

# Unicode

Jeszenszky Péter

2022. szeptember 1.

# Mi a Unicode?

- Univerzális karakterkódolási szabvány írott karakterekhez és szöveghez.
- Lefedi a világ összes modern és ősi nyelvének összes karakterét.
- Tartalmaz továbbá műszaki szimbólumokat, írásjeleket és sok más olyan karaktert, melyet írott szövegben használnak.
- Széles körben használt és támogatott.

# Lefedettség

Példák:

- Cseroki: <https://www.unicode.org/charts/PDF/U13A0.pdf>
- Birodalmi arámi:  
<https://www.unicode.org/charts/PDF/U10840.pdf>
- Rovásírás:  
<https://www.unicode.org/charts/PDF/U10C80.pdf>
- Egyiptomi hieroglifák:  
<https://www.unicode.org/charts/PDF/U13000.pdf>
- Hangulatjelek:  
<https://www.unicode.org/charts/PDF/U1F600.pdf>
- Alkímiai szimbólumok:  
<https://www.unicode.org/charts/PDF/U1F700.pdf>

# Szabvány

- Fejlesztője a *Unicode Consortium*, mely egy non-profit szervezet.
  - Lásd: <https://www.unicode.org/consortium/consort.html>
- Az aktuális szabvány a 14.0.0 számú, melyen 2021. szeptember 14-én jelent meg.
  - Lásd: <https://www.unicode.org/versions/latest/>

# Universal Coded Character Set (UCS) (1)

- Az ISO által meghatározott szabványos karakterkészlet.
- Az aktuális szabvány: *ISO/IEC 10646:2020 Information technology – Universal Coded Character Set*  
<https://www.iso.org/standard/76835.html>
  - Letölthető innen: <https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

# Universal Coded Character Set (UCS) (2)

- Fejlesztése a *Unicode Consortium*-mal együttműködésben történik.
  - Mindkét szabványban megegyeznek a karakterek és kódjaik.
  - Az eltérés az, hogy a Unicode további megszorításokat szab az implementációk számára annak biztosításához, hogy egységesen kezeljék a karaktereket különböző platformok és alkalmazások között.
- További információk:
  - *Frequently Asked Questions – Unicode and ISO 10646*  
[https://www.unicode.org/faq/unicode\\_iso.html](https://www.unicode.org/faq/unicode_iso.html)

# Alapfogalmak

- **Kódtér (*codespace*)**: a karaktereket kódoló egész számok tartománya.
- **Kódpont (*code point*)**: a kódtér egy eleme, egy karaktert kódoló egész szám.

A kódpontokra történő hivatkozáshoz a szokásos gyakorlat a numerikus érték 4–6 számjeggyel ábrázolt hexadecimális alakjának megadása az `U+` előtag után.

- A vezető nullák elhagyása, de legalább 4 hexadecimális számjegy szükséges.
- Példák: `U+0020`, `U+265F`, `U+130E0`



# Tulajdonságok

- A karakterekhez (kódpontokhoz) jelentés hozzárendelése, melyet karakter tulajdonságok határoznak meg.
  - Száznál több karakter tulajdonság azonosítása, mint például:
    - Név
    - Általános kategória (betű, szám, szimbólum, írásjel, ...)
    - Kisbetű/nagybetű
- A Unicode Karakter Adatbázis (*Unicode Character Database*) (UCD) tartalmazza a karakter tulajdonságokat, mely itt elérhető: <https://unicode.org/ucd/>

# Karakternevek

Minden egyes karaktert egy egyedi karakternév azonosít, mint például:

- U+0041 – LATIN CAPITAL LETTER A (A) <https://www.fileformat.info/info/unicode/char/0041/index.htm>
- U+2605 – BLACK STAR (★) <https://www.fileformat.info/info/unicode/char/2605/index.htm>
- U+1F63A – SMILING CAT FACE WITH OPEN MOUTH (😺) <https://www.fileformat.info/info/unicode/char/1f63a/index.htm>

# Karakterek és karakterjelek

- Egy Unicode kódponttal azonosított karakter egy absztrakt entitás, mint például *LATIN CAPITAL LETTER A* vagy *BENGALI DIGIT FIVE*.
- Egy karakter vizuális ábrázolását **karakterjelnek** (*glyph*) nevezik.
  - Kapcsolódó fogalom: **betűtípus** (*font*)
- A Unicode szabvány nem határoz meg karakterjel képeket.
- A karakterek egy eszközön (például képernyőn vagy nyomtatón) történő vizuális megjelenése teljes egészében a karakterek rendereléséért felelős szoftverre vagy hardverre van bízva.

