

Python parancsok

- `[]` – egy nyitó és egy záró szögletes zárójel, listát lehet vele megadni.
- `abs()` – megadja a zárójelben lévő szám abszolút értékét.
- `and` – if vagy elif után, illetve a while-ciklus feltételei között szerepelhet, és azt lehet vele kikötni, hogy a két oldalán megadott két feltételnek egyszerre kell teljesülnie. Magyarul: és.
- `append()` – nem önálló parancs, hanem a lista adattípus tagfüggvénye. Magyarul: hozzáfűz. Listához tudunk vele új elemet hozzáfűzni.
- `bool, bool()` – Az első szó egy adattípus, a Boolean neve a Pythonban. A szó jelentése Bool-i, de magyarra “logikai” változóként fordítjuk. Értéke True (igaz), vagy False (hamis) lehet. A második egy függvény, ami logikai értékke alakítja azt, ami a zárójelben van. Ha lehet.
- `break` – arra való, hogy kilépjünk egy ciklusból, még mielőtt az végigfutna. Magyarul: megszakítás.
- `close()` – nem önálló parancs, hanem a fájl típus tagfüggvénye. Megnyitott fájl bezárására való. Magyarul: bezár.
- `count()` – nem önálló parancs, hanem a lista adattípus tagfüggvénye. Magyarul: megszámol. Megmondja, hogy a zárójelben szereplő elem hányszor fordul elő a listában.
- `def` – függvényt lehet vele megadni, azaz definiálni. Nem is azt jelenti, hogy def, azaz süket, hanem a define, magyarul: definiál szó rövidítése. .
- `elif` – elágazáskor a további (az if-ág nem-teljesülése esetén még megvizsgálandó) feltételeket vezeti be. Az else és az if szó összevonásából született meg, ilyen angol szó nincs. Ha programkódot olvasol, akkor az elif olvasása “különb, ha”
- `else` – elágazáskor az utolsó ágat vezeti be, amely akkor fut le, ha egyik korábbi ág feltétele sem teljesült. Magyarul: különben.
- `extend()` – nem önálló parancs, hanem a lista adattípus tagfüggvénye. Magyarul: bővít. Egy listával bővíti az eredeti listát: az új listát a régi végére biggyeszti.
- `False` – a bool adattípus egyik lehetséges értéke. Magyarul: hamis. Mindig nagy betűvel kezdjük.
- `float, float()` – Az első szó egy adattípus, a floating point number, magyarul lebegőpontos szám típust jelzi. A második egy függvény, ami lebegőpontos számmá alakítja azt, ami a zárójelben van. Ha lehet.
- `for` – a bejárás, vagy más néven for-ciklus első szava. Nem tudom jól lefordítani.
- IDE – ez nem parancs, az integrált fejlesztői környezet (Integrated Development Environment) rövidítése. Tulajdonképpen egy kódszerkesztőprogram, pár egyéb képességgel. A Python saját IDE-je az IDLE, de sok más IDE is használható hozzá.
- IDLE – ez nem parancs, hanem a Python fejlesztői környezetének neve. Van jelentése is, mert a készítőik vicces kedvükben voltak. Magyarul: tétlen.
- `if` – a programban a elágazás első feltételét vezeti be. Magyarul: ha.
- `index()` – nem önálló parancs, hanem a lista adattípus tagfüggvénye. Visszaadja a zárójelben megadott elem első előfordulásának indexét a listában.
- `input()` – választ vár a felhasználótól. A zárójelben elhelyezhetsz kérdést, hogy a felhasználó tudja, hogy mire kell válaszolnia. A választ érdemes változóban tárolni. Magyarul: bemenet, bemenő jel.
- `insert(hova, elem)` – nem önálló parancs, hanem a lista adattípus tagfüggvénye. Magyarul: beszúr. Két paramétere van, a listába az új elem-et a lista “hova” helyére szúrja be.
