

GeoGebra 脚本应用入门

(哈尔滨 唐家军)

【按语】和多数软件一样，GeoGebra 也是对象型操作软件，但它还为应用者提供了更深层的操作，也就是可编程操作。GeoGebra 支持的编程语句被称为“脚本”。GeoGebra 支持两种脚本，一种是 Java 通用脚本，一种是 GeoGebra 专属脚本，当然 Java 脚本应用更广泛。本文只是对 GeoGebra 官网中关于 GeoGebra 的专属脚本应用入门进行了翻译，没有拓展，只是增加了案例截图，属于真正“入门”级读物。更多的脚本使用，看用户自己需要和编程能力了。官网帮助是针对 4.0 版本，在官网基础上，另外附上 GeoGebra5.0.134.0 的脚本指令和一个综合实例。本文适于会 GeoGebra 基本操作的人士。

教程：GeoGebra 脚本应用简介

有时，学生使用 GeoGebra 提供的基础功能与你互动任务单不是很给力，尤其是年轻学生更是感觉如此。GeoGebra 脚本为创建更复杂但易用的交互构件助你一臂之力。在 GeoGebra 中，写脚本（小程序）是轻而易举的事情。

目 录

目 录	1
1 背景：指令	2
2 脚本基础	2
3 按钮	3
4 输入框	3
4.1 链接输入框	4
4.2 携带点击时脚本的输入框	4
5 后续	4
6 脚本案例	5
6.1 按钮：“递增/递减”按钮	5
附一，脚本指令	5
附二，多页面案例	30

脚本的通常用法：

- 1、校验学生在文本域输入的回答并使用脚本核查是否正确。
 - 2、学生按一个按钮后创建一个新任务或挑战，其要么自动生成，要么使用你定义的挑战集。
 - 3、学生按一个按钮或完成任务后，开启一个动画或触发新的内容。
- (等等..)

GeoGebra 中有两种脚本语言，GeoGebra 脚本和 Java 脚本。因为使用 GeoGebra 指令和语法，且 GeoGebra 使用者更容易理解和学习，故本文只关注 GeoGebra 脚本。这两个功能相当的脚本语言有些许差异，但 GeoGebra 脚本在大多数情况下已足够给力。

注:脚本是 GeoGebra 经验用户的话题，在进入这个话题前，如果对 GeoGebra 不很熟悉，我们推荐你马上看一下其他教程。（当然，那里会有脚本基本工作原理和如何在你的构件中加入脚本。）

1 背景：指令

GeoGebra 不仅可以工具创建对象，还可以通过指令和指令栏实现创建对象。

案例:利用指令“圆形 $[(1, 2), 3]$ ”会得到一个圆心在 $(1, 2)$ 半径为 3 的圆。

即便使用工具创建一个对象，实际上就是 GeoGebra 使用命令在后台创建对象。在对象“属性”对话框的“定义”下可看到创建对象的指令。所以，工具几乎就是使用鼠标帮助输入这些指令的。

到目前，我们只谈能创建对象的命令。多数指令能创建对象，但只有部分指令适用于脚本，稍后会谈及更多细节（急于求知者，参见“脚本指令”）。

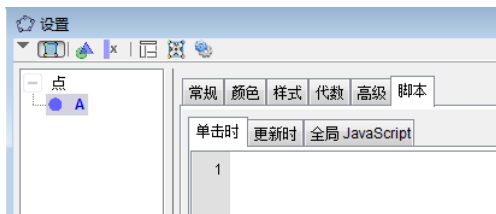
2 脚本基础

脚本在本质上就是 GeoGebra 指令程序，可以在以下两种情况执行：

- 单击时：单击一个对象后执行某个脚本。
- 更新时：对象的值或属性被改变后执行某个脚本。这将发生在使用者随处移动构件对象、改变滑杆值等时刻。

在 GeoGebra 中，脚本是对象中心式，这意味着你可为每个对象定义脚本，一种是“单击时”，一种是“更新时”。但没有脚本能脱离对象执行（Java 中有一个可脱离对象执行的脚本）。

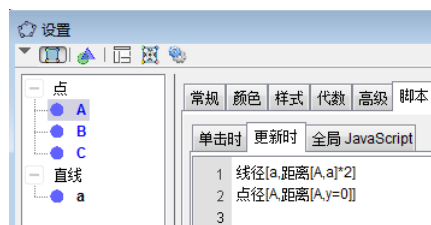
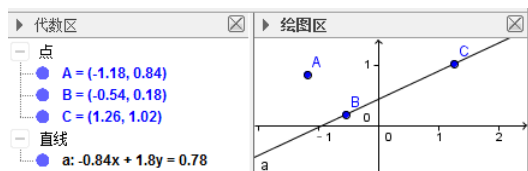
不论是“代数区”或“绘图区”，只要右键对象都可以为其添加脚本。点对象“属性”且选“脚本”表签。会出现三个表单，即“单击时”、“更新时”和“全局 JavaScript”。第三个不在本文讨论，其他两个可以为选择的对象添加对象“单击时”或“更新时”执行的脚本。



现在是时候实地创建一个脚本了，我们将创建一个使用“更新时”脚本的傻瓜案例：

案例:(更新脚本，你还可以在 GeoGebra 通道找到全部构件)

- 1、首先，打开一个新的 GeoGebra 窗口。
- 2、现在创建一个点“A”和一条直线“a”。如果你对在指令栏中创建对象不甚熟悉，试试“直线 $[B, C]$ ”指令（译注：需要先构造点 B 和 C）。



- 3、根据上文描述打开点 A 脚本面板的“更新时”表单（译注：右键点 A，属性，脚本，更新时）。
- 4、在脚本中输入指令“线径 $[a, \text{距离}[A, a] * 2]$ ”和“点径 $[A, \text{距离}[A, y=0]]$ ”（译注：如图分两行；可以复制本短文相关引号内的指令文本）

5、现在，回车确认。

因为是脚本，请点击“确定”按钮（极易遗忘！）后关闭对话框。

想要弄明白脚本做了什么，让点 A 随意移动验证你的假设。

译注：在绘图区创建点 A 和直线 a，在点 A 的脚本中使用以上格式的“线径”和“点径”指令，然后“确定”。因为线径和点径都关联了既有对象，故既有对象改变时，就执行脚本。注意，如果在“单击时”输入脚本，则只有点击点 A 时，脚本才执行。如果在“更新时”输入，拖动点 A 移动位置，就执行脚本。

如此，你注意到我们赋予点 A “更新时”执行脚本，而不是直线。这意味着脚本在移动点 A 的任意时刻都会被执行，而脚本不修改点本身的任何特性。

上文提及的“线径[]”和“点径[]”指令，我们已经用到。有时你需要至少略读“脚本指令列表”了解这些指令的可能应用，但不必局限于这种指令，你还可以在你的脚本中使用诸如“ $f(x)=x^2$ ”或“c=圆形[(0,0),5]”等。

除一些特别指令外，下边描述更多细节，都与执行一个在指令栏输入简单直线的 GeoGebra 脚本有相同的效果。现在，基本上知道如何在 GeoGebra 中写脚本了，本教程接下来会描述一些特别的脚本对象，如“输入框”或“按钮”。除了脚本指令外使用最多的是很可能是“条件函数[...]”，如你没有这些知识储备，应该先看看“逻辑”和“指令列表”。

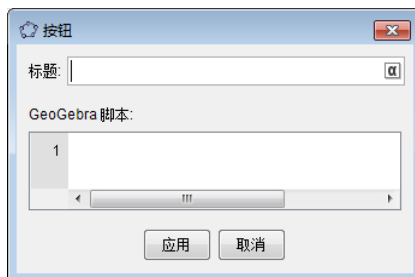
译注：某些“条件函数”用法：“条件函数[a>0, 中心定位[0,0]]”；“条件函数[a>0, 放大[1]]”；“条件函数[x(A)>0, 标签模式[A,1]]”；“开启动画[条件函数[a>0, true, false]]”；“显示对象[A, 条件函数[x(A)>0, true, false]]”；“显示网格[条件函数[a>0, true, false]]”；“标签模式[A, 条件函数[x(A)>0, 1, 2]]”；“跟踪[A, 条件函数[x(A)>0, true, false]]”。指令意义参见“附一”。

注：许多脚本指令只执行一次动作且不形成一个如你所熟悉的从属对象般的永久关联，故“线径[a, 距离[A, a]*2]”只复制执行脚本时的“距离[A, a]*2”当前值，然后，其值不再变化直到你再次执行指令（改变关联对象值而执行脚本）或人工改变线径。本案例制造了一种关联的效果，但脚本只是在移动点 A 的时刻被执行。你可以看看移动线的替代效果，线径不会因此改变除非重新移动点。

注：如果你是其他程序语言的玩家，会注意到在 GeoGebra 脚本中不需要在每行结尾使用分号。

3 按钮

因为学生知道且会尝试点击，故按钮很重要。无特别之处，点击按钮执行脚本和为其他对象添加“单击时”执行脚本效果相同。你可以使用[Ⓚ]“按钮”工具创建一个按钮。



4 输入框

输入框(也称文本域)，可参见^{a=1}“输入框”工具，其也许是 GeoGebra 中最给力的对象。它允许学生输入文本、数值、函数和任意其它你能（手动）由文本转换为 GeoGebra 对象的内容。

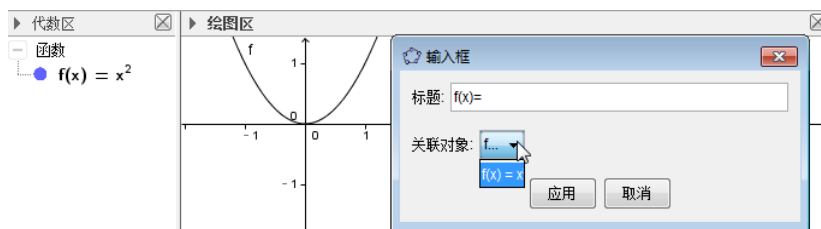
在你的构件中有多种应用输入框的方法，你既可使得输入框与对象链接，还可关联一个“单击时”脚本或稍后使用另一个对象处理（如一个按钮）。别急，所有细节马上揭晓。

4.1 链接输入框

“链接输入框”是链接到现有 GeoGebra 对象的输入框，这意味着输入框显示当前的对象值，且如果你在输入框中改变值，原件对象会随着改变它的值。你可以利用这个让你的学生不使用滑竿也能改变函数定义或数值。使得他们仅能在改变简单对象值的框内输入，从而容易操作且没有破坏学生任务单（译注：开发者默认软件是学生学习用的，故总有学生与软件学习任务单的互动）的危险。

案例:(链接输入框，本构件案例可以在 GeoGebra 通道再次找到)

