

Handouts/Lecture 4 (Calculus and graphics)/Lecture 4: Interactive Functions/Interact.sagews

Author Eugene Strakhov

Date 2019-06-29T19:08:12

Project 07c06dbe-4967-451f-aa68-dd9268bd2ece

Location [Handouts/Lecture 4 \(Calculus and graphics\)/Lecture 4: Interactive Functions/Interact.sagews](#)

Original file [Interact.sagews](#)

Интерактивные функции

Интерактивные функции

1

- [Документация](#)
- [Примеры](#)
- [SageWiki](#) (RUS)

2

С помощью декоратора `@interact` можно создавать интерактивные функции. Фактически они представляют собой формы с элементами управления, позволяющими интерактивно и наглядно менять параметры задачи.

С помощью декоратора `@interact` можно создавать интерактивные функции. Фактически они представляют собой формы с элементами управления, позволяющими интерактивно и наглядно менять параметры задачи.

```
3 # Интерактивное решение уравнений
4 @interact
5 def any_name(eq = input_box(x^2 == 0, 'Задайте уравнение:', SR)) :
6     assume(x, 'real')
7     sol = eq.solve(x)
8     if sol == [] :
9         print('Действительных корней нет.')
10    else :
11        print('Действительные корни уравнения:')
12        for s in sol :
13            print(s)
```

Задайте уравнение:

 Submit

(Evaluate this cell to use this interact.)

```
14 # Добавим функциональности
15 @interact
16 def _(eq = input_box(x^2 == 0, 'Equation to solve', SR),\
17     dom = selector(['Integer', 'Rational', 'Real', 'Complex'], buttons = True, label = 'Domain'),\
18     lf = checkbox(False, 'Use LaTeX-form')) :
19     sol = eq.solve(x)
20     if lf == True :
21         show(str(dom)+' solutions:')
22     else :
23         print(str(dom)+' solutions:')
24     if dom == 'Integer' :
25         dom = ZZ
26     elif dom == 'Rational' :
27         dom = QQ
28     elif dom == 'Real' :
29         dom = RR
30     else :
31         dom = CC
32     for s in sol :
33         if s.right() in dom :
34             if lf == True :
35                 show(s)
36             else :
37                 print(s)
```

Equation to solve

 Submit

Domain
IntegerRationalRealComplex
Use LaTeX-form
☐
(Evaluate this cell to use this interact.)

```

38 # Управление размещением элементов
39 @interact(layout = [['eq', 'dom'], ['lf']])
40 def _(eq = input_box(x^2 == 0, 'Equation to solve', SR),\
41       dom = selector(['Integer', 'Rational', 'Real', 'Complex'], buttons = True, label = 'Domain'),\
42       lf = checkbox(False, 'Use LaTeX-form')) :
43     sol = eq.solve(x)
44     if lf == True :
45         show(str(dom)+' solutions:')
46     else :
47         print(str(dom)+' solutions:')
48     if dom == 'Integer' :
49         dom = ZZ
50     elif dom == 'Rational' :
51         dom = QQ
52     elif dom == 'Real' :
53         dom = RR
54     else :
55         dom = CC
56     for s in sol :
57         if s.right() in dom :
58             if lf == True :
59                 show(s)
60             else :
61                 print(s)

```

Equation to solve

Domain
IntegerRationalRealComplex
Use LaTeX-form
☐
(Evaluate this cell to use this interact.)