- `int, int()` – Az első szó egy adattípus, az integer, magyarul egész szám típust jelzi. A második egy függvény, ami egész számmá alakítja azt, ami a zárójelben van. Ha lehet.
- `import` – modult tölthetünk be vele. Illik a program elején szerepeltetni.
- `list()` – lista létrehozására szolgál. Magyarul: lista.
- `lower()` – nem önálló parancs, hanem a string adattípus tagfüggvénye. A stringet kisbetűssé alakítja. Ha eddig is kisbetűs volt, nos, annyi baj legyen! A lower case magyarul kisbetűt jelent.
- `join()` – nem önálló parancs, hanem a string adattípus tagfüggvénye. A zárójelben egy olyan bejárható objektumot, például listát adhatunk meg, ami további stringeket tartalmaz. A zárójelben lévő objektum stringeit úgy fűzi egygyé (azaz konkatenálja) a join, hogy közéjük az előtte szereplő stringet teszi. Például az `'X'.join(['alma', 'körte', 'szilva'])` eredménye `almaXkörteXszilva`. Magyarul: illeszt, összeforr, egybeköt, egyesít.
- `len()` – megmondja, hogy hány elemű a zárójelben lévő lista, vagy azt, hogy hány karakteres a zárójelben lévő karakterlánc. A length, azaz hossz szóból származik, de ilyen angol szó nincs.
- `map, map()` – Az első szó a map (magyarul: leképez) objektumtípust jelzi, a másik egy függvény, amivel map típusú objektumok elő. A függvény két paramétert vár: az első egy másik függvény neve, a második egy bejárható objektum, például lista. A `map()` függvény a megadott függvényt a lista minden elemére alkalmazza. Például a `map(str, [1, 2, 3])` parancs az `[1, 2, 3]` lista minden egyes tagjára lefuttatja az `str()` függvényt.
- `max()` – A zárójelben egy bejárható objektumot, például listát, vagy range-t adunk meg. A `max()` függvény megadja a bejárható objektumban lévő számok közül a legnagyobbat, ha karakterláncokat vizsgálunk, akkor az ábécérendben a leghatúl állót.

- [min\(\)](#) – A zárójelben egy bejárható objektumot, például listát, vagy range-t adunk meg. A min() függvény megadja a bejárható objektumban lévő számok közül a legkisebbet, ha karakterláncokat vizsgálunk, akkor az ábécérendben a legelől állót.
- [None](#) – változónak adható értékül, ha semmilyen konkrét értéket nem akarunk a változóban tárolni, vagy az eddig benne tároltát óhajtjuk “kidobni”. Magyarul: semmi. Mindig nagy betűvel kezdjük.
- [open\(\)](#) – nem önálló parancs, hanem a fájl típus tagfüggvénye. Fájl megnyitására való. Magyarul: nyit.
- [or](#) – if vagy elif után következhet, és azt lehet vele kikötni, hogy a megadott két feltétel közül elég, ha az egyik teljesül. Magyarul: vagy.
- [pop\(\)](#) –nem önálló parancs, hanem a lista adattípus tagfüggvénye. Kiveszi a listából és visszaadja azt, ami a zárójelben van. Ha nem írunk semmit a listába, akkor a lista utolsó elemével teszi ugyanezt. Magyarul: kipattint, itt inkább kiugraszt.
- [print\(\)](#) – kiírja, ami a zárójelben van. Magyarul: nyomtasd.
- [random](#) – ez nem parancs, hanem a véletlenszámok kezeléséhez szükséges modul neve.
- [random.randint\(\)](#) – véletlen egész számot állíthatunk elő vele. Az alsó és a felső határt a zárójelben, vesszővel elválasztva adod meg.
- [range, range\(\)](#) – Az első szó a range (magyarul: sorozat, konkrétan számsorozat) objektumtípust jelzi, a másik egy függvény, amivel range típusú objektumok, azaz számsorozatok állíthatók elő. Leggyakrabban for akármilyen in range(100) jellegű utasításban használjuk, a számlálós ciklus szimulálására.