- 1、打开空白 GeoGebra 窗口。
- 2、在指令栏输入一个函数，如“ $f(x)=x^2$ ”。
- 3、在工具栏选 $\text{f}=\square$ “输入框”工具(右边第二个工具组)。
- 4、现在，在你的工作区的某处点击，会弹出一个对话框。在“标题”处输入“ $f(x)=$ ”且“关联对象”选你的函数 f 。



- 5、完成（译注：点“应用”）。移动函数图象观察在文本域中如何自动显示此函数新定义。你也可以修改文本域且按回车观察如何利用文本域重新定义函数。

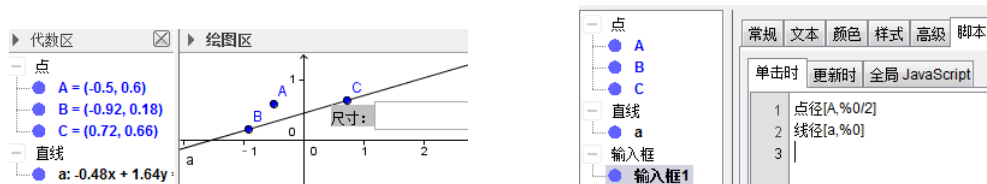
注:不仅上边提及的函数可用，甚至数值和几乎全部对象都可以链接，试试看。

4.2 携带点击时脚本的输入框

与上一个例子形成对比，我们现在使用“单击时”脚本把它的值应用于其他对象而不把输入框链接到一个特定对象。脚本与上一个案例很相似，唯一新问题是脚本中如何引用文本域的值。其答案就是“%0”。

案例:(携带点击脚本的输入框，参见 [GeoGebra 通道](#))

- 1、新建一个 GeoGebra 窗口
- 2、创建点 A 和直线 a。
- 3、选择“输入框”工具点击绘图区（同上），此次标题使用“尺寸:”且让“关联对象”域空着（译注：点击“应用”后退出）。



- 4、跳转到新输入框的“单击时”脚本表单(上文有如何做的说明)。
- 5、现在加入如下脚本(注意%0)：点径[A,%0/2]、线径[a,%0]。
- 6、再次点击“确定”关闭对话框。
- 7、现在，在“输入框”输入一个新的值并按回车。如此你会发现占位符“%0”真实地引用了输入框的值。

5 后续

如果你对脚本了解一些（一点儿也行）：十分欢迎你的帮助和分享！你可以为以下主题添加内容或更多想法：

- 给初学者的手把手指导。
- 提供更多具体案例(同时把使用了脚本的案例上传 GeoGebra 通道)。

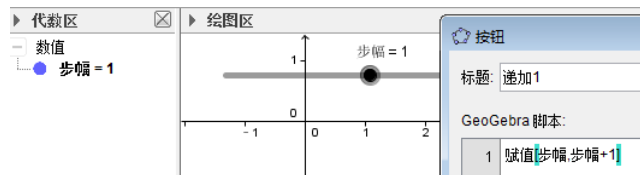
6 脚本案例

6.1 按钮：“递增/递减”按钮

你可以使用一个按钮取代拖动，实现一个滑杆参数逐步递改。

案例：

- 1、创建一个整数滑杆，名称为“步幅”。
- 2、使用“按钮”工具创建一个按钮。
- 3、输入按钮标题（“递加 1”）作为显示标签。
- 4、输入脚本“赋值[步幅, 步幅+1]”或简化为“步幅=步幅+1”。（别忘记了点“应用”）



改“+”为“-”就实现递减。改变 1 为需要的值，改变递变量。

注：指令“步幅=步幅+1”只适用于自由数值。用到从属数值时会得到一个重复定义的错误。

附一，脚本指令

以下列出了在 GeoGebra 5.0.134.0 中可用的专属脚本指令，前边是英文名称，后边是中文名称。在中文界面下应用时，输入英文和中文都可以，但因为软件的中文翻译也在不断进步，当中文指令不奏效时，可以尝试输入英文指令。在英文界面下中文指令不可用。

1 Button. 按钮

Button[]; 按钮[].

创建一个新按钮。

Button[<Caption>]; 按钮[“<标题>”]

创建给定标题的按钮。

案例：“按钮[“开始”]”创建一个标题为“开始”的按钮。（至于此按钮执行的动作，需要在此按钮的“属性”中的“脚本”中设置。）



注：参见 “按钮”工具。

2 setBackgroundColor. 背景色

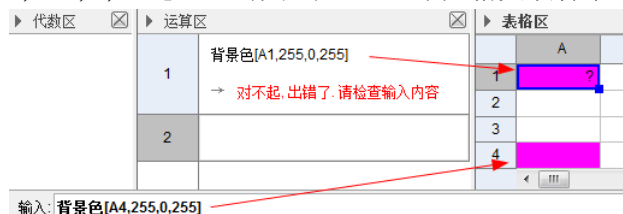
这个指令在不同的英语变型中拼写不同：SetColor (US)、SetBackgroundColour (U

K+Aus)。

SetBackgroundColor[<Object>, <Red>, <Green>, <Blue>]; 背景色[<对象>, <红色 0-1>, <绿色 0-1>, <蓝色 0-1>]。

更改指定对象的背景颜色，用于文本对象和表格区的对象。红、绿和蓝能表示大量的相关颜色组成，其值最小 0，最大 1。超过这个区间的数值 t ，使用函数 $2|t/2 - \text{round}(t/2)|$ 映射。

案例：“背景色[A1, 1, 255, 0, 255]”，工作表中“A1”单元格更改背景色。



常用的基本颜色

名称	红色值 Red	绿色值 Green	蓝色值 Blue
黑色	0	0	0
蓝色	0	0	255
绿色	0	255	0
青色	0	255	255
红色	255	0	0
洋红色	255	0	255
黄色	255	255	0
白色	255	255	255

SetBackgroundColor[<Object>, <"Color">]; 背景色[<对象>, <"颜色">]。

改变给定对象的背景颜色。这个用于文本和表格区对象。“颜色”键入文本。这个指令接受超过 100 个英语颜色名称（见参考：颜色）。部分颜色名称见下图，名称显示的颜色就是字体或者文字背景的颜色。

Black	Indigo	Light Gray
Dark Gray	Purple	Pink
Gray	Brown	Violet
Dark Blue	Orange	Yellow
Blue	Gold	Light Yellow
Dark Green	Lime	Light Orange
Green	Cyan	Light Violet
Maroon	Turquoise	Light Purple
Crimson	Light Blue	Light Green
Red	Aqua	White
Magenta	Silver	

3 SetLabelMode. 标签模式

`SetLabelMode[<Object>, <Number>]`; 标签模式[<对象>, <0_名称|1_名称+数值|2_数值|3_标题>]。

从下面表单中选给定对象的标签模式。与下列表中不同整数按照 0 对待。

数值	模式
0	名称
1	名称与数值
2	数值
3	标题

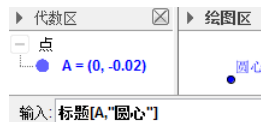
注：此命令也可以用窗口菜单实现：点选对象→“属性”→“常规”→“显示标签”右的下拉菜单中选择想要的模式。

4 SetCaption. 标题

`SetCaption[<Object>, <Text>]`; 标题[<对象>, “<标题文本>”]。

更改指定对象的标题，文本必须用一对英文的双引号“”括住。

案例：“标题[A, “圆心”]”。



注：标题在“标签模式”选择“标题”时才会显示。

5 PlaySound. 播放声音

`PlaySound[<Note>, <Duration>, <Instrument>]`; 播放声音[<音符>, <持续时间>, <乐器>]。

播放一个音符。

音符是在下表中列出代表音乐符号的 0 到 127 间的一个整数。当音符=60 时，演奏中音 C。持续时间是演奏音符的时间。

乐器是代表演奏音符合成乐器的一个整数。见可能乐器的技术说明。

多数乐器都支持，但在不同的电脑平台有差异。

MIDI 符 八度音阶	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	51	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107

9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

PlaySound[<Note Sequence>, <Instrument>]; 播放声音[<音符顺序>, <乐器>]。

播放由特定乐器演奏的一串音符。

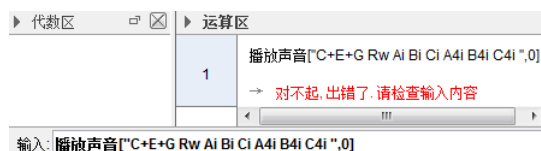
音符顺序是使用 JFugue 符指令组成的文本串。

乐器是播放音符串时使用的默认乐器。

基本指令见下表。全部指令设置在 JFugue 大全(英语)中有介绍。

基本 JFugue 指令符	指令
A-G	通过这些字符播放音乐;用 b 和#实现升调和降调; 增加一个数字改变八度, e. g. A4; 默认音符长度为 4。
[number]	通过数字 (0-127) 播放 MIDI 音符。
+	同时播放多个音符 e. g. C+E+G。
R	停止。
w, h, q, i, s	添加在最后或暂停时; 设置音符长度: w=全音符 h=半音符 q=四分音符 i=八分音符 s=十六分音符。
/n	添加到一个音符或休止; 集长度为 n (十进制数): /1=全音符/. 5=半音符/. 25=四分音符/2=twomeasures。
I[number]	设置乐器。
V	V 后跟一个数字从 0 到 15 台 MIDI 通道。V9 设置 MIDI 乐器通道。
Space	播放前面的音符、音符组合或暂停

案例:“播放声音[“C+E+G Rw Ai Bi Ci A4i B4i C4i”,0]”演奏四分之一节 CEG 合奏; 休止一节; 演奏八分之一节 A, B, C; 低八度再演奏一次。



使用钢琴乐器。

注: 尽管在运算区显示出错,但也有正常声音输出。

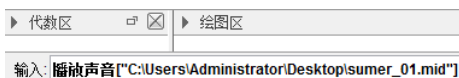
案例:“播放声音[“I[56] C5q D5q I[71] G5q F5q”,0]”使用不同乐器播放音符。喇叭声=56 和单簧管=71。

案例:“播放声音[“V0 A3q B3q C3q B3q V1 A2h C2h”,0]”不同声音合奏。

PlaySound[<File>]; 播放声音[<文件>]

播放一个电脑上的 mid 文件 (*.mid) 或包含 JFugue 串的文本文件 (*.txt)。

“文件”是完整路径文件,如:“播放声音[“C:\Users\Administrator\Desktop\sumer_01.mid”]”。



PlaySound[<Function>, <Min Value>, <Max Value>]; 播放声音[<函数>, <最小值>, <最大值>]。

通过范围在 [-1, 1] 的时间值函数播放一个声音。时间单位是秒且声音从最小时间到最大。声音采用 8 位采样变换率为每秒 8000 样本。

案例：“播放声音[tan(1/(sin(10 x^2))), 0, 10]”播放 10 秒貌似摩托车启动的声音。

PlaySound[<Function>, <Min Value>, <Max Value>, <Sample Rate>, <Sample Depth>]; 播放声音[<函数>, <最小值>, <最大值>, <采样率>, <样本深度>]。