# Kódtér

- A kódteret a  $0_{16}$  és  $10FFFF_{16}$  közé eső egészek alkotják.
- Ez összesen 1 114 112 kódértéket jelent, melyekből jelenleg 144 697 használt csupán.
- Kódtáblák: <https://www.unicode.org/charts/>

# Síkok és blokkok

- A kódter felosztása síkoknak nevezett részekre, melyek mindegyike 65 536 ( $2^{16}$ ) kódpontot tartalmaz.
  - Egy kódpont utolsó négy hexadecimális számjegye a karakter helyét adja meg a síkon, a többi számjegy pedig magát a síkot jelöli.
    - Például az U+130F7 kódpont az 1. sík  $30F7_{16}$  sorszámú karaktere.
- A síkok száma 17 ( $0_{16}, \dots, 10_{16}$ ).
- A síkok felosztása blokkoknak nevezett nem átfedő részekre.
  - Egy blokk kódpontok egy névvel ellátott tartománya, melyben a kódpontok száma mindig 16 többszöröse.
  - Egy írásrendszer karaktereit több blokk tartalmazhatja.

# Basic Multilingual Plane (BMP)

- Az első 65 536 kódpontot (U+0000–U+FFFF) tartalmazó sík (0. sík).
- A gyakran használt karaktereket tartalmazza a világ összes modern írásrendszeréhez, valamint számos történelmi és ritka karaktert is tartalmaz.
- A BMP-be tartozik a Unicode karakterek többsége szinte minden szöveges adat esetén.

# Karakterkódolások

- A szabvány által meghatározott karakterkódolások:
  - UTF-8
  - UTF-16
  - UTF-32
- Mindhárom karakterkódolással ábrázolható valamennyi Unicode karakter.
- Az UTF a Unicode transzformációs formátum (*Unicode transformation format*) rövidítése.

- Minden kódpont ábrázolása 4 byte-on történik (rögzített szélességű kódolás).
- Ez a legegyszerűbb karakterkódolás.
- Feldolgozás szempontjából ez a leghatékonyabb, azonban tárolás szempontjából a legkevésbé hatékony.



# UTF-16

- Minden kódpont ábrázolása 2 vagy 4 byte-on történik (változó szélességű kódolás).
- A BMP karaktereinek ábrázolása 2 byte-on, az összes többi pedig 4 byte-on.
- Kompromisszumot képvisel a hatékony hozzáférés és a hatékony tárháshasználat között.

# UTF-8 (1)

- A kódpontok ábrázolása 1-4 byte-on (változó szélességű kódolás):
  - Az U+0000-U+007F tartományba eső kódpontokat egy byte ábrázolja (128 ASCII karakter).
  - Az U+0080-U+07FF tartományba eső kódpontokat 2 byte ábrázolja.
  - A BMP összes többi kódpontját 3 byte ábrázolja.
  - A BMP-n kívüli kódpontokat 4 byte ábrázolja.
- Egy kódpontot ábrázoló byte-sorozat első byte-ja meghatározza a sorozat hosszát (egyszerű feldolgozhatóság).

## UTF-8 (2)

- A használt byte-ok számának tekintetében a legtömörebb kódolás.
- Kevésbé hatékony kelet-ázsiai írásrendszerek esetén, mint például a kínai, japán és koreai.

# Byte-sorrend (1)

- Az UTF-16 és UTF-32 kódolási formákhoz a byte-sorrendet (*big-endian* vagy *little-endian*) is meg kell határozni.
- A byte-sorrend szerint az alábbi karakterkódolási sémák megkülönböztetése:
  - UTF-8
  - UTF-16, UTF-16BE, UTF-16LE
  - UTF-32, UTF-32BE, UTF-32LE
- Az UTF-16 és UTF-32 kódolási sémáknál a byte-sorrendet a BOM adja meg a szöveg elején.

## Byte-sorrend (2)

### Byte-sorrend jelző (*byte order mark*) (BOM):

- A byte-sorrend jelzésére az U+FEFF (*ZERO WIDTH NO-BREAK SPACE*) karakter használata.
- Ezt a karakter nem a szöveg része és a feldolgozás előtt el kell távolítani!

Kódolási séma	Byte-sorozat
UTF-16 big-endian	FE FF
UTF-16 little-endian	FF FE
UTF-32 big-endian	00 00 FE FF
UTF-32 little-endian	FF FE 00 00

# ISO/IEC 8859

- 8-bites karakterkódolási szabványok (ISO/IEC 8859-1, ... , ISO/IEC 8859-16).
  - Lásd: ISO/IEC 8859 – 8-bit single-byte coded graphic character sets
- Számunkra fontosak:
  - ISO/IEC 8859-1 (Latin-1): a nyugat-európai nyelvekhez
  - ISO/IEC 8859-2 (Latin-2): a közép-európai nyelvekhez
    - Az albán, bosnyák, cseh, horvát, lengyel, magyar, német, román, szerb (latin betűs írás), szlovák, szlovén, szorb nyelvekhez alkalmas.

# Unicode és programozási nyelvek

- A modern programozási nyelvek általában a Unicode-on alapulnak, azaz a forrásprogramok Unicode karaktersorozatok.
- A Unicode-on alapuló programozási nyelvek: C#, ECMAScript, Java, Kotlin, Python, Swift, ...

