

# INFORMÁCIÓ

Az információ latin eredetű szó, amely értesülést, hírt, üzenetet, tájékoztatást jelent. Egyben az informatika alapfogalma. Számos jelentése, kifejtése ismert, különböző tudományágak különböző módon közelítik meg, írják le. Egyértelműen elfogadott definíciója nem ismert.

Általánosságban információnak azt az adatot, hírt tekintjük amely számunkra releváns és ismerethiányt csökkent. Egyik legegyszerűsített megfogalmazás szerint az információ nem más, mint valóság (vagy egy részének) visszatükröződése. Tudományos értelmezését az információelmélet fogalmazza meg.

**Mennyisége:** Az információ mennyisége az idővel exponenciálisan nő, ugyanis az információ felhasználásakor nem semmisül meg, többszörösen újrafelhasználható.

**Minősége:** Az információ minősége alapvetően a feldolgozás és a tömörítés szintjétől, valamint az információátvitel minőségétől függ. Ide tartozik, hogy az információ mennyire megbízható, pontos, illetve mennyire időszerű.

**Mértékegysége:** Az információmennyiség mértékegysége a bit

## I. KÓDOLÁSOK

### 1.1. ANSI:

Az ANSI (American National Standards Institute, Amerikai Nemzeti Szabványügyi Intézet) egy nonprofit magánvállalkozás, melynek célja ipari szabványok kidolgozása az Amerikai Egyesült Államok számára. Az ANSI szabványai széles körben elterjedtek. Informatikában **az ANSI szabványosította az ASCII karakterkészletet**, illetve az irányító szekvenciákat (például az X3.64 szabályozza, hogy egy terminál képernyőjén a kurzor miként mozgatható.) A Microsoft Windowsban az ANSI kifejezés az operációs rendszer által támogatott karakterkészletet jelenti, amely Észak-Amerikában és Nyugat-Európában a CP1252, Magyarországon a CP1250. Ezek a kódolások nagyban hasonlítanak az ISO 8859-es karaktertáblához, ezért többen tévesen azt hiszik, hogy a kettő pontosan megegyezik.

### 1.2. ASCII:

Az ASCII (angol betűszó: American Standard Code for Information Interchange) egy karakterkészlet és karakterkódolási szabvány, amely a latin ábécén alapul és az angol nyelvben és sok nyugat-európai nyelvben használatos betűket tartalmazza. Leggyakoribb felhasználása a számítógépeken illetve szövegeket kezelő egyszerű eszközökön használt szövegek reprezentációja. Az Unicode 32-től 126-ig (hexadecimálisan: 0020-007E) tartó szakasza, a latin ábécé betűit és a fontosabb írásjeleket tartalmazza. A kódrendszer 128 karakterhelyet tartalmaz, ezek az alábbi táblázatban a Decimális oszlopban találhatóak. Az első 32 karakter (0–31), valamint a 127 kódú karakter úgynevezett vezérlőkódokat tartalmaz.

Szöveges adatokhoz használt szabványos egybájtos karakterkódolási séma. Az ASCII hét vagy nyolc bit kombinációját használja a **128 vagy 256 lehetséges karakter** előállítására. A szabványos ASCII **hét bittel** állítja elő az amerikai angolban használt összes kis- és nagybetűt, a számokat 1-től 9-ig, a központosítási jeleket és a speciális vezérlőkaraktereket. A legtöbb jelenleg használt x86 alapú rendszer támogatja a bővített ASCII-karakterek használatát. A bővített ASCII engedélyezi a nyolcadik bit használatát, amivel további 128 jelet lehet előállítani, köztük betűket az angoltól eltérő különböző nyelvekhez, valamint grafikai jeleket.

### 1.3. Unicode :

A Unicode (ejtsd: junikód) a különböző írásrendszerek egységes kódolását és használatát leíró nemzetközi szabvány. A Unicode nem csak a kódolással, hanem a karakterek osztályozásával, megjelenítésével és használatával is részletesen foglalkozik. Az Unicode szabvány **16 biten tárolt** síkokra osztja a szabvány legutóbbi változatában rögzített mintegy 100 ezer karaktert. Az első sík (BMP – Basic Multilingual Plane) mintegy 64 ezer karakterhelyén a legtöbb ma használatos jelet megtalálni.

### 1.4. UTF-8:

A teljes, karakterenként 4 byte-ot foglaló UTF-32 kódolással szemben a tömörebb UTF-8 kódolás a legelterjedtebb. Az UTF-8 változó hosszúságú kódolással (1–6 byte) képezi le a Unicode karaktertáblát. 1 byte-on tárolt kódjai az ASCII-nak felelnek meg, így a latin betűs UTF-8 kódolású szövegek a régi ASCII környezetben is olvashatók maradnak.