通过范围在[-1, 1]的时间值函数播放一个声音。时间单位是秒且声音从最小时间到最大。采样方法由“样本深度”和“采样率”决定。

“采样率”是采样函数值每秒变化数值。允许数值为 8000、11025、16000、22050 或 44100。

“样本深度”每节样本数据大小。允许值为 8 和 16。

PlaySound[<Boolean Play>]; 播放声音[<是否播放? true|false>]。

暂停或重新播放。

播放声音[true]=播放, 播放[false]=暂停。

国际通用 GM 音色表

钢琴类 (Piano)

音色号	英文	英文简写	中文
1	AcousticGrandPiano	GrandPno	原声大钢琴
2	BrightGrandPiano	Britepno	亮音钢琴
3	ElectricGrandPiano	E. GrandPno	电子大钢琴
4	Honky-tonkPiano	HonkyTonk	酒吧钢琴
5	ElectricPiano1	E. Piano1	电钢琴 1
6	ElectricPiano2	E. Piano2	电钢琴 2
7	Harpsichord	Harpsi	拨弦古钢琴
8	Clavinet	Clavi	电子击弦古钢琴

半音性打击乐器 (ChromaticPercussion)

9	Celesta	---	钢片琴
10	Glockenspiel	Glocken	钟琴
11	MusicBox	---	八音盒
12	Vibraphone	Vibes	颤音琴
13	Marimba	---	马林巴
14	Xylophone	Xylophon	木琴
15	TubularBells	TubulBel	管钟
16	Dulcimer	---	扬琴

风琴类 (Organ)

17	HammondOrgan	HarhondOrgn	哈蒙德风琴
18	PereussiveOrgan	PercOrgn	击音管风琴
19	RockOrgan	RockOrgn	摇滚风琴
20	ChurchOrgan	ChurchOrgn	教堂管风琴
21	ReedOrgan	ReedOrgn	簧片风琴
22	Accordion	---	手风琴
23	Harmonica	---	口琴
24	TangoAccordion	TangoAcid	探戈手风琴

吉它类 (Guitar)

25	AcousticGuitar(nylon)	NylonGtr	尼龙弦吉它
----	-----------------------	----------	-------

26	AcousticGuitar(steel)	SteelGtr	钢弦吉它
27	ElectricGuitar(jazz)	JazzGtr	爵士吉它
28	ElectricGuitar(clean)	CleanGtr	纯音吉它
29	ElectricGuitar(muted)	Mute. Gtr	闷音吉它
30	OverdrivenGuitar	Overdriver	激励音吉它
31	DistortionGuitar	Dist. Gtr	失真吉它
32	GuitarHarmonics	GtrHarmo	吉它泛音

贝司类 (Bass)

33	AcousticBass	Aco. Bass	原声贝司
34	ElectricBass(finger)	FngeBass	指弹电贝司
35	ElectricBass(pick)	PickBass	拨片电贝司
36	FretlessBass	Fretless	无品贝司
37	StapBass1	---	打弦贝司 1
38	StapBass2	---	打弦贝司 2
39	SynthBass1	SynBass1	合成贝司 1
40	SynthBass2	SynBass2	合成贝司 2

弦乐器 (Strings)

41	Violin	---	小提琴
42	Viola	---	中提琴
43	Cello	---	大提琴
44	Contrabass	Contrabs	低音提琴
45	TremoloStrings	Trem. Str	弦乐震音
46	PizzicatoStrings	Pizz. Str	弦乐拨音
47	OrchestralHarp	Harp	竖琴
48	Timpani	---	定音鼓

合奏 (唱) 组 (Ensemble)

49	StringEnsemble1	Strings1	弦乐组 1
50	StringEnsemble2	Strings2	弦乐组 2
51	SynthStrings1	Syn. Str1	合成弦乐组 1
52	SynthStrings2	Syn. Str2	合成弦乐组 2
53	ChoirAahs	Choir. Aah	唱诗班啊声
54	VoiceOohs	VoiceOoh	哦声合唱
55	SynthVoice	Synvoice	合成人声
56	OrchestraHit	Orch. Hit	管弦乐齐奏

铜管乐器 (Brass)

57	Trumpet	---	小号
58	Trombone	---	长号
59	Tuba	---	大号
60	MutedTrumpet	Mute. Trp	小号减弱音器
61	FrenchHorn	Fr. Horn	法国号
62	BrassSection	BrasSect	铜管组

63	SynthBrass1	SynBras1	合成铜管 1
64	SynthBrass2	SynBras2	合成铜管 2

簧片乐器 (Reed)

65	SopranoSax	SprnoSax	高音萨克斯
66	AltoSax	---	中音萨克斯
67	TenorSax	---	次中音萨克斯
68	BaritoneSax	Bari. Sax	低音萨克斯
69	Oboe	---	双簧管
70	EnglishHorn	Eng. Horn	英国管
71	Bassoon	---	大管
72	Clarinet	---	单簧管

管鸣乐器 (Pipe)

73	Piccolo	---	短笛
74	Flute	---	长笛
75	Recorder	---	竖笛
76	PanFlute	---	牧笛
77	BottleBlow	Bottle	瓶笛
78	Shakuhachi	Shakhchi	尺八
79	Whistle	---	口哨
80	Ocarina	---	陶笛

合成领奏 (SynthLead)

89	Pad1 (newage)	NewagePd	背景 1 (新时代)
90	Pad2 (warm)	WarmPd	背景 2 (温暖的)
91	Pad3 (polysynth)	PolySyPd	背景 3 (复合合成)
92	Pad4 (choir)	ChoirPad	背景 4 (唱诗班)
93	Pad5 (bowed)	BowedPad	背景 5 (弓弦音色)
94	Pad6 (metallic)	MetalPad	背景 6 (金属般)
95	Pad7 (halo)	HaloPad	背景 7 (问候)
96	Pad8 (sweep)	SweepPad	背景 8 (宽阔的)

合成效果 (SynthEffects)

97	FX1 (rain)	Rain	效果 1 (下雨)
98	FX2 (soundtrack)	SoundTrk	效果 2 (音轨)
99	FX3 (crystal)	Crystal	效果 3 (晶体)
100	FX4 (atmosphere)	Atmosphr	效果 4 (气氛)
101	FX5 (brightness)	Bright	效果 5 (明亮)
102	FX6 (goblins)	Goblins	效果 6
103	FX7 (echoes)	Echo	效果 7 (回声)
104	FX8 (sci-fi)	Sci-Fi	效果 8

民间乐器 (Ethnic)

105	Sitar	---	西塔尔
106	Banjo	---	班卓
107	Samisen	---	三味线
108	Koto	---	日本筝
109	Kalimba	---	卡林巴
110	Bagpipe	---	风笛
111	Fiddle	---	小提琴
112	Shanai	---	山奈

打击乐 (Percussive)

113	TinkleBell	TnklBell	铃铛
114	Agogo	---	阿果果
115	SteelDrums	SteelDrm	钢鼓
116	Woodbiock	Woodblok	帮子
117	TaikoDrom	TaikoDrm	太叩鼓
118	MelodicTom	MelodTom	旋律性哏哏鼓
119	SynthDrum	Syn. Drm	合成鼓
120	ReverseCymbal	RevCymbal	反钹

音响效果 (SoundEffects)

121	GuitarFretNoise	FretNoiz	吉它滑品噪音
122	BreathNoise	Brthnoiz	呼吸声
123	Seashore	---	海浪声
124	BirdTweet	Tweet	鸟叫
125	TelephoneRing	Telephone	电话铃声
126	Helicopter	Helicptr	直升飞机声
127	Applause	---	掌声
128	Gunshot	---	枪声

6 Slider. 参数

Slider[<Min>, <Max>, <Increment>, <Speed>, <Width>, <Is Angle>, <Horizontal>, <Animating>, <Boolean Random>]; 参数 [<最小值>, <最大值>, <增量>, <速度>, <宽度(px)>, <角度?true|false>, <水平?true|false>, <启动动画? true|false>, <随机?true|false>]

创建一个滑杆参数 (变量)。指令中的参数可进行下列设置:

最小值, 最大值: 设置滑杆的取值范围—这些参数是强制性的。

增量: 设置滑杆值的增量—默认为: 0.1。

速度: 设置滑杆在动画时的速度—默认为: 1。

宽度: 设置滑杆的显示像素宽度—默认为: 100。

角度: 设置滑杆是否为一个角度值。这项参数可以是 true 或 false—默认为: false。

水平: 设置滑杆在显示时是否为一条横向摆放 (true)、还是一条纵向摆放 (false) 的线段—默认为: true。

动画：设置滑杆自动播放动画—默认为：false。

随机：设置滑杆在[最小值, 最大值]区间上是否设为连续的数值（false），还是在此区间上为随机数（true）—默认为：false。

7 SetPointSize. 点径

`SetPointSize[<Point>, <Number>]`；点径[<点|多边形|多面体|展开图>, <数值 0—9>]。

改变点的大小。

`SetPointSize[<Polygon>, <Number>]`；点径[<点|多边形|多面体|展开图>, <数值 0—9>]。

改变多边形顶点的大小。

`SetPointSize[<Polyhedron>, <Number>]`；点径[<点|多边形|多面体|展开图>, <数值 0—9>]。

改变多面体顶点的大小。

`SetPointSize[<Net>, <Number>]`；点径[<点|多边形|多面体|展开图>, <数值 0—9>]。

改变展开图顶点的大小。

注：自 GeoGebra 5 这个指令可以用于设置多边形、多面体和展开图的顶点大小；“0”隐藏对象。可以将点径设为 0 或是大于 9 的数值，但在属性对话框中只能在范围[0, 9]之内进行更改。

8 SetPointStyle. 点型

`SetPointStyle[<Point>, <Number>]`；点型[<点>, <数值 0_圆点|1_叉形|2_空心点|3_十字|4_菱形|5_空心菱形|6_上三角|7_下三角|8_左三角|9_右三角>]。

依据下表更改指定点的点型（在范围[0, 9]之外的数值无效）。

数值	型号名称	符号
0	圆点	●
1	叉形	×
2	空心点	○
3	十字	+
4	菱形	◆
5	空心菱形	◇
6	上三角	▲
7	下三角	▼
8	左三角	◀
9	右三角	▶