- [read\(\)](#) – nem önálló parancs, hanem a fájl típus tagfüggvénye. Fájl tartalmának egyszerre, vagy pár karakterenként történő beolvasására való. Magyarul: olvas.
- [readline\(\)](#) – nem önálló parancs, hanem a fájl típus tagfüggvénye. Szövegfájl egy sorának beolvasására való. Magyarul: olvas sort.
- [readlines\(\)](#) – nem önálló parancs, hanem a fájl típus tagfüggvénye. Szövegfájl sorait olvashatod be egy listába, a lista egyes elemei a szövegfájl sorai lesznek. Magyarul: olvas sortokat.
- [remove\(\)](#) – nem önálló parancs, hanem a lista adattípus tagfüggvénye. Magyarul: eltávolít. A zárójelben megadott elem első előfordulását kiveszi a listából.
- [reversed\(\)](#) – a zárójelben megadott listát vagy karakterláncot megfordítva adja vissza. Magyarul: megfordítva, megfordítottan.
- [round\(szám, tizedesjegy\)](#) – tizedesjegy pontosságra kerekíti a megadott számot. Magyarul: kerekít.
- [return](#) – függvény visszatérési értékét lehet vele megadni. Magyarul: visszatér.
- [time.sleep\(\)](#) – a zárójelben megadott másodpercet vár a programod. Magyarul: alszik.
- [set\(\)](#) – halmaz létrehozására szolgál. Magyarul: halmaz.
- [sort\(\)](#) – nem önálló parancs, hanem a lista adattípus tagfüggvénye, és arra jó, hogy a listát helyben rendezze – azaz az a lista, amelyikre meghívod, rendezett lesz. Magyarul: rendez. Paraméterezését lásd a sorted() függvénynél.
- [sorted\(\)](#) – a zárójelben megadott listát rendezi, és új listát ad vissza, az eredetit nem bántja. A reverse=True megadásával lehetőség van csökkenő sorrendet kérni, és kulcsfüggvényt is adhatsz meg, amivel úgy rendezel, ahogy jól esik – részletekért lásd a rendezős postot. Magyarul: rendezett, rendezve.
- [split\(\)](#) – nem önálló parancs, hanem a karakterlánc adattípus tagfüggvénye, és arra jó, hogy a zárójelben megadott határolókarakternél szétszedi a karakterláncot listává. Ha nem adunk meg semmit, akkor a whitespace karaktereknél hasít. Magyarul: hasít.
- [str, str\(\)](#) – Az első szó a string típust jelzi. A string szónak jó sok jelentése van, például madzagot is jelent, de nekünk jobb a fűzér, informatikai szövegekben pedig karakterláncnak fordítandó. A legtöbb esetben jó, ha stringet olvasol, és szöveget gondolsz. A második szó egy függvény, ami szöveggé, karakterláncná alakítja azt, ami a zárójelben van, ha lehet.
- [strip\(\)](#) – nem önálló parancs, hanem a karakterlánc adattípus tagfüggvénye. A karakterlánc elejéről és végéről lepusztítja a zárójelben megadott karaktereket. Ha nem adunk meg semmit, akkor a whitespace-karaktereket takarítja le, azaz a szóközt, tabulátort, soremelést, kocsivisszát. Magyarul: megfoszt, lemeztelenít. A whitespace magyarul fehér helyet jelent, de soha nem szoktuk lefordítani.
- [sum\(\)](#) – A zárójelben egy bejárható objektumot, például listát, vagy range-t adunk meg. A sum() függvény megadja a bejárható objektumban lévő számok összegét. Magyarul: összeg.
- [True](#) – a bool adattípus egyik lehetséges értéke. Magyarul: igaz. Mindig nagy betűvel kezdjük.
- [while](#) – a róla elnevezett while-ciklus első szava. Addig ismétlődik a ciklus, amíg a while után megfogalmazott feltétel teljesül. Magyarul: amíg.