

Informatika 1 – 2011

Első előadás

Szabó Adrienn

2011. szeptember 7.

Tartalom

A tárgyról

Sage alapok

A Sage mint számológép

Függvények a Sage-ben

Függvényábrázolás, rajzolás

Python kitekintés

Az Informatika 1 tárgy tematikája

- ▶ Sage (és Python) – 6 előadás
- ▶ XHTML, CSS – 2 előadás
- ▶ \LaTeX – 3 előadás
- ▶ Táblázatkezelés (Excel) – 1 előadás

Sage bevezető

Miért kellene segédprogramok a matematikához?

- ▶ számolási hibák
- ▶ gyorsabb
- ▶ négyszín-tétel

Miért a Sage?

- ▶ mert nyílt forráskódú
- ▶ mert ingyenes
- ▶ mert könnyen kiterjeszhető, programozható (Python-ra épül)

Sage – mint számológép

▶ sage: 3 + 5 + 8
16

▶ sage: 3 - 8
-5

▶ sage: 2 * 9
18

▶ sage: 6 / 3
2

▶ sage: 8 / 3
8/3

▶ sage: 2 ** 3
8

▶ sage: 3^3
27

▶ sage: 27 // 5
5

▶ sage: 27 % 5
2

▶ sage: 8.0 / 3
2.666666666666667

Sage – mint okos számológép

Változókat definiálhatunk és műveleteket hajthatunk végre velük:

```
▶ sage: a = 8 / 3
sage: b = 1 / 3
sage: a + b
```

Beépített változókat is használhatunk:

```
▶ sage: e + pi
```

Sage – mint okos számológép

Változókat definiálhatunk és műveleteket hajthatunk végre velük:

```
▶ sage: a = 8 / 3
sage: b = 1 / 3
sage: a + b
3
```

Beépített változókat is használhatunk:

```
▶ sage: e + pi
```

Sage – mint okos számológép

Változókat definiálhatunk és műveleteket hajthatunk végre velük:

```
▶ sage: a = 8 / 3
sage: b = 1 / 3
sage: a + b
3
```

Beépített változókat is használhatunk:

```
▶ sage: e + pi
```


Sage – mint okos számológép

Változókat definiálhatunk és műveleteket hajthatunk végre velük:

```
▶ sage: a = 8 / 3
sage: b = 1 / 3
sage: a + b
3
```

Beépített változókat is használhatunk:

```
▶ sage: e + pi
pi + e
```

Sage – mint szimbolikus számológép

Ha szimbólumként akarjuk, használni valamelyik változót, akkor ezt a `var()` függvénnyel jelezhetjük a Sage programnak. Egyszerre többet is definiálhatunk:

```
▶ sage: (a,b)=var('a,b')
```

Az algebrai kifejezéseket összeggé alakíthatjuk az `expand()` függvénnyel vagy `.expand()` metódussal, ill. szorzattá a `factor()`-ral:

```
▶ sage: expand((a+b)^2)
a^2 + 2*a*b + b^2
```

```
▶ sage: 187.factor()
11 * 17
```

Sage – mint szimbolikus számológép 2

Egyenleteket, egyenletrendszereket is megold nekünk a Sage:

- ▶ sage: `x = var('x')`
sage: `solve(x^2 + 3*x + 2, x)`
`[x == -2, x == -1]`
- ▶ sage: `x, y = var('x, y')`
sage: `solve([x+y==6, x-y==4], x, y)`
`[x == 5, y == 1]`

Sage – mint szimbolikus számológép 3

Néha a `solve()` nem találja meg a keresett megoldást:

```
▶ sage: x = var('x')
sage: solve(cos(x) == sin(x), x)
[sin(x) == cos(x)]
```

Ekkor a `find_root()` segíthet egy numerikus megoldást találni egy megadott intervallumon (legyen pl: $0 < y < \pi/2$):

```
▶ sage: y = var('y')
sage: find_root(cos(y) == sin(y), 0, pi/2)
0.78539816339744839
```

Sage – relációk, Boole algebra

▶ sage: 3 < 5
True

▶ sage: 4 == 4
True

▶ sage: 3+4 <= 6
False

▶ sage: a > b
a > b

▶ sage: a=5; b=7
sage: a > b
False

▶ sage: a !=b
True

▶ sage: True and True
True

▶ sage: True and False
False

▶ sage: True or False
True

▶ sage: not False
True

Sage – függvények

Gyakran használt matematikai függvények:

- ▶ `sqrt()`, `cos()`, `sin()`, `tan()`, `cot()`
- ▶ `is_prime()`, `factor()`, `expand()`, `solve()`,
`simplify()`
- ▶ `plot()`, `parametric_plot()`
- ▶ és sok van még ...

Sage – függvények 2

Mi is definiálhatunk új függvényeket, a `def` kulcsszóval (ami a Python nyelv része):

- ▶ sage:

```
def paros(x):  
...     return x%2 == 0
```
- ▶ sage:

```
paros(9)  
False
```
- ▶ sage:

```
paros(22)  
True
```

Sage – plot

A `plot()` függvénnyel ábrákat készíthetünk. Általában függvény-ábrázolásra használjuk, de egyébeket is rajzolhatunk (pl. köröket, sokszögeket).

A `plot` első paramétere az ábrázolandó dolog, utána még sok paraméterrel módosíthatjuk a képet (intervallum, színek, stb).

- ▶ `sage: plot(x^2, (-3,3), aspect_ratio=1)`
- ▶ `sage: plot(sin(x), (-pi, 2*pi),
rgbcolor='red')`

Sage – plot 2

Akár több függvényt is ábrázolhatunk közös koordinátarendszerben, ehhez mentsük el az egyes grafikus objektumokat egy-egy változóba, majd használjuk a `show` metódust:

- ▶ `sage: g1 = plot(sin, (-2*pi, 2*pi))`
- ▶ `sage: g2 = plot(cos, (-2*pi, 2*pi),
 rgbcolor='red')`
- ▶ `sage: show(g1 + g2)`

Python – Sage különbségek

- ▶ Hatványozás műveleti jele Python-ban `**`, a `^` mást jelent!

```
>>> 3^3
```

```
0
```

```
>>> 1^3
```

```
2
```

- ▶ Osztás (`/`) operátor Pythonban egész-osztást végez.

```
>>> 3/5
```

```
0
```