9 SetDynamicColor. 动态颜色

这个指令在不同的英语变型中拼写不同：`SetDynamicColor` (US)、`SetDynamicColour`。

`SetDynamicColor[<Object>, <Red>, <Green>, <Blue>]`；动态颜色[<对象>, <红>, <绿>, <蓝>]。

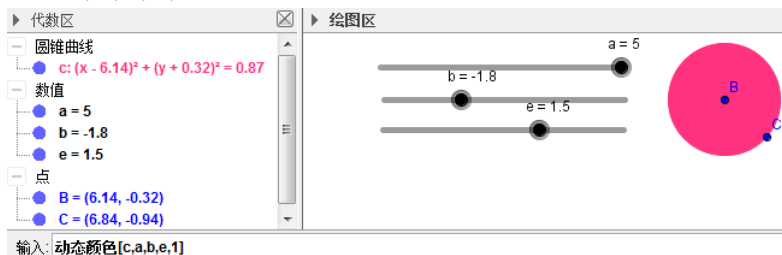
设置对象的动态颜色。

`SetDynamicColor[<Object>, <Red>, <Green>, <Blue>, <Opacity>]`；动态颜色[<对象>, <红>, <绿>, <蓝>, <透明度>]。

<蓝>,<虚实 0—100>]。

设置对象的动态颜色和透明度。

案例：“动态颜色[c, a, b, e, 1]”返回由参数 a、b、c 控制颜色，不透明的圆。



注：所有数值有一个数值范围 0（含/透明的）到 1（不含/不透明的）。

10 ZoomIn. 放大

ZoomIn[<Scale Factor>]; 放大[<缩放因子>]。

基于当前比例将绘图区视图以指定缩放因子进行放大，屏幕中心为缩放中心点。

案例：“放大[1]”不做改变；“放大[2]”放大视图；“放大[0.5]”等同于“缩小[2]”，缩小视图。

ZoomIn[<Scale Factor>, <Center Point>]; 放大[<缩放因子>, <中心点>]。

基于当前比例将绘图区视图以指定缩放因子进行放大，第二个参数指定了缩放中心点。

案例：“放大[2, (0, 0)]”就是以原点为中心，放大视图。

ZoomIn[<Min x>, <Min y>, <Max x>, <Max y>]; 放大[<x 最小值>, <y 最小值>, <x 最大值>, <y 最大值>]。

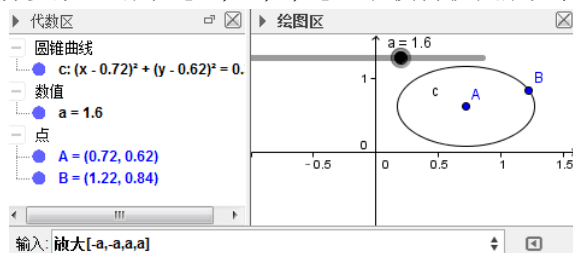
将绘图区视图缩放为由顶点（最小 x, 最小 y）、（最大 x, 最大 y）所确定的矩形区域。

案例：“放大[0, 1, 5, 6]”将视图区域放大到角点自(0, 1)到(5, 6)的矩形内。

注：如果这里的参数是从属对象或具有标签设定（不是未定义数值），视图的边界会变为动态。

若要避免这种情况，可使用复制自由对象指令。

案例：如 a 是一个滑杆参数，“放大[-a, -a, a, a]”就使得视图放大依赖于滑杆 a。

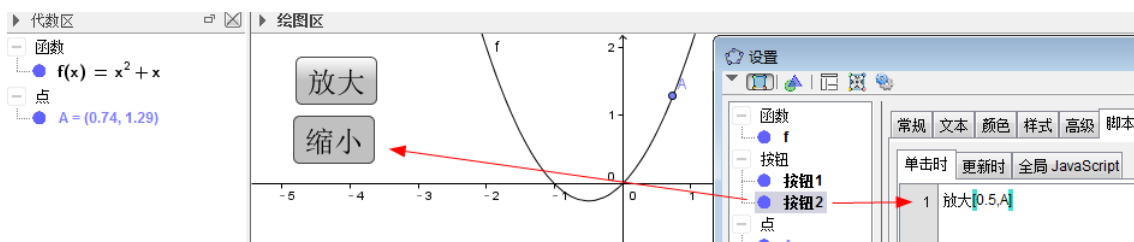


注：如果存在多个绘图区视图，则本指令用于激活使用的那个视图。

参见“缩小”、“活动视图”、“平移视图”指令。

案例：放大的使用技巧-由切线来线性近似（逼近）函数。

创建一个点 A 在其之上的函数 f，以及一个编程代码（脚本）为“放大[2, A]”的按钮，可以观察到，当放大到一个相当大的“放大倍率”时，函数看上去相当于这个点上的切线。在不同于点 A 的位置上使用另一个具有缩小指令的按钮，能让证实这种情况。



11 SetValue. 赋值

SetValue[<Boolean>, <0|1>]; 赋值[<布尔对象>, <0|1>]。

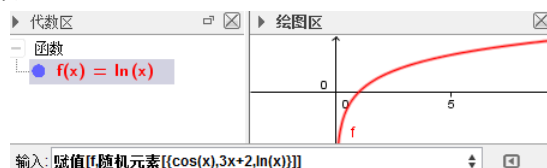
给一个布林型对象/复选框赋值: 1=true, 0=false。

案例: 如果 b 是布林型对象, “赋值[b, 1]” 设定布林型对象 b 为 “true”。

SetValue[<Object>, <Object>]; 赋值[<对象 1>, <对象 2>]。

假设 A 是第一个对象, B 是第二个对象。如果 A 是自由对象或是一个附着在路径或区域上的点, 则取对象 2 的当前值 (也就是, 此后对象 1 的值并不会再因为对象 2 的值发生改变而改变)

案例: 如果 f 是一个函数, 使用 “赋值[f, 随机元素[{cos(x), 3x+2, ln(x)}]]” 定义随机量, 函数 f 会是列表中提供的函数之一。

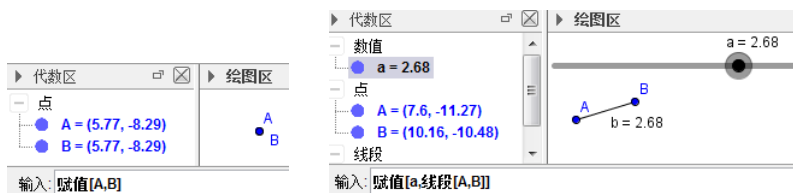


注: 因为是 “当前值”, 在指令栏重复执行这个指令, 函数 f 才会随机变为列表中的一个函数。

“赋值[A,B]” 是使用点 B 的值作为当前 A 的值, 但点 B 移动后, 点 A 的值不随着变动;

“赋值[A, (0, 0)]”, 给点 A 的当前坐标设定为 (0, 0);

“赋值[a, 线段[A, B]]” 将线段 AB 的当前值赋予变量 a, 变为 a 的当前值, 但 a 可以改变。当 A 或 B 改变后, a 并不随着变。



SetValue[<List>, <Number>, <Object>]; 赋值[<列表>, <列表索引>, <对象>]。

设 n 是 “数值” (列表索引)。赋值指令设置自由列表中第 n 个元素的值为对象的当前值。“数值” 最大为列表 L 长度+1。

案例: 若 L 是一个列表型对象, 值为 {1, 2, 3, 4, 5}; n 是一个数值型对象, 值为 3; a 是一个值为 9 的对象。设定值[L, n, a] 将列表 L 的值设置为 {1, 2, 9, 4, 5}。



注: 如果想要在脚本中更改一个对象的值, 尽量使用 “赋值” 指令而不是 “=”。

12 Checkbox. 复选框

Checkbox[]; 复选框[]。

创建一个复选框。

Checkbox[<Caption>]; 复选框[“<标题>”]。

创建给定标题的复选框。

Checkbox[<List>]; 复选框[<列表>]。

创建一个复选框, 其不被勾选时, 隐藏列表中的对象。

案例：设点 A 和点 B。“复选框[{A, B}]”创建复选框 c。当 c 被点击，A 和 B 显示，否则，隐藏。
Checkbox[<Caption>, <List>]; 复选框[“<标题>”, <列表>]。
 创建一个指定标题的复选框，其不被勾选时，隐藏列表中的对象。



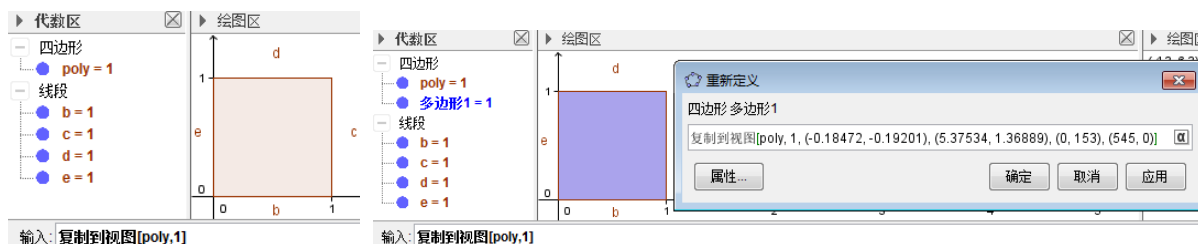
注：参见“复选框”工具；如果两次复选框包括了相同的对象，那么这个对象只受后面创建的复选框控制。

13 AttachCopyToView. 复制到视图

AttachCopyToView[<Object>, <View 0|1|2>]; 复制到视图[<对象>, <视图 0|1|2>]。

如果视图=0，会创建给定对象一个副本。当“视图=1”或“视图=2”时，这个指令在指定绘图区中创建给定对象大小不变的副本。

案例：设“poly=多边形[(0, 0), (1, 0), (1, 1), (0, 1)]”。如果绘图区 I 被激活，“复制到视图[poly, 1]”在同样的位置创建一个与原多边形大小相同的正方形。



参数意义：

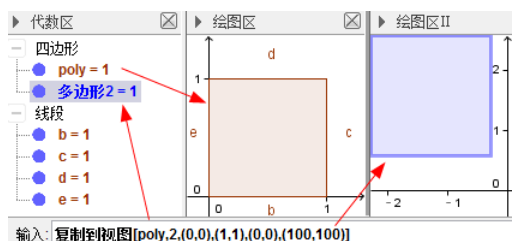
- 0: 副本出现在绘图区，跟随画面缩放变化，跟随对象移动；
- 1: 副本出现在绘图区，不跟随画面缩放变化，只跟随对象移动；
- 2: 副本出现在绘图区 II，不跟随画面缩放变化，只跟随对象移动。

注：一旦副本被创建，另外三个（位置）参数被添加到指令的定义之中。

AttachCopyToView[<Object>, <View 0|1|2>, <Point 1>, <Point 2>, <Screen Point 1>, <Screen Point 2>]; 复制到视图[<对象>, <视图 0|1|2>, <点 1>, <点 2>, <屏幕点 1>, <屏幕点 2>]。

