Tartalomjegyzék

Szimbolikus kifejezések

Ezen a gyakon szimbolikus kifejezésekkel foglalkozunk, így ne felejtsétek el a változókat, amik szimbolikusak, annak definiálni:

```
x = var('x')
```

1. Szimbolikus bevezetés

- Az *is_prime()* függvénnyel határozd meg, hogy a 2013 * 2014 - 1 prím-e!
- Egészítsd ki a kódot, hogy mûködjön!

```
a = <!>
<!> = a
b.factor()
```

- Határozd meg 2011 * 2012 + 1 gyökét az *sqrt()* függvénnyel!
- Egészítsd ki a kódot, hogy megoldja az egyenletet!

```
<!> = var('x')
<!>(2 * x ** 2 - 9 * x - 56 == 0, <!>)
```

- Az egyenlet megoldását add értékül egy változónak.
- Majd ennek a változónak (a megoldásnak) kérd le az elsõ elemét (mintha lista lenne, mivel az).
- Végül a megoldás jobb oldalát kérd le a *right()* metódussal.

- Oldd meg a *sin*(*x*) + *log*(*x*) - *pi* = 0 egyenletet a *solve()*-al, miután ez nem sikerült, oldd meg a *find_root()*-al (0 és 100 között van egy megoldás)!
- Legyen az *f* függvény a (2*x* + 5*y*)^3 ! (Ne felejtsd el felvenni y-t is mint szimbolikus változót.)
- Helyettesíts f-be a *subs()* függvénnyel, x = 316, y = 276-ot!
- Egészítsd ki a kódot, hogy összegre bontsa a kifejezést!

```
(a, b) = <!>
((2 * a - b) ** 3).<!>
```

2. Másodfokú egyenlet

- Oldd meg az általános 2. fokú egyenletet a *solve()* segítségével.
- Kérd le ennek az egyik (általános) megoldását.
- Majd helyettesítsd be az 5, 3, 2 értékeket az együtthatók helyére (a tagok fokával csökkenõ sorrendben).

3. Rajzolás bevezetés

- Egészítsd ki a kódot, hogy cosinus görbét rajzoljon ki 0-tól 4*pi-ig.

```
plot(<!>, (0, <!>))
```

- Rajzold ki az (x-2)^2 + 3 másodfokú polinomot -2-t?l 4-ig, zöld színnel!
- Rajzoljunk kört: cirlce((középpont koordinátái), sugár, egyebek). Az "http://wiki.math.bme.huegyebek"http://wiki.math.bme.hu lehetnek: szín, aspect_ratio=True hogy az x és y tengelyek skálázása azonos legyen (különben ellipszist kaphatunk!).
- Rajzold a másodfokú polinomot és a kört egymás mellé a show függvénnyel.

- Rajzoljátok ki a (*cos(x)*)^(0.5) + (*log(x)*)^2 - *pi* függvényt a (0, 20) intervallumon, majd vegyétek észre, hogy mennyire fura.
- Rajzoljátok ki a (*cos(x)*)^(-0.5) + (*log(x)*)^2 - *pi* függvényt a (0, 20) intervallumon, majd vegyétek észre, mennyire gyönyörû.
- Nincsnek határok, találjatok szebb függvényt!

- Azért közbe vegyétek észre, hogy a *cos(x)*^(0.5) miatt a függvény valósak felett nem mindenhol értelmezett, ezért lettek ilyen szépek a függvények.
- Javítsátok ki õket, hogy mindenhol értelmezettek legyenek és mégse legyenek annyir megváltoztatva. Majd vegyétek észre így mennyire unalmasak lettek.

4. Több megoldás

- A korábbi *sin*(*x*) + *log*(*x*) - *pi* = 0 egyenletnek keresd meg 3 gyökét a [0, 20] intervallumon. Esetleg ha segít rajzold ki elõtte plot-al.
- Írj (sage) függvényt ami 3 bemenetet kap: egy egyenlet (f), intervallum (I) és gyökök száma (n). A függvény keresse meg az **f** legalább **n** megoldását az **I** intervallumon. Egy kis segítség:
    - A korábbi gyakorlaton négyzetgyököt számoltunk intervallum felezéssel. Itt is lehet intervallum felezést alkalmazni.
    - Nem kell foglalkozni azzal, hogy egy megoldás pont az intervallum szélére esik, hisz a *find_root()* numerikus hibával adja meg a megoldást, így ez elég valószínûtlen.
    - A megoldásokat rakja egy listába, majd ezt adja vissza.

- Próbáljátok ki a *sin*(*x*) + *log*(*x*) - *pi* = 0 egyenlet, [0, 20] intervallum és 3 bemenetekkel.
- Majd próbáljátok még ki a (x - 1) * (x + 2) * (x - 6) függvényen, hogy biztosan jól mûködik-e.