Az UTF-8 (8-bit Unicode Transformation Format, 8 bites Unicode átalakítási formátum) változó hosszúságú Unicode karakterkódolási eljárás, melyet Rob Pike és Ken Thompson alkotott meg. Bármilyen Unicode karaktert képes reprezentálni, ugyanakkor visszafelé **kompatibilis a 7 bites ASCII szabvánnyal**. Emiatt egyre inkább az internetes karakterkódolás standardjává válik.

Az angolszász, majd az európai országokból kiindulva az ASCII után először az úgynevezett Latin-1 kódolás terjedt el, ami tartalmazza az összes angol nyelvhez szükséges betűt, illetve számos európai nyelv betűit, de például a magyar „ő” és „ú” betűket **nem (ezek helyett – helytelenül – gyakran használják a hullámos illetve a kalapos betűket: û ô vagy õ)**. Magyarhoz lehet azonban a Latin-2 (közép-európai) kódolást is használni, ami ismeri az ő és ú betűinket, de nem ismer más fontos betűket, például a cirill, görög, vagy például az örmény, indiai, arab és héber betűket, a kínai írásjegyeket és a japán kanákat. A Unicode és az UTF-8 kódolás egyszerre támogatja mindezen karakterek megjelenítését, és így minden nyelv egységes kódolást tud használni, megelőzve a betűk nem tervezett „átalakulását”.

Az UTF-8 kódolás lényege, hogy a 7 bites ASCII kódtábla karaktereit (angol kis- és nagybetűk, számok és gyakoribb írásjelek) az ASCII kódjukkal jelöli, egyéb karakterek kódját pedig „feldarabolja”, és a darabokat egy vezérlőjelet követő több egymás utáni bájtban helyezi el úgy, hogy a bájtok mindegyike 127 felett van (azaz nem téveszthető össze a 7-bites ASCII kódok egyikével sem). Ennek számos előnye van: 7-bites ASCII szöveg UTF-8 kódja saját maga, az UTF-8 szöveget tévesen valamilyen más elterjedt kódolásnak (például ISO-8859.1-nek) értelmezve általában nem kapunk nagyon rossz eredményt (magyar szöveg esetén csak az ékezetes karakterek és a gondolatjel és egyéb speciális tipográfiai jelek helyén lesz két-két értelmetlen karakter), a félbeszakadt UTF-8 karakterről egyértelműen felismerhető, hogy hibás, és nagyrészt ASCII szöveg kódolása nem igényel sok extra bájtot.

## II. SZÁMÁBRÁZOLÁS

### II.1. Lebegőpontos számábrázolás

A lebegő pontos számábrázolásnál lehet tört számot is ábrázolni. Úgy, hogy felírjuk egy törtszám abszolút értékét "0.5 és 1" közé eső szám 2 valahányadik hatványának felírjuk  
PL:  $3,6=0,9 \cdot 2^2$

A számokat ebben az esetben normalizált alakban használjuk.

$$N_2 = \pm M \cdot 2^{\pm E}$$

, ahol  $N_2$  az ábrázolandó bináris szám  
M normalizált mantissza  
E karakterisztika

A mantisszát leggyakrabban előjeles abszolút értékes formátumban tárolják. A normalizálásra kétféle gyakorlat terjed el.

### a) Töltre normalizálás

---

A bináris pontot addig toljuk el, amíg a mantissza értéke 1/2 és 1 közötti értékű nem lesz. Például:

$$N_2 = 0,0001011001_2 = 0,1011001 \cdot 2^{-3}$$

Mivel a  $2^{-1}$  helyértéken lévő bit mindig 1 értékű, ezért a szám eltárolása előtt kiveszik. Ezt **implicitbit**nek hívják. Így a tárolt mantissza (m) értéke:

$$m: 011001000..$$

### b) Egészre normalizálás

---

Ez esetben a normalizált mantissza értéke 1 és 2 közé esik. Például:

$$N_2 = 0,0001011001_2 = 1,011001 \cdot 2^{-4}$$

Itt az egészek helyén áll mindig 1, ezért tárolása szükségtelen. A tárolt mantissza azonos az előzővel:

$$m: 011001000..$$

Természetesen műveletvégzés előtt mindkét esetben a nem tárolt biteket vissza kell helyezni, hiszen ellenkező esetben hibás eredményt kapnánk.

A karakterisztikához egy egész számot adnak hozzá, és így tárolják. Ezt a megoldást **eltolt vagy ofsztet karakterisztikának** hívják. Az eltolásra azért van szükség, hogy a karakterisztikát eltoljuk a pozitív számok tartományába, és így nem kell az előjelét ábrázolni. Az eltolás mértékére (d) két megoldás használatos:

$$d = 2^{k-1} - 1$$
$$d = 2^{k-1}$$

ahol k a karakterisztika ábrázolására szánt bitek száma. Így az eltolt karakterisztikát (c) a

$$c = E + d$$

összefüggéssel számíthatjuk ki.

(forrás: <http://www.freeweb.hu/kondy/lebego.html>)