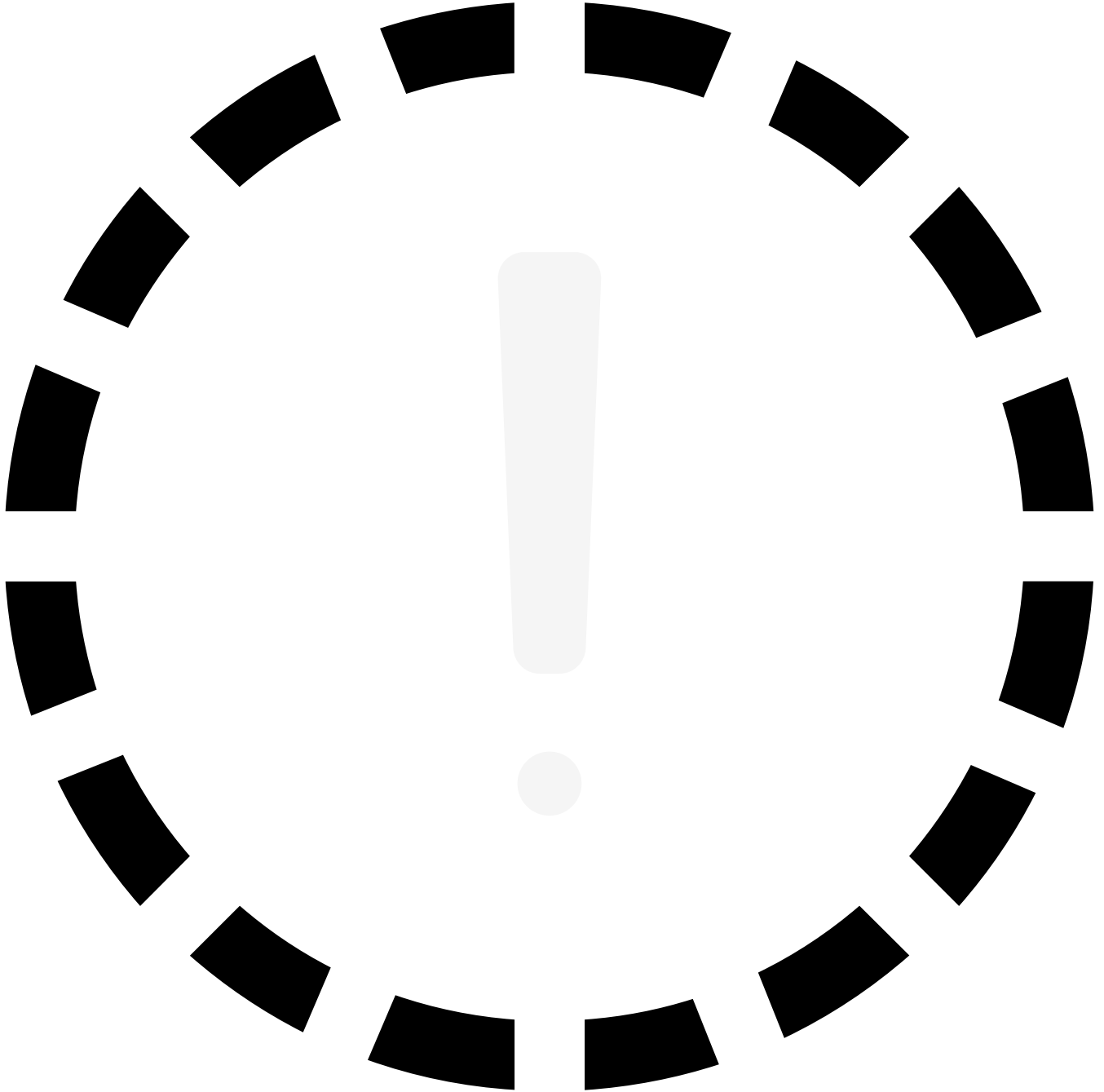
```

62 # Матрицы и векторы
63 @interact(layout = [['A', 'b']])
64 def ____(A = input_grid(3, 3, default = [[1, 0, 0], [0, 1, 0], [0, 0, 1]], label = 'Matrix'),\
65         b = input_grid(3, 1, default = [[1, 2, 3]], label = 'Right-hand side', to_value = lambda x : vector(flatten(x)))) :
66     A = matrix(SR, A)
67     b = vector(SR, b)
68     if A.rank() == A.augment(b).rank() :
69         show('Solution:', A.solve_right(b))
70     else :
71         show('System is inconsistent.')

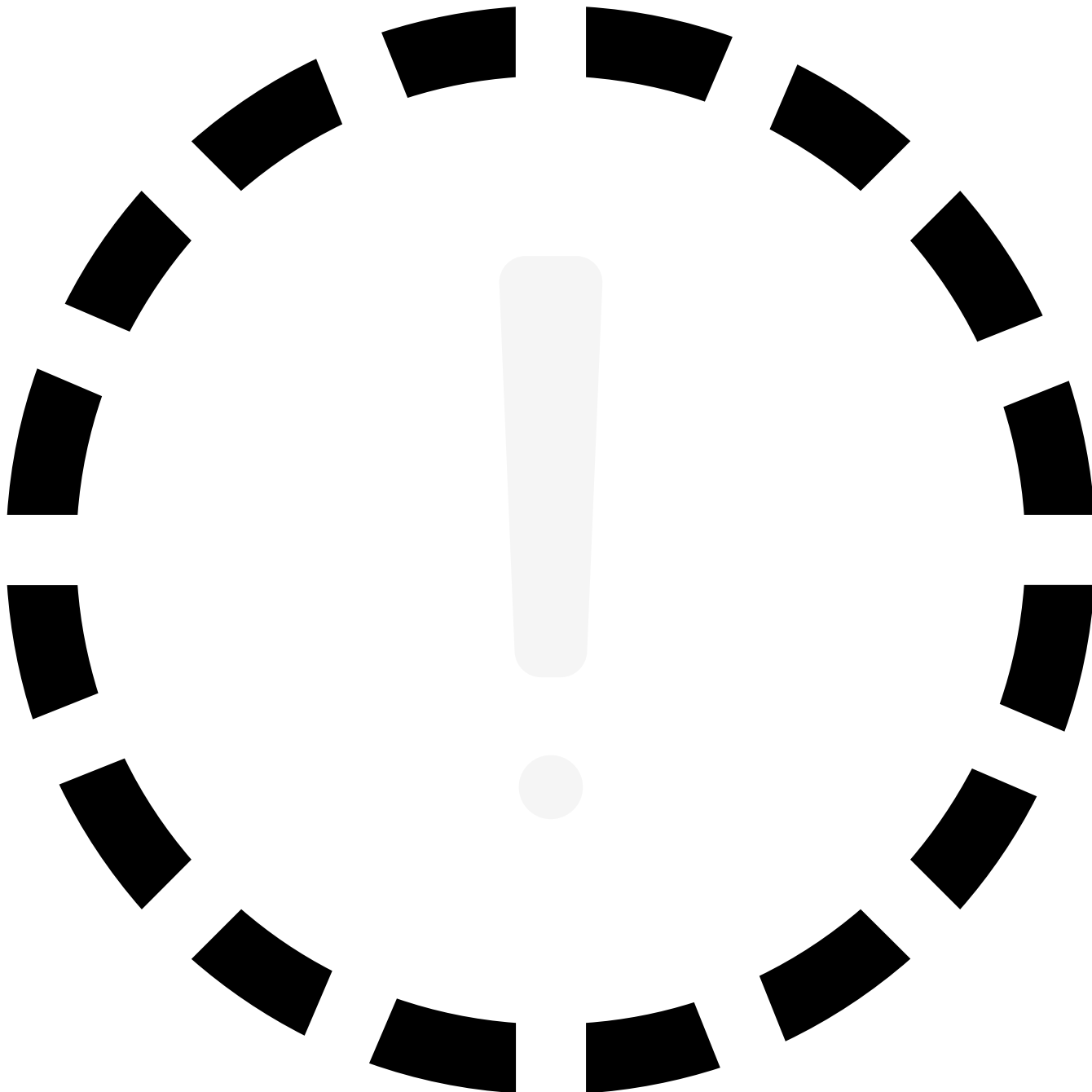
```