如果视图=0，会创建给定对象一个副本。当“视图=1”或“视图=2”时，这个指令创建在指定绘图区中大小变化为点 1 绑定到屏幕点 1，点 2 绑定到屏幕点 2 的给定对象副本。

案例：设“poly=多边形[(0, 0), (1, 0), (1, 1), (0, 1)]”。如果绘图区 I 被激活，“复制到视图[poly, 2, (0, 0), (1, 1), (0, 0), (100, 100)]”在绘图区左上角创建 100 像素 x100 像素的正方形。



注：即使对象的某些部分在角点 1 和 3 的矩形框选范围以外，其所有部分都会被映射。

视图参数的意义：

- 0: 在原对象位置复制出对象，复制出的对象的大小和位置跟随原对象，后面的对应关系无作用。
- 1: 在正确的位置复制出对象，复制出的对象的大小不会缩放，位置跟随原对象。
- 2: 在偏移了一个正确的位置的地方复制出对象，复制出的对象的大小会缩放，位置跟随原对象。

后面四个点的意义：

\langle 点 1 \rangle , \langle 点 2 \rangle , \langle 屏幕点 1 \rangle , \langle 屏幕点 2 \rangle 表示的是一种对应关系。其中点 1 与屏幕点 1 对应，点 2 与屏幕点 2 对应。屏幕点的坐标系是以视图的左上角为坐标原点，向右是 x 轴正方向，向上是 y 轴正方向，单位是像素。

恰当的对对应关系例如 $(0, 0)$, $(10, -10)$, $(0, 0)$, $(100, 100)$

对象向上复制对象也向上，对象向右复制对象跟随向右。

如果对应关系直接写为 $(0, 0)$, $(10, 10)$, $(0, 0)$, $(100, 100)$

对象向上复制对象也向下，对象向右复制对象跟随向右。

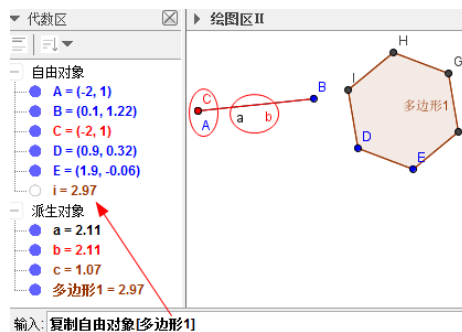
对应关系中尽量不使用点对象（例如 A），那样对应关系会变得非常复杂。

14 CopyFreeObject. 复制自由对象

`CopyFreeObject[<Object>]`；复制自由对象[<对象>]。

创建对象的一个自由副本。

案例：有自由点 A、B，派生对象线段 AB (a) “复制自由对象[A]”和“复制自由对象[a]”都能得到源对象相同位置的一个新对象分别是点 C 和线 b，这两个新对象是自由对象。但如果“复制自由对象[多边形 1]”会得到一个数值（多边形面积），并不能得到一个新的自由的多边形。



15 SetPerspective. 格局

本页涉及的特性只支持 geoGebra5.0

`SetPerspective[<Text>]`；格局[“<文本 A_代数区|B_概率统计|C_运算区|D_绘图区 II|G_绘图区|L_作图过程|P_属性|R_数据分析|S_表格区|T_3D 绘图区>”]。

更改视图的布局和视觉效果，文本参数描述了所需的布局。文本应是一个有效的表达式-视图是由变量（字母）表示：水平排列的字母代表区域水平并列放置，用分号“/”划分的表示它们垂直排列。

字母	视区
A	代数区
B	概率统计
C	运算区
D	绘图区 II
G	绘图区
L	作图过程

P	属性
R	数据分析
S	表格区
T	3D 绘图区

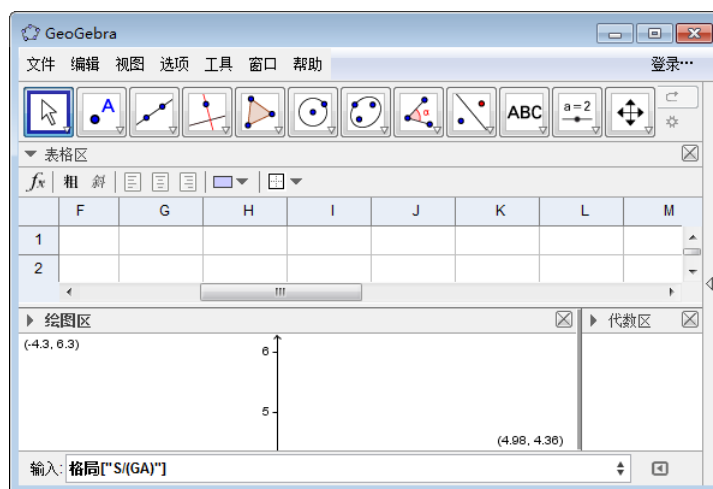
案例：

“格局[“G”]”：设置只显示绘图区。

“格局[“AGS”]”：设置显示代数区、绘图区和表格区，且水平排列。

“格局[“S/G”]”：设置显示表格区和绘图区，表格区在上方，绘图区在下方排列。

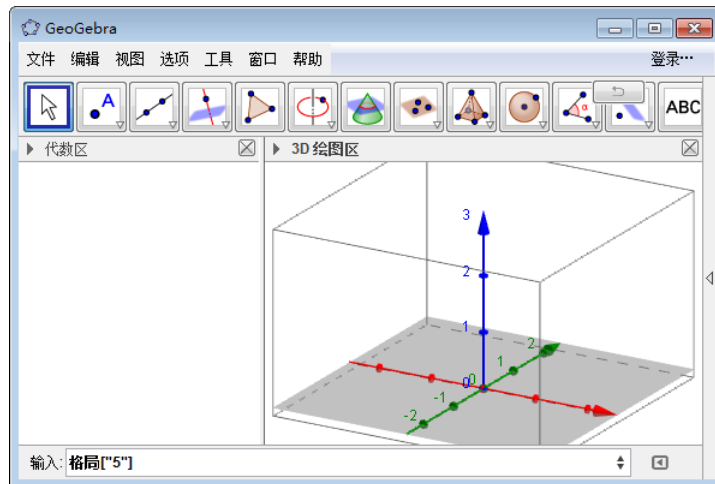
“格局[“S/(GA)”]”：如上条，底部左边是绘图区，右边是代数区。



作为替代，文本中可以使用如下预先定义好视区的数字：

文本	视区
“1”	代数区和绘图区
“2”	几何区(绘图区)
“3”	表格区
“4”	运算区
“5”	3D 绘图区
“6”	概率统计

这些大致分别对应于“AG”，“G”，“SG”，“CG”，“AT”和“B”，但也可能会影响到“指令栏”和“工具栏”的显示。



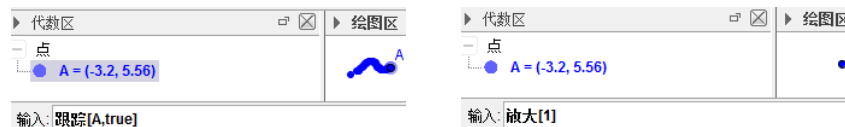
注:参见“活动视图”指令。

16 SetTrace. 跟踪

`SetTrace[<Object>, <true | false>]`; 跟踪[<对象>, <true|false>]。

切换特定对象的跟踪“开/关”。

案例: 创建点 A, 然后键入“跟踪[A, true]”。选择“移动”工具且拖动点, 显示它的踪迹。



注: 使用“放大[1]”可以清除所有踪迹, 但不是唯一方法。

17 UpdateConstruction. 更新作图

`UpdateConstruction[]`; 更新作图[]。

重算全部的对象(随机数重新生成)同于“F9”键或“Ctrl+R”。

注: 若要刷新视图(例如移除绘图区视图中的跟踪轨迹)可以用“放大[1]”来代替, 这等同于“[Ctrl]+[F]”(刷新)。如果运用了两个绘图区视图, 还需要首先使用激活视图[1]或激活视图[2]。

下文言及的特性只支持 GeoGebra 5.0。

`UpdateConstruction[<Number of times>]`; 更新作图[<更新次数>]。

执行指令更新作图几次。

案例:“更新作图[2]”刷新作图两次(如将多次投掷骰子的结果记录到表格)。若要多次执行这个指令, 可以运用“序列[更新作图[], k, 1, 10]”。

18 SetTooltipMode. 工具提示模式

`SetTooltipMode[<Object>, <Number>]`; 工具提示模式[<对象>, <数值 0_自动|1_开启|2_关闭|3_标题|4_下一单元格>]。

从下面表单中选给定对象的工具提示模式(在范围[0, 4]以外的值按照 0 对待):

数值	模式
0	自动

1	开
2	关
3	标题
4	下一单元格

当鼠标悬停在对象上方时，出现的提示信息可以入上表设定。

注：在范围[0, 4]之外的数值被处理为 0；当对象是单元格创建的对象时，“4”表示显示的是其右边单元格内的信息。

19 SetActiveView. 活动视图

SetActiveView[<View>]；活动视图[<视图编号 1/G_绘图区|2/D_绘图区 II |-1/T_3D 绘图区|A_代数区|S_表格区|C_运算区>]

激活给定绘图区。

注：参见“放大”、“缩小”、“平移视图”和“格局”指令。“1/G_绘图区”表示“1”或者“G”都可以激活绘图区。

案例：“活动视图[T]”和“活动视图[-1]”都是激活 3D 绘图区。

20 ParseToNumber. 解析到数

ParseToNumber[<Number>, <String>]；解析到数[<数值>, <字符串>]。

在使用本指令前定义数值 a 为自由对象，解析字符串且保存为 a。

案例：定义 a=3 和 text1="6"。“解析到数[a, text1]”返回 a=6。相当于将文本型数字转为数值型数字。

注：用到的文字必须是全数值文字，否则结果为“未定义”；参见“解析函数”指令。

21 ParseToFunction. 解析函数

ParseToFunction[<Function>, <String>]；解析函数[<函数>, <字符串>]。

在使用本指令前必须定义为函数 f 为自由对象，解析字符串且保存为函数 f。

案例：定义“f(x)=3x²+2”和“text1=f(x)=3x+1”。“解析函数[f, text1]”返回“f(x)=3x+1”。相当于将文本型的函数表达式，转换为可进行函数运算的表达式。

注：用到的文字必须是全函数文字（x 的多项式或 f(x)=x 的多项式），否则结果为“未定义”；参见解析到数指令。

22 StartRecord. 开始记录

StartRecord[]；开始记录[]。

如果当前是暂停，重启所有到电子表格的记录动作，且为每个对象保存一个值。

StartRecord[<Boolean>]；开始记录[<true|false>]。

当布尔值是“false”，暂停记录到表格的动作，否则重启之（且为每个对象保存为一个值）

23 Pan. 平移视图

Pan[<x>, <y>]；平移视图[<x>, <y>]

平移视图，x 为向左平移的像素量，y 为向上平移的像素量。

Pan[<x>, <y>, <z>]; 平移视图[<x>, <y>, <z>]。

如果是 3D 视图则在 (x, y, z) 方向移动当前视图，否则在 2D 视图在 (x, y) 方向移动视图。

注：如果同时呈现多个绘图区，会用于激活的区域；参见“放大”、“缩小”和“激活绘图区”指令。

24 StartAnimation. 启动动画

StartAnimation[]; 启动动画[]。

重启被暂停的动画。

StartAnimation[<Boolean>]; 启动动画[<true|false>]。

如果布尔值是 false，暂停所有动画，否则重启动画。

StartAnimation[<Point or Slider>, <Point or Slider>, ...]; 启动动画[<参数|点>, <参数|点>, ...]。

启动给定的点和滑杆动画，点必须是在路径上。

StartAnimation[<Point or Slider>, <Point or Slider>, ..., <Boolean>]; 启动动画[<参数|点>, <参数|点>, ..., <true|false>]。

启动(当布尔值=true)或长停(当布尔值=false)给定点和滑杆的动画，点必须是在路径上。

注：参见“动画”。

25 StartLogging. 启动日志

本页言及的特性只 GeoGebra 5.0 支持。

StartLogging["<Sensor>", <Variable>, "<Sensor>", <Variable>, ...]; 启动日志["<感应器 1>", <变量 1>, "<感应器 2>", <变量 1>, ...]

