentum kapcsolatát más dokumentumokkal (pl. stíluslappal).

Attribútumok:

- ▶ **href**="url"
- ▶ **rel**="relation" - megadja a csatolt külső állomány viszonyát az aktuális dokumentumhoz (pl. **rel**="stylesheet")
- ▶ **base** - relatív URL-ek kiindulási pontja
- ▶ **style** - belső stílus (ld. később CSS-ek esetén)

```
<head>
  <title>Document title</title>
  <meta charset="UTF-8">
  <meta name="description" content="Web Programming examples">
  <meta name="author" content="John Doe">
  <link rel="stylesheet" href="theme.css"> <!-- coming soon... -->
</head>
```

**body**-n belül:

- ▶ **h1**-től **h6**-ig – főcím, alcím, stb.
- ▶ **p** – bekezdés (paragrafus)
- ▶ **br** – újsor (üres)
- ▶ **pre** – megőrzi a fehér karaktereket
- ▶ **hr** – vízszintes vonal (üres)
- ▶ **div** – elhatárol egy részt a dokumentumon belül
- ▶ **span** – elemeket csoportosít (pl. egységes stílus érdekében)

```
<h1>1. Főcím</h1>
<h2>1.1. Alcím</h2>
<h3>1.1.1. Alalcím</h3>
```

```
<p>Egy bekezdés vagyok, s ez a tartalmam</p>
```

```
Ez egy szöveg, melyet sortörés (break) követ,<br>tehát ez új sorban jelenik meg, noha a kódban nem<br><br>
```

```
<div>Egy division vagyok. Későbbi formázásnál leszek hasznos.</div>
```

```
<!-- Ez egy komment, nem jelenik meg a weboldal kirajzolásakor -->
```

## 1. Főcím

### 1.1. Alcím

#### 1.1.1. Alalcím

Egy bekezdés vagyok, s ez a tartalmam

Ez egy szöveg, melyet sortörés (break) követ,  
tehát ez új sorban jelenik meg, noha a kódban nem

---

Egy division vagyok. Későbbi formázásnál leszek hasznos.

**body**-n belüli szövegformázó elemek:

- ▶ **b** - vastagított
- ▶ **strong** - hangsúlyozottabb szövegkiemelés (vastagított betűvel)
- ▶ **em** - szövegrész kiemelése (dőlt)
- ▶ **i** - dőltbetűs szöveg
- ▶ **code** - monospace (állandó betűszélességű) betűtípus, (alkalmas kódrészleteknek)
- ▶ **sub**, **sup** - alsó-, felső index

# (Fontosabb) HTML elemek



`<p><b>Vastagbetűs (bold) szöveg.</b></p>`

`<p><em>Kiemelt szöveg (emphasized, dőltbetűs).</em></p>`

`<p><b>Vastagbetűs szöveg <em>egy kiemelt résszel</em>.</b></p>`

`<p><code>Kódrészlet (monospace írógép betűtípus)</code></p>`

`<p>Pythagoras tétele:  $a^2 + b^2 = c^2$ </p>`

**Vastagbetűs (bold) szöveg.**

*Kiemelt szöveg (emphasized, dőltbetűs).*

**Vastagbetűs szöveg egy kiemelt résszel.**

Kódrészlet (monospace írógép betűtípus)

Pythagoras tétele:  $a^2 + b^2 = c^2$

Listák (nem számozott: `ul`, `li`, számozott: `ol`, `li`)

```
<ul>  
  <li>Első listaelem (li - list item) egy rendezetlen (vonalkás) listában (ul - unordered list)</li>  
  <li>Második elem</li>  
</ul>
```

```
<ol>  
  <li>Első elem egy rendezett (számozott) listában (ol - ordered list)</li>  
  <li>Második elem</li>  
</ol>
```

- Első listaelem (li - list item) egy rendezetlen (vonalkás) listában (ul - unordered list)
- Második elem