- Unicode karakterek megadásához használhatunk `\hhhhh` formájú vezérlősorozatokat, ahol `hhhhh` a Unicode karakter kódpontját ábrázoló legalább egy és legfeljebb 6 karakterből álló hexadecimális számjegysorozat.
  - Ha 6-nál kevesebb a számjegyek száma és a `[0-9a-fA-F]` karakterek valamelyike követi az utolsó számjegyet, akkor a vezérlősorozat végének jelzéséhez egy tetszőleges *whitespace* karaktert kell megadni.
    - Egy vezérlősorozatot követő *whitespace* karakter figyelmen kívül lesz hagyva.
  - Példa:
    - `\9`, `\09`, ..., `\000009` a vízszintes tabulátor karaktert jelenti.
    - `\A9`, `\0A9`, ..., `\0000A9` a copyright szimbólumot (©) jelenti.
- Lásd: <https://www.w3.org/TR/css-syntax-3/#escaping>



# ECMAScript

- Sztring literálokban, reguláris kifejezés literálokban, sablon literálokban és azonosítókban bármely Unicode karakter kifejezhető az alábbi formájú Unicode vezérlősorozatokkal is:
  - `\uhhhh`, ahol `hhhh` a kódpontot ábrázoló négy hexadecimális számjegy.
    - Példa: `\u00A9`, `\u262F`
  - `\u{hhhhhh}`, ahol `hhhhhh` is a kódpontot ábrázoló hexadecimális számjegysorozat, mely legalább 1 és legfeljebb 6 számjegyből áll.
    - Példa: `\u{A9}`, `\u{1F63A}`
- Lásd:  
<https://262.ecma-international.org/12.0/#sec-source-text>




# JSON

- Sztringekben a BMP-hez tartozó Unicode karakterek megadhatóak `\u` formájú vezérlősorozatokkal, ahol *hhhh* a kódpontot ábrázoló négy hexadecimális számjegy.
  - Példa: `\u00A9`, `\u262F`
- Lásd: <https://www.rfc-editor.org/rfc/rfc8259#section-7>

- Szövegben, attribútumértékekben és literális egyed értékekben Unicode karakterek kifejezhetők az alábbi formájú karakterhivatkozásokkal:
  - `&#nnnn;`, ahol *nnnn* a kódpontot ábrázoló decimális számjegysorozat.
    - Példa: `&#169;`, `&#9775;`, `&#128570;`
  - `&#xhhhh;`, ahol *hhhh* a kódpontot ábrázoló hexadecimális számjegysorozat.
    - Példa: `&#xA9;`, `&#x262F;`, `&#x1F63A;`
- Lásd: <https://www.w3.org/TR/xml/#dt-charref>

- Számos Unicode karakter fejezhető ki *&név*; formájú nevesített karakterhivatkozásokkal.
  - Példák:
    - `&ampEacute`; (U+00C9 = É)
    - `&ampeacute`; (U+00E9 = é)
    - `&ampstar`; (U+2606 = ☆)
- Támogatott karakterhivatkozás nevek: <https://html.spec.whatwg.org/#named-character-references>

# Unicode karakterek bevitele

- Linux: GTK+ alkalmazásokban használjuk a  +  +  billentyűkombinációt, mely után adjuk meg a hexadecimális kódpontot.
- Lásd:
  - *Unicode beviteli módszerek* [https://hu.wikipedia.org/wiki/Unicode\\_beviteli\\_m%C3%B3dszerek](https://hu.wikipedia.org/wiki/Unicode_beviteli_m%C3%B3dszerek)
  - *Unicode input* [https://en.wikipedia.org/wiki/Unicode\\_input](https://en.wikipedia.org/wiki/Unicode_input)

# Karakterkódolás felismerése

- Unix-szerű rendszerekben a `file` parancs használható szövegállományok karakterkódolásának meghatározásához.

- Példa a használatra:

```
file --mime-encoding file.txt
```

```
file --mime-encoding *.txt
```

- Lásd: <https://man7.org/linux/man-pages/man1/file.1.html>

# Konverziós eszközök (1)

iconv:

- Webhely: <https://www.gnu.org/software/libiconv/>
- Tároló: <https://savannah.gnu.org/projects/libiconv/>
- Programozási nyelv: C
- Licenc: LGPLv2.1
- Példa a használatra:

```
iconv --list
```

```
iconv -f UTF-8 -t LATIN2 input.txt -o output.txt
```

# Konverziós eszközök (2)

## Recode:

- Tároló: <https://github.com/rrthomas/recode/>
- Programozási nyelv: C
- Licenc: GPLv3
- Példa a használatra:

```
recode --list  
recode UTF-8..ISO-8859-2 file.txt  
recode UTF-8..UTF-16 *.txt
```



# Online eszközök

- *Shapecatcher: Draw the Unicode character you want!*  
<https://shapecatcher.com/>
- *Unicode Character Search*  
<https://www.fileformat.info/info/unicode/char/search.htm>
- *Unicode Character Table* <https://unicode-table.com/>
- *Unicode Party: The Unicode Emoji Search Engine*  
<https://unicode.party/>
- &what; <http://www.amp-what.com/>

# Ajánlott irodalom

- Victor Stinner. *Programming with Unicode*.  
<https://unicodebook.readthedocs.io/>  
[https://github.com/vstinner/unicode\\_book/](https://github.com/vstinner/unicode_book/)