记录诸如语音等数据。

注：参见“停止日志”指令。

26 Delete. 删除

Delete[<Object>]; 删除[<对象>]。

删除对象以及所有它的从属对象。


案例：“删除[a]”清除掉 a。

CAS Syntax (运算区语法)

Delete[<Object>]; 删除[<对象>]。

在 GeoGebra 中删除对象以及从属对象，并移除运算区中所有指派给这个对象的数值。

案例：“删除[a]”清除掉 a。

注：参见  “删除”工具。

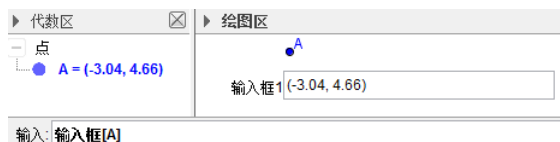
27 InputBox (Textfield) . 输入框

InputBox[]; 输入框[]。

创建一个新的输入框。

InputBox[<Linked Object>]; 输入框[<链接对象>]。

创建一个输入框并将其与一个链接对象相关联。



输入框内的内容与点的坐标链接，移动点，框内数据随变；修改框内数据，点会移动到新的坐标位置。

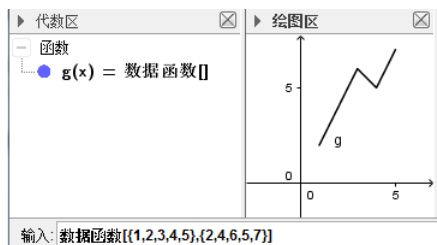
注：参见^{a=1}“输入框”工具。

28 DataFunction. 数据函数

`DataFunction[<List of Numbers>, <List of Numbers>]`；数据函数[<数值列表 1>, <数值列表 2>]。

以给定两个数值列表中的对应值描点并以线段连接，形成新的函数。

案例：



注：两个列表长度必须相同；目前不支持列表名称引用，只支持在指令中直接输入列表。

29 ZoomOut. 缩小

`ZoomOut[<Scale Factor>]`；缩小[<缩放因子>]。

基于当前比例将绘图区视图以指定缩放因子进行缩小，屏幕中心为缩放中心点。

案例：“缩小[2]”缩小视图。

`ZoomOut[<Scale Factor>, <Center Point>]`；缩小[<缩放因子>, <中心点>]。

基于当前比例将绘图区视图以指定缩放因子进行缩小，第二个参数指定了缩放中心点。

案例：“缩小[2, (0, 0)]”：以原点为中心，缩小视图区。

注：“缩小[t]”和“缩小[t, A]”分别等效于“放大[1/t]”和“放大[1/t, A]”；如果存在多个绘图区视图，则用于激活使用的那个；参见“放大”、“活动视图”、“平移视图”指令。

30 SetFixed. 锁定对象

`SetFixed[<Object>, <true | false>]`；锁定对象[<对象>, <true|false>]。

使对象锁定(选 true)或不锁定(选 false)。

注：锁定的对象可以右键拖移。

31 SetSeed. 锁定随机数

`SetSeed[<Integer>]`；锁定随机数[<整数>]。

锁定随机数使得后续随机数由种子决定。

案例：“锁定随机数[33]”：无可见现象。

注：参见“区间随机数”指令、“随机元素”指令、“均匀分布”指令、“随机二项分布数”指令、

“正态分布随机数”指令、“泊松分布随机数”指令。

锁定随机数指令必须和随机数指令同时使用才有效，单独在输入框输入无效。可写在对象脚本，或嵌套执行指令。

1. 脚本：

```
SetSeed[2]
```

```
a=RandomBetween[0,100]
```

2. 执行指令

“执行[{"SetSeed[5]","a=RandomBetween[0,100]"}]”（请查阅执行指令）

32 SetFilling. 填充

SetFilling[<Object>, <Number>]; 填充[<对象>, <数值 0—1>]。

更改给定对象的模糊度（不透明程度）。数值必须是区间[0, 1]，“0”意味着透明，而“1”意味着 100%不透明，忽略其他数值。

33 StoptLogging. 停止日志

本页言及的特性只 GeoGebra 5.0 支持。

StopLogging[]; 停止日志[]。

停止记录诸如语音类数据。

注：参见启动日志指令。

34 SetLayer. 图层

SetLayer[<Object>, <Layer>]; 图层[<对象>, <层次>]。

设置指定对象的图层，图层的数值必须是在数集 {0, 1, ..., 9} 中的整数。

注：0 表示最下面的图层，9 表示最上面的图层。

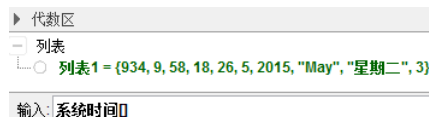
35 GetTime. 系统时间

GetTime[]; 系统时间[]。

创建一个以下序的当前系统时间的列表：

毫秒、秒、分、时（0-23）、日、月（1-12）、年、月份（文本格式）、星期（文本格式）、星期日数（1=周日、2=周一，类推）

案例：“系统时间[]”返回一个如 {647, 59, 39, 23, 28, 2, 2011, "February", "Monday", 2} 的列表。

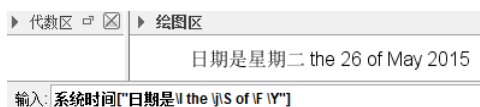


GetTime["<Format>"]; 系统时间["<格式>"]。

当格式为文本，使用反斜杠 (\) 前缀下列字符：

d, D, j, l, N, S, w, z, W, F, m, M, n, t, L, Y, y, a, A, g, G, h, H, i, s, U—这些字符的解释在 <http://php.net/manual/en/function.date.php>

案例：“系统时间[日期是\l the \j\S of \F \Y]”会给出“日期是 Thursday the 5th of July 2012”。



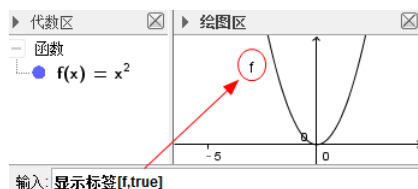
年: Y (2014), y(14)
 月: F 或 M(Jul), m 或 n (7)
 日: d 或 J (11); z (191:第几天)
 星期: l(小写L)(星期五), N 或 w (5)
 时: G 或 H (21) g 或 h (9)
 分: i (20)
 秒: s (57)
 上下午: a (pm), A (PM)

36 ShowLabel. 显示标签

ShowLabel[<Object>, <Boolean>]; 显示标签[<对象>, <true|false>]。

在绘图区显示或隐藏给定的对象。

案例: 假设 “ $f(x)=x^2$ ”。 “显示标签[f, true]” 显示函数的标签。



37 SetVisibleInView. 显示对象

SetVisibleInView[<Object>, <View Number 1|2>, <Boolean>]; 显示对象[<对象>, <视图编号 1_绘图区|2_绘图区II>, <true|false>]。

在指定的绘图区(1 或 2)显示(true)或隐藏(false)对象。

38 SetConditionToShowObject. 显示条件

SetConditionToShowObject[<Object>, <Condition>]; 显示条件[<对象>, <条件>]。

设置对象在指定的条件下显示。

范例: 设有一个 A 点与一个数值变量 t, 如果作以下的设定: “显示条件[A, t>1]”

这时只有在 “t>1” 的情况下, A 才会显示。

注: 对象名与条件式都不需要用 “双引号”。

显示条件可以是真假值、数值对象的大小关系 ($=$ 、 \neq 、 \leq 、 \geq 、 $>$ 、 $<$)、对象的属于关系(\in)等。

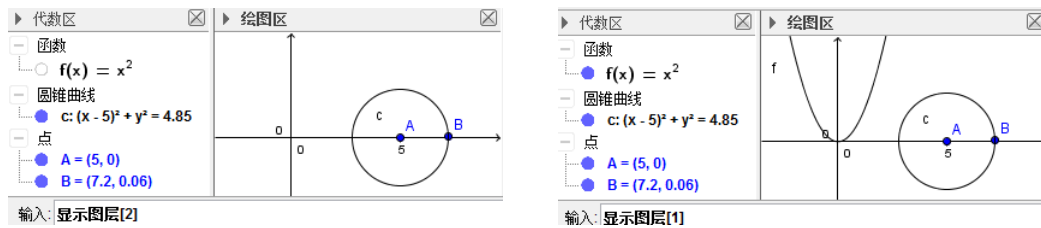
如: “显示对象[A, a]”: 当 a 有意义时显示 A; “显示对象[A, x(B) \neq x(C)]”: 当 B 和 C 的横坐标相同时显示 A; “显示对象[A, 2<s<9]”: 当 s 在区间内时显示 A; “显示对象[A, s \in {0, 1, 2, 3, 4, 8, 9}]”: 当 s 是列表中任意一个元素时显示 A。