1. Első elem egy rendezett (számozott) listában (ol - ordered list)
2. Második elem

# (Fontosabb) HTML elemek



`img` - képbeszúrás (üres)

```
<p>Ez egy kép egy krumpliról átméretezve:</p>
```

```

```

Ez egy kép egy krumpliról átméretezve:



## Táblázatok (table, tr, th, td)

```
<table border="1" width="50%" align="center"> <!-- táblázat -->
  <tr> <!-- sor (table row) -->
    <th>Első fejléc cella (table header)</th>
    <th>Második fejléc cella</th>
  </tr>
  <tr>
    <td>Első sor, első cella adata (table data)</td>
    <td>Első sor, második cella</td>
  </tr>
  <tr>
    <td>Második sor, első cella</td>
    <td>Második sor, második cella</td>
  </tr>
</table>
```

Első fejléc cella (table header)	Második fejléc cella
Első sor, első cella adata (table data)	Első sor, második cella
Második sor, első cella	Második sor, második cella



# (Fontosabb) HTML elemek



**a** – horgony (anchor), hipelinket jelöl – **részletek**

```
<p>  
  <a href="http://www.google.com">  
    Egy hiperlink vagyok - a google.com-ra viszlek.  
  </a>  
</p>
```

```
<p>  
  <a href="http://www.google.com" target="_blank">  
    Ez a hiperlink új tabban nyílik meg.  
  </a>  
</p>
```

[Egy hiperlink vagyok - a google.com-ra viszlek.](http://www.google.com)

[Ez a hiperlink új tabban nyílik meg.](#)

Elemek típusa (a megjelenítés szempontjából):

- ▶ **blokk**-típusú elem: kihasználja a teljes rendelkezésre álló szélességet; előtte és utána sortörés. pl. `p`, `div`, `h1-h6`, `ul`, `ol`, `li`
- ▶ **inline** elem: az aktuális sorban jelenik meg, csak annyi szélességet foglal el, amennyi szükséges, nincs sortörés. pl. `a`, `em`, `span`, `i`, `b`

További átfogó tutorial: <http://www.learn-html.org/>

## HTML elemek

- ▶ [W3C - html5 elemek indexe](#)
- ▶ [w3schools - HTMLtagek indexe](#) (külön megjelölve, hogy mi az, amit a [HTML5](#) vezetett be)

## Attribútumok

- ▶ [W3C - HTML5 attribútumok indexe](#)
- ▶ [w3schools - HTMLattribútumok indexe](#)

## Form elemek:

- ▶ lehetővé teszik, hogy a felhasználó adatokat vigyen be a rendszerbe
- ▶ form adatok küldése: `submit` segítségével (a `form` tag `action` attribútumában megadott URL-re)
- ▶ ezen adatok feldolgozása (általában) szerver oldalon történik

## Részletesebb dokumentáció:

- ▶ lásd: [Form elemek](#)
- ▶ W3C dokumentáció: [Forms](#)
- ▶ w3schools példák: [HTML form elemek](#)

# Űrlapok (formok)



`<h1>Form példa</h1>`

`<p>Az alábbi leadásával (submit) a böngésző egy <code>POST</code> kérést küld az <code>index.html</code> végpontra.<br>Jelenleg ez nem fog semmit tenni :)</p>`

```
<form action="/index.html" method="POST">
  <p>Név: <input type="text" name="name" value="Alapértelmezett érték"/></p>
  <p>Jelszó: <input type="password" name="password"/></p>
  <p>Nem:
    <select name="gender">
      <option value="male">Férfi</option>
      <option value="female" selected>Nő</option>
      <option value="other">Más</option>
    </select>
  </p>
  <p><input type="submit" value="Leadom!"/></p>
</form>
```

Eredmény:

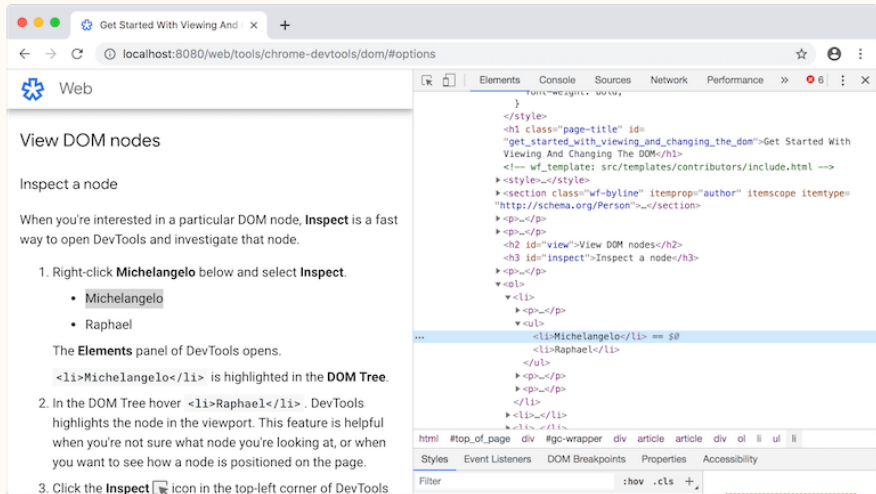
## Form példa

Az alábbi leadásával (submit) a böngésző egy POST kérést küld a `index.html` végpontra. Jelenleg ez nem fog semmit tenni :)

Név:

Jelszó:

Nem:  ▾



## 4. rész

### HTML5 szemantikus elemek



- ▶ **szemantikus elemek:** egyértelműen mondanak valamit az értelműkről/tartalmukról
  - ▶ pl. a **div** and **span** **nem** szemantikusak, míg a **form**, **img** és **table** **igen**
- ▶ HTML5-ben bevezetett új szemantikus elemek: **section**, **article**, **header**, **footer**, **main**, **nav**, **aside**, stb.
- ▶ ezek **nem** változtatják az oldal kinézetét közvetlenül, cserébe:
  - ▶ globálisan stilizálhatóak CSS segítségével (*TBD*)
  - ▶ segítenek a keresőmotoroknak az indexelésben (SEO – Search Engine Optimization)
- ▶ ajánlott olvasmány

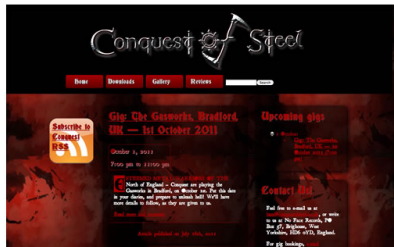
Egy átlagos **HTML5** oldal felépítése:

- ▶ **Fejléc** (Header) – az oldal tetején (logo, linkek stb)
- ▶ **Törzs** (Main content column) – a fontos tartalom megjelenítésére szolgáló rész
- ▶ **Oldalsáv**-ok (Sidebar) – a fő tartalomhoz kapcsolódó információk, vagy perzisztens információk pl. bevásárlókosár tartalma
- ▶ **Navigációs menü** (Navigation bar) – mehet oldalsávba vagy a fejlécbe
- ▶ **Lábléc** (Footer) – szerzői jog, elérhetőség stb.

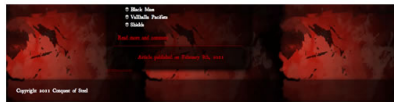
Néhány példa:

- ▶ <http://www.slashdot.org>
- ▶ <http://www.amazon.com>

# HTML oldal felépítése: példa



Rest of content



Rest of content

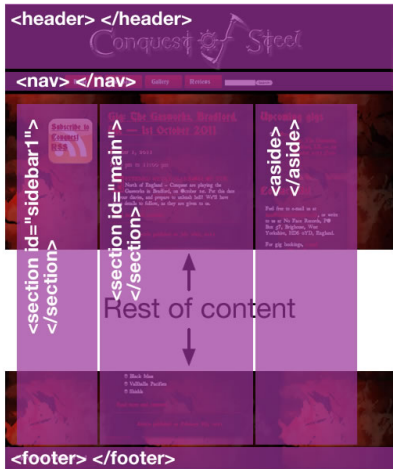


# HTML oldal felépítése: Példa HTML 4-ben



```
<body>
  <div id="header">
    <!-- fejlec tartalom -->
  </div>
  <div id="nav">
    <!-- navigacios menu -->
  </div>
  <div id="sidebar1">
    <!-- oldalsav tartalom -->
  </div>
  <div id="main">
    <!-- fo tartalom -->
  </div>
  <div id="sidebar2">
    <!-- oldalsav tartalom -->
  </div>
  <div id="footer">
    <!-- lablec -->
  </div>
</body>
```

