

# Informatika 1

## 6. előadás

Kovács Kristóf

Budapesti Műszaki Egyetem

2013. október 16.

# Listák függvényei

- `.append(<item>)`
  - hozzáfüzi az elemet a lista végére
- `.remove(<item>)`
  - törli egy előfordulását az adott elemnek
- `.clear()`
  - üríti a listát (más adatszerkezeteknél is működik)
- `.index(<item> [, start [, stop]])`
  - megkeresi, hogy a megadott elem melyik indexen található, a start és stop paraméter opcionális, meg lehet adni velük mely indexek között keresse
- `.count(<item>)`
  - megszámolja, hogy az adott elem hányszor szerepel a listába
- `len(<list>)`
  - visszaadja a lista elemeinek a számát
- `.sort()`, `sorted(<list>)`
  - `.sort` rendezi a listát, a másik a lista rendezettjével tér vissza de az eredeti listát nem változtatja

# Halmazok függvényei

- `union(<set>, <set>)`
  - a két halmaz úniójával tér vissza
- `.intersection(<set>)`
  - a két halmaz metszetével tér vissza
- `.difference(<set>)`
  - a két halmaz különbségével tér vissza
- `.discard(<item>)`, `.remove(<item>)`
  - törli az adott elemet, a discardot nem zavarja, ha nincs mit törölni
- `.isdisjoint(<set>)`
  - igaz, ha nincs közös elemük
- `.issubset(<set>)`
  - igaz, ha részhalmaza a paraméter
- `.issuperset(<set>)`
  - igaz, ha részhalmaza a paraméternek

# Szótárak függvényei

- `dict[<key>]`
  - érték lekérése, megváltoztatása és új kulcs-érték hozzáadása
- `.get(<key>)`
  - a kulcshoz tartozó elemmel tér vissza, ha nem létezik, akkor `None`-al tér vissza
- `.has_key(<key>)`
  - igaz, ha van ilyen kulcsú eleme a szótárnak
- `.items()`
  - a szótár elemei egy listában, kulcs-érték tuple-ben
- `.keys()`
  - a szótár kulcsai egy listában
- `.values()`
  - a szótár értékei egy listában

# Függvények szimbolikus számításokhoz

- `var(<string>)`
  - szimbolikus változóként definiálja az adott változót
- `.subs(<args>)`
  - egy kifejezésbe behelyettesíti a megadott értékeket és visszaadja az eredményt
- `log(<number>)`
  - természetes alapú logaritmus
- `sqrt(<number>)`
  - négyzetgyök
- `sin(<number>)`, `cos(<number>)`, `tan(<number>)`, ...
  - szögfüggvények
- `arcsin(<number>)`, `arccos(<number>)`, `arctan(<number>)`, ...
  - a szögfüggvények inverzei
- `floor(<number>)`
  - alsó egészrész
- `ceil(<number>)`
  - felső egészrész

# Kifejezéseken használt függvények

- `expand(<expr>)`
  - kibont egy kifejezést (elvégzi a zárójelfelbontást)
- `.factor()`
  - megkísérli szorzattá alakítani a kifejezést
- `simplify()`
  - egyszerűsít egy kifejezést
- `solve(<expr>, <var>)`, `.solve(<var>)`
  - megold egy egyenletet (egyenletrendszert több kifejezés és változó esetén), szimbolikusan számol, átalakításokkal
- `roots(<expr>, <var>)`, `.roots(<var>)`
  - numerikusan határozza meg a kifejezés gyökeit, multiplicitással
- `find_root(<expr>, <number>, <number>)`
  - a kifejezés egy gyökét keresi az adott intervallumban
- `.limit(<var> = <number>)`
  - a kifejezés határértéke az adott pontban
- `.derivative(<var>)`
  - a kifejezés deriváltja az adott változó szerint

# Függvényábrázolás sage-ben

- `plot(<expr>, (<number>, <number>), [options])`
  - kirajzolja az adott függvényt, az adott intervallumon, az adott opciókkal
- opciók lehetnek pl:
  - `aspect_ratio=1`
    - megtartja a tengelyek arányát
  - `rgbcolor='red'`
    - piros színnel rajzolja ki a függvényt
- `show(<plot> + <plot> [+ <plot> ...])`
  - kirajzolja két vagy több függvény grafikonját közös koordinátarendszerben