39 ShowLayer. 显示图层

ShowLayer[<Number>]; 显示图层[<数值>]。

使给定图层的所有对象处于显示状态。不覆盖“条件显示”。

案例：“显示图层[2]”使所有处于第二图层的对象显示。



函数 f 在第一图层，圆在第二图层，在代数区都设置为隐藏。分别执行两条指令后显示如图。

40 ShowGrid. 显示网格

ShowGrid[]; 显示网格[]。

在激活绘图区显示网格。

ShowGrid[<Boolean>]; 显示网格[<true|false>]。

在激活绘图区显示/隐藏网格。

案例：“显示网格[true]”在激活绘图区显示网格; “显示网格[false]”在激活绘图区隐藏网格。

ShowGrid[<View>, <Boolean>]; 显示网格[<视图 1|2>, <true|false>]。

在由数字 1 或 2(在 GeoGebra5 的 3D 绘图区中, 或 3)指定的绘图区中显示/隐藏网格。

案例：“显示网格[1, true]”在绘图区显示网格; “显示网格[2, false]”在绘图区 II 中隐藏网格。

注：参见“显示坐标轴”指令。

41 ShowAxes. 显示坐标轴

ShowAxes[]; 显示坐标轴[]

在激活绘图区显示坐标轴。

ShowAxes[<Boolean>]; 显示坐标轴[<true|false>]。

在激活绘图区显示/隐藏坐标轴。

案例：“显示坐标轴[true]”在激活的绘图区显示坐标轴; “显示坐标轴[false]”在激活的绘图区隐藏坐标轴。

ShowAxes[<View>, <Boolean>]; 显示坐标轴[<视图>, <true|false>]。

在由数字 1 或 2(在 GeoGebra5 的 3D 绘图区中, 或 3)指定的绘图区中显示/隐藏坐标轴。

案例：“显示坐标轴[1, true]”在绘图区中显示坐标轴; “显示坐标轴[2, false]”在绘图区 II 中隐藏坐标轴。

注：参见“显示网格”指令。

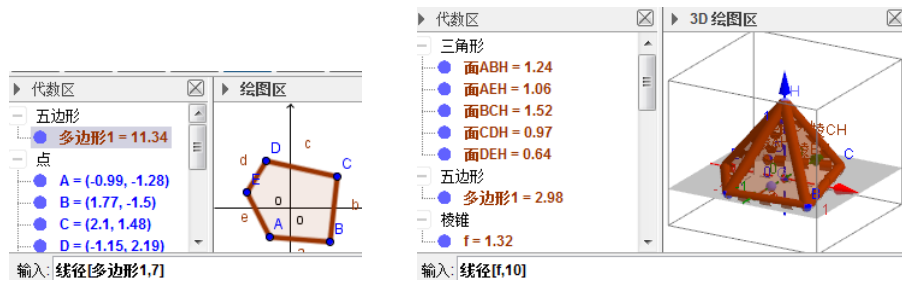
42 SetLineThickness. 线径

SetLineThickness[<Object>, <Number>]; 线径[<直线|射线|线段>, <数值 1-13>]

设 N 为数值。线径指令将指定线的线径设置为 $N/2$ 像素。

注：自 GeoGebra 5 这个指令可用于设置多边形和多面体的线径(边线和棱)。

只支持范围 0 到 13 内的数值。



43 SetLineStyle. 线型

SetLineStyle[<Line>, <Number>]; 线型[<直线|射线|线段>, <数值 0_实线|1_长划线|2_短划线|3_点|4_划线-点>]

依据下表更改指定对象的线型（在范围[0, 4]之外的数值无效）。

数值	型号
0	实线
1	长虚线
2	短虚线
3	点虚线
4	点划线

44 SelectObjects. 选择对象

SelectObjects[]; 选择对象[]。

取消所有已选定对象。

SelectObjects[<Object>, <Object>, ...]; 选择对象[<对象 1>, <对象 2>, ...]。

取消当前所有选定对象并选定参数列出的对象。所有参数对象必须是有标签的对象。

案例：设有点 A、B 和 C。“选择对象[A, B, C]”选定点 A、B 和 C；指令“选择对象[中点[A, B]]”无效（因为中点没有提前被定义，也就是没有标签）。

注：这个指令会马上取消对象在进程中的任何拖拽操作（在脚本中十分有用）。

45 SetColor. 颜色

这个指令在不同的英语变型中拼写不同：SetColor (US)、SetColour (UK+Aus)。

SetColor[<Object>, <Red>, <Green>, <Blue>]; 颜色[<对象>, <红>, <绿>, <蓝>]。

更改指定对象的颜色。红、绿和蓝能表示大量的相关颜色组成，其值最小 0，最大 1。超过这个区间的数值 t，使用函数“ $2|t/2 - \text{round}(t/2)|$ ”映射到 0 到 1 的区间。

SetColor[<Object>, <"Color">]; 颜色[<对象>, <"颜色">]。

改变给定对象的颜色。“颜色”键入文本。这个指令接受超过 100 个英语颜色名称（见参考：颜色）。其中一些可以使用本地语言，列示如下。

Black	Indigo	Light Gray
Dark Gray	Purple	Pink
Gray	Brown	Violet

Dark Blue	Orange	Yellow
Blue	Gold	Light Yellow
Dark Green	Lime	Light Orange
Green	Cyan	Light Violet
Maroon	Turquoise	Light Purple
Crimson	Light Blue	Light Green
Red	Aqua	White
Magenta	Silver	

46 HideLayer. 隐藏图层

HideLayer[<Number>]; 隐藏图层[<图层编号(0-9)>]。
使所有在指定图层上的对象不可视。不会覆写条件显示。
注: 参见“显示图层”指令。

47 RunClickScript. 运行单击脚本

RunClickScript[<Object>]; 运行单击脚本[<对象>]。
运行与对象相关联的“单击时”脚本(如果有的话)。

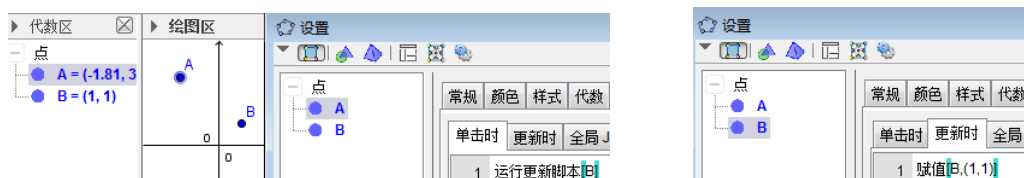
案例: 设 A 和 B 是两个点。点 B 的“单击时”脚本是“赋值[B, (1, 1)]”。设置点 A 的“单击时”脚本为“运行单击脚本[B]”, 当点 A 被点击时, 点 B 会移动到(1, 1)位置。



注: 参见“运行更新脚本”指令。

48 RunUpdateScript. 运行更新脚本

RunUpdateScript[<Object>]; 运行更新脚本[<对象>]。
运行对象的“更新时”脚本(如果有的话)。



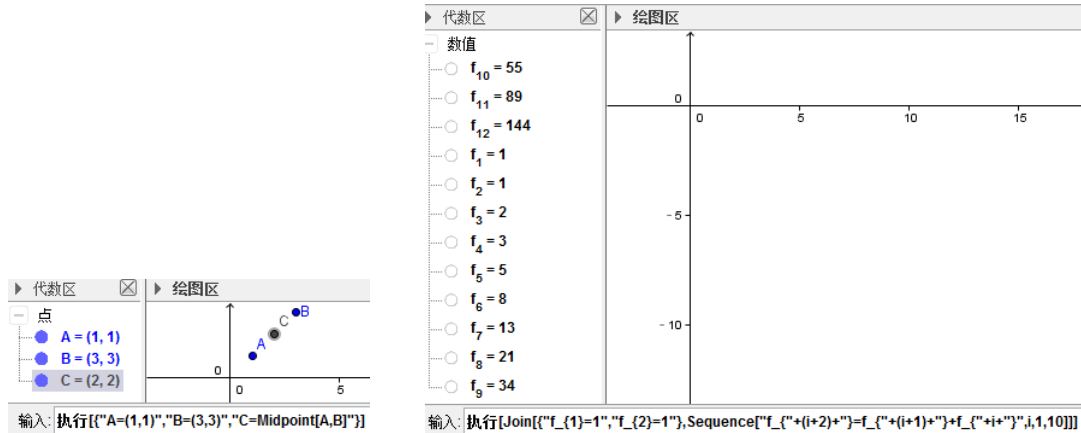
注: 参见“运行单击脚本”指令。

49 Execute. 执行

Execute[<List of Texts>]; 执行[<文本列表>]。
执行以文本形式输入在列表中的指令。

注： 请注意使用这个指令时，文本中的指令名称必须使用英文，不管在 GeoGebra 语言选项中勾选了何种语言。

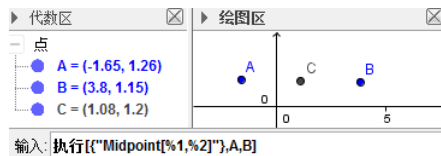
案例： “执行[{"A=(1,1)", "B=(3,3)", "C=Midpoint[A,B]"}]" 创建点 A、B 和它们的中点 C。“执行[Join[{"f_{1}=1", "f_{2}=1"}, Sequence["f_{"+(i+2)+"}=f_{"+(i+1)+"}+f_{"+i+"}"], i, 1, 10]]]" 创建斐波那契数列的前 10 个元素。



`Execute[<List of Texts>, <Parameter>, ... , <Parameter>];` 执行[<文本列表>, <参数 1>, <参数 2>, ...]。

用“参数 1”代换“%1”，“参数 2”代换“%2”，并对列表中的文本以此类推。最多可以指定 9 个参数。代换之后，所得脚本会被执行。

案例： “执行[{"Midpoint[%1,%2]"}, A, B]" 创建线段 AB 的中点。



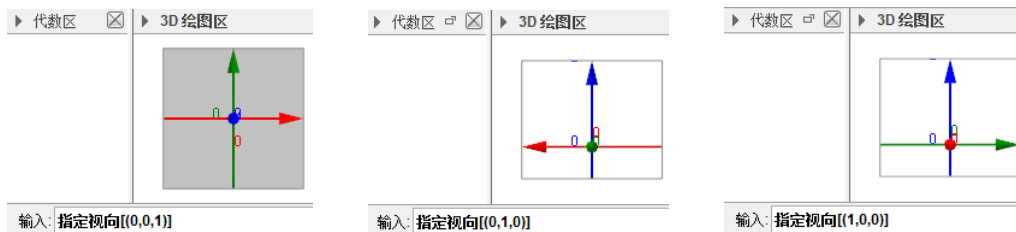
注： 这个指令中的指令名称必须是英语才能工作。

50 SetViewDirection. 指定视向

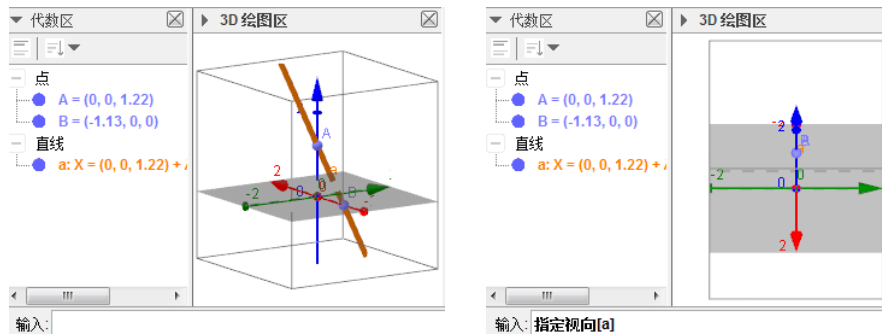
`SetViewDirection[<Direction>];` 指定视向[<方向, eg. (0, 0, 1)>]。