# HTML oldal felépítése: Példa HTML 5-ben



```
<body>
  <header>
    <!-- fejléc tartalom -->
  </header>
  <nav>
    <!-- navigációs menu -->
  </nav>
  <section id="sidebar1">
    <!-- oldalsáv tartalom -->
  </section>
  <main>
    <!-- fő tartalom -->
  </main>
  <aside>
    <!-- fő tartalomhoz kapcsolódó kiegészítések -->
  </aside>
  <footer>
    <!-- labjegyzet -->
  </footer>
</body>
```

- ▶ A **section** elem feladata egy oldal (funkcionalitás vagy tartalom szerint) jól elkülöníthető részeinek megjelölése.
- ▶ Szintén használható egy cikk (**article**) több különböző részre való bontására. Az előző példában:
  - ▶ **sidebar1** - hasznos linkeket tartalmaz amelyek mindig elérhetőek az oldalon belül
  - ▶ **main** - tartalmazza az oldalon található blog posztokat
- ▶ Az **article** elem megjelöl egy egyedi, önmagában is értelemmel rendelkező részt a weboldal tartalmából
  - ▶ példák: blog posztok, videók, képek, hírbejegyzések, stb.

## ▶ header és footer

- ▶ Fejléc és láblécbe tartozó tartalom megjelölésére használatosak.
- ▶ Az előbbi példában a fejlécben található egy logo míg a láblécben található a szerzői jogi információ.

## ▶ aside

- ▶ Olyan információt tartalmazó részek megjelölését szolgálja, amelyek kapcsolatban vannak a fő tartalommal, de nem tartoznak konkrétan hozzá.
- ▶ Pl. a jobb oldali sidebar a példában *a csapat jövőbeli koncertjeinek dátumát és történetét tartalmazza*

## ▶ nav

- ▶ Navigációs linkeket tartalmazó rész.
- ▶ Alkalmas menünek, tartalomjegyzéknek, illetve az oldal bizonyos részeihez való ugrásra

- ▶ **address** – az oldal szerzőjének (esetleg szerzőinek) adatai/elérhetősége (NEM tetszőleges cím megadására!)
- ▶ **samp** (sample output) és **kbd** (keyboard input) – példa program kimenetére, illetve bemenetére – úgy jelenik meg, mint a **code**
- ▶ **abbr** – rövidítés
- ▶ **q** illetve **cite** – a **q** által megadott idézet forrását jelöli a **cite**
- ▶ **figure** és **figcaption** – egy kép, illetve a hozzá tartozó felirat



- ▶ a **h1**–**h6** címelemek adják meg az oldal szerkezetét
- ▶ együttesen képezik az oldal **vázlatát**
- ▶ pl. képernyőolvasók (*screenreader*) használják, hogy vakok számára lehetővé tegyék az oldalon való eligazodást
- ▶ online alkalmazás, mely láthatóvá teszi az oldal vázlatát:  
<https://gsnedders.html5.org/outliner/>
- ▶ **hgroup** (HTML5-specifikus elem)
  - ▶ lehetővé teszi a **h\*** elemek csoportosítását (ezek közül egy, a legfelsőbb szintű fog megjelenni a vázlatban)
  - ▶ pl. főcím **h1**-gyel, alatta egy mottó **h2**-vel, ami nem kell megjelenjen a vázlatban

```
<hgroup>
  <h1>Szünet</h1>
  <h2> ... jobb később, mint soha</h2>
</hgroup>
```