

# ATTO script language / interpreter

User Manual 1.0

Az Atto nyelv interpretere egy procedurális paradigmát támogató, szimpla, minimalista interpreter, amely alapvető vezérlési struktúrákat, aritmetikai és logikai műveleteket, valamint címkéket támogat. Az alábbiakban bemutatásra kerül a Turing-teljes Atto nyelv szintaxisa, operátorai, változó-használata és a nyelv kulcsszavai.

## Az interpreter helyes használata

**atto.exe** < **inputfile** > **outputfile**

Ahol is az **inputfile** a futtatni kívánt ATTO szintaxisú kódot (szkriptet) tartalmazza, az **outputfile** pedig a futás során létrejövő kimenet.

Az Atto programokban (szkriptekben) minden utasítást, operátort, változót, számértéket, nyelvi elemet legalább egy SPACE karakter kell, hogy elválasszon a másiktól. Az ATTO szkriptek írása során a következő elemek használhatók:

## Változó-azonosítók

Egy karakteres változónevek (azonosítók), A-tól Z-ig terjedően.

Az A-Q változók 32 bites, előjeles egész számot tárolnak.

Az R változó véletlenszám generálására használható.

Az S-Z változók egy bájtós értéket tárolnak, ASCII karakterként írathatók ki (PRN).

## Kulcsszavak és szintaxis

A nyelv mindössze négy kulcsszót **IF**, **JMP**, **PRN**, **RET** támogat. Mindegyik kulcsszó specifikus feladatot lát el:

- **IF**: Feltételes utasítás, amely végrehajt egy utasítást (Jump, Print vagy értékadás), ha a mögötte álló feltétel teljesül. Az IF-nek "ELSE" ága nincs. Az **IF** kulcsszó után a következő szintaxis használható:

**IF** [**feltétel**] [**művelet**]

Példa: **IF A > B JMP .LABEL**

- **JMP**: Ugró utasítás, amely a mögötte álló címke sorszáma ugrik.

Példa: **JMP .END**

- **RET**: Return utasítás, amely az utolsó ugrás előtti címre adja a vezérlést. Az interpreter egy belső változóból (pseudo stack) nyeri ki annak a sornak a számát, amelyre a RET végrehajtása során visszatér.

- **PRN**: Nyomtatási utasítás, amely egy változó értékét vagy ASCII kód szerinti értelmezését írja ki. A **PRN** csak beépített változók (**A-Z**) értékét képes kinyomtatni.

**PRN A** vagy **PRN A U T S**

Ha a változó azonosítója az S-Z tartományban van, az értéke ASCII kódnak minősül és annak megfelelő karakterként íródik ki.

**Aritmetikai Műveletek** - Az **ATTO** nyelv az alábbi aritmetikai operátorokat támogatja:

**+** : Összeadás  
**-** : Kivonás  
**\*** : Szorzás  
**/** : Egész osztás  
**%** : Maradékképzés

**Logikai Operátorok** - A nyelv logikai operátorai:

**<** : kisebb, mint  
**>** : nagyobb, mint  
**#** : nem egyenlő  
**=** : egyenlő

## Inkrementer, Dekrementer, Multiplikátor

A változók közvetlen növelésére, csökkentésére vagy megszorzására használhatók a következő szimbólumok:

**+** : Inkrementálás (értéknövelés 1-gyel (ugyanaz mint  $A = A + 1$ ))  
**-** : Dekrementálás (értékcsökkentés 1-gyel (ugyanaz mint  $A = A - 1$ ))  
**\*** : Multiplikálás önmagával (ugyanaz mint  $A = A * A$ )

Szintaktikai példa:

A + (inkrementálja A értékét),  
B - (dekrementálja B értékét),  
C \* (négyzetre emeli C értékét).

## Címkék

A címkék (például **.LOOP**) a program különböző részeire mutatnak, ahová a **JMP** utasítással ugorhatunk. A címkék első karaktere kötelezően a **.** (pont). Címkék (és az **IF/JMP**) segítségével építhetők számlálás, illetve elől- vagy hátul-tesztelő ciklusok is (lásd alább). Egy címke és a **RET** utasítás közé pedig szubrutinok sorait helyezhetjük el.

## Komment sorok

A programkód kommentjeként használhatók a **;** karakterrel kezdődő sorok. Ezek a futtatás során figyelmen kívül maradnak. A program futásának eredményét nem befolyásolják.

## String-ek

Az **ATTO** interpreter ASCII kód alapú karaktereket támogat, amelyekből stringek állíthatók össze. A **PRN** utasítás használatával ezen változók (S..Z) értékei ASCII karakterként íródnak ki.

## Matematikai és logikai kifejezések

Az **ATTO** nyelven íródott szkriptekben csak un. szimpla matematikai vagy logikai kifejezések használhatók. Ez annyit jelent, hogy csak olyan kifejezések megengedettek, amelyekben egy matematikai illetve egy logikai operátor van. Zárójelezést a nyelv nem ismer. Az összetett matematikai/logikai kifejezéseket fel kell bontani.

**A = B + 4** (megfelel), **A = B + 4 \* 2** (nem felel meg (két matematikai operátor (+,\*) van a kifejezésben))

**IF A < B JMP .LOOP** (megfelel), **IF A < B + 2 JMP .LOOP** (nem felel meg, mert két operátor van a kifejezésben)

## Ciklusok képzése

```
B = 0
.LOOP
B +
PRN B
IF B < 9 JMP .LOOP
```

```
-----

B = 0
.LOOP
IF B > 5 JMP .END
B +
PRN B
JMP .LOOP

.END
```

A **TRC** egy extra kulcsszó, ami a nyomkövetést kapcsolja be. Ez annyit tesz, hogy a kimeneti fájl-ba íródik a futtatott kód listája, a címkék neve és sorszáma, a változók esetében pedig a futtatás végi állapot. a TRC-t a szkript első sorában kell elhelyezni.