Matrix

--	--	--



Right-hand side



(Evaluate this cell to use this interact.)

```
72 # Простой интерактивный график
73 @interact
74 def _(f = x^2) :
75     p = plot(f, (x, -2, 2))
76     p.show()
```

f

Submit

(Evaluate this cell to use this interact.)

```
77 # График с выбором некоторых опций
78 @interact(layout = [['f', 'dx'], ['color', 'thickness', 'marker'], ['axes', 'frame', 'aspect_ratio']])
79 def _(f = input_box(x^2, 'Function to plot:'), SR, \
80     dx = range_slider(-20, 20, default = (-10, 10), label = 'Set range:'), \
81     color = selector(colors.keys(), label='Select color:'), \
82     axes = True, \
83     frame = False, \
84     thickness = 1, \
85     marker = '', \
86     aspect_ratio = slider(1/50, 1, step_size = 1/50, default = 1)) :
87     p = plot(f, (x, dx[0], dx[1]), color = color, \
88             axes = axes, frame = frame, axes_labels = ['$x$', '$y$'], thickness = thickness, marker = marker, aspect_ratio = aspect_ratio)
89     p.show()
```

Function to plot:

Submit

Set range:

-10.0 - 10.0

Select color:

indigo

▼

thickness

Submit

marker

Submit

axes



frame



aspect_ratio

1

(Evaluate this cell to use this interact.)