设置 3D 视图为指定方向

案例： “指定视向[(0, 0, 1)]”：视角为从 z 轴正方向向负方向正视。



注： 视图方向也可以设置指向一条直线、一个线段和一个平面，等等；参见 “指定视向”工具。



如上图左有直线 a；上图右就是直线 a 指定的视向结果（视线沿着 a）。

51 CenterView. 中心定位

CenterView[<Center Point>]; 中心定位[<视图中心坐标(x,y) | 视图中心点>]。

把绘图区的中心移动到指定点。

案例：“中心定位[(0,0)]”将原点移动到屏幕（视图区）的中心。

注：如果存在多个绘图区，则应用在被激活的那一个。

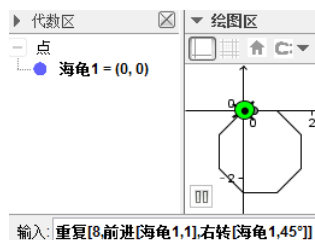
52 Repeat. 重复

Repeat[<Number>,<Scripting Command>,<Scripting Command>,...]; 重复[<数值>,<脚本指令 1>,<脚本指令 2>,...]

当 n 是给定的数值，重复脚本指令 n 次。

案例：依次执行以下指令，会动态绘制出正八边形。

“海龟[]”；点击视区左下角的“▶”（播放）按钮；“重复[8,前进[海龟 1,1],右转[海龟 1,45°]]”。



53 Rename. 重命名

Rename[<Object>,<Name>]; 重命名[<对象>,<名称>]。

将指定对象的标签设置为指定名称。

案例：设“ $c:x^2+2y^2=2$ ”。“重命名[c,ell]”设置圆锥曲线的标签为“ell”。

54 SetSpinSpeed. 转速

本页言及的特性只 GeoGebra 5.0 支持。

SetSpinSpeed[<Number>]; 转速[<速度值>]。

设置 3D 视图绕当前垂直屏幕的显示轴的转速。键入值的符号和值定义以下旋转：

如果数字为正数，则 3D 视图逆时针旋转。

如果数字为负数，那么 3D 视图顺时针旋转。

如果数字为 0，那么 3D 视图不旋转。

55 SetCoords. 坐标

SetCoords[<Object>,<x>,<y>]; 坐标[<对象>,<横坐标>,<纵坐标>].

改变笛卡尔坐标系中已有对象坐标。这个指令使用了坐标的值, 不是对象的定义, 所以对象仍然是自由对象。

注: 这个指令可作用于滑杆参数、布尔值、按钮、复选框、输入框、图像等, 不包括几何对象和函数。如果选择了“屏幕上的绝对位置”, 横坐标 x 、纵坐标 y 则为屏幕像素单位。

56 SetAxesRatio. 坐标轴比例

SetAxesRatio[<Number>,<Number>]; 坐标轴比例[<数值 1>,<数值 2>].

更改激活绘图区中的坐标轴比例, 使得 x -轴上 X 的单位对应为“数值 1”的像素值, y -轴上 Y 的单位对应“数值 2”, 点(0,0)保持其坐标。(x-轴不动, y -轴按比例缩放)

下文言及的特性只支持 GeoGebra 5.0。

SetAxesRatio[<Number>,<Number>,<Number>]; 坐标轴比例[<数值 1>,<数值 2>,<数值 3>].

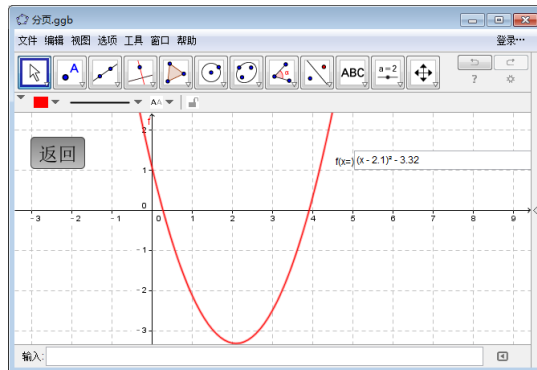
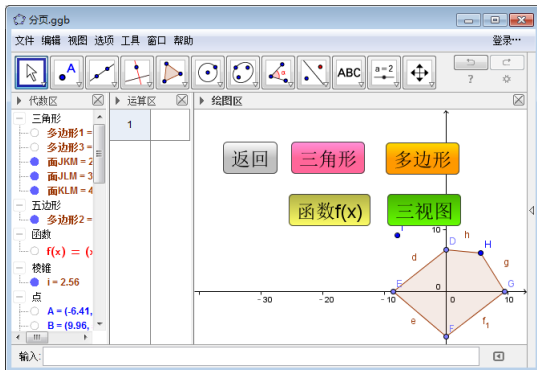
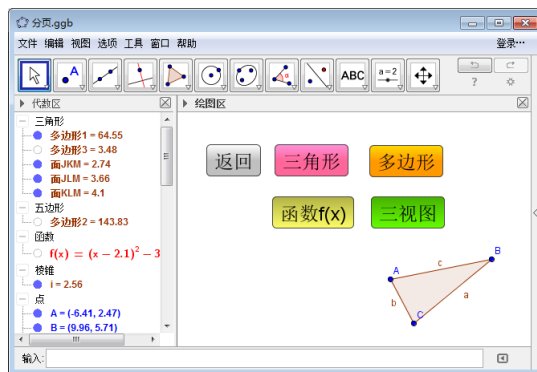
相似的语法, 用于 3D 绘图区。

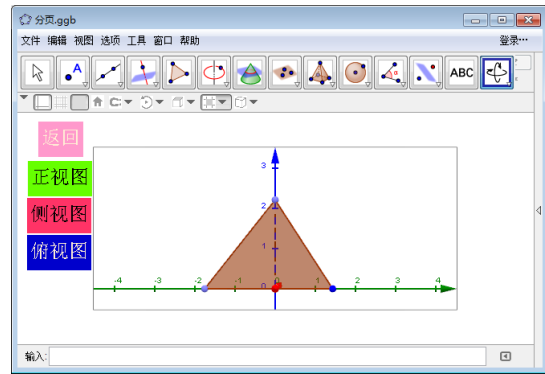
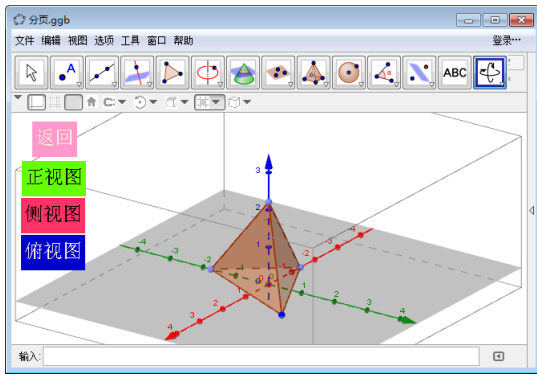
附二, 多页面案例

GeoGebra 构图漂亮简洁, 支持新建 Windows 窗口和 GeoGebra 窗口, 但在一个 GeoGebra 窗口中不支持多个页面, 略有缺憾。本案例介绍使用点击按钮“切换”出多个页面的效果。

制作目标: 制作一个文档, 从表面看有如下几页。封皮、三角形、多边形、函数图象和棱锥三视图。通过点击按钮可以切换各页, 在各页中有返回到封皮页的按钮。

制作原理: 将封皮的图片三个点锁定到“绘图区”的角点上, 如此, 封皮就能随绘图区大小改变其大小了。将三角形和多边形构造在“绘图区”、函数图象构造在“绘图区 II”中、棱锥构造在“3D 绘图区”中。通过布尔值显示对象或者绘图区, 表面上实现不同“页”显示不同内容的效果。





各页特点:

封皮页，使用了图片锁定其角点到 GeoGebra 的角点，随窗口变化大小。有 4 个链接按钮能到其他“页”。

三角形页，有代数区和绘图区，绘图区没有坐标轴。有多个按钮。

多边形页，有代数区、运算区和绘图区，绘图区有坐标轴。有多个按钮。

函数 f 页，只有绘图区，有坐标轴和坐标网格，还有一个“链接输入框”。有多个按钮。

三视图页，只用 3D 绘图区，有一个棱锥和几个文本假装是按钮。点击文本可以返回封皮或者看到棱锥的某一个视图。

制作步骤:

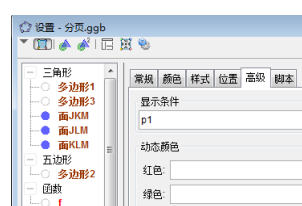
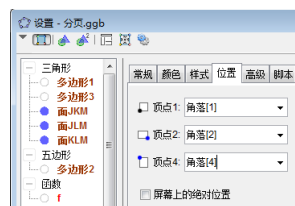
假设板友已经有画板基本操作技能，只介绍关键步骤。

一、构造基本图形和部件。

- 1、使用“图像工具”在“绘图区”插入图片。
- 2、构造 5 个真假值 p1、p2、p3、p4 和 p5。
- 3、在绘图区构造一个三角形 ABC 和一个五边形 DEFG（可以有重叠部分）。
- 4、激活“绘图区 II”，构造函数“ $f(x)=x^2-1$ ”。
- 5、在“3D 绘图区”构造三棱锥。

二、封皮图像片设定

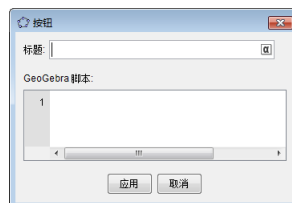
设置图像的属性中的位置和显示条件。



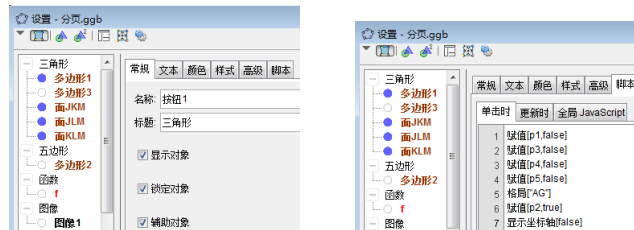
三、构造按钮设置

1、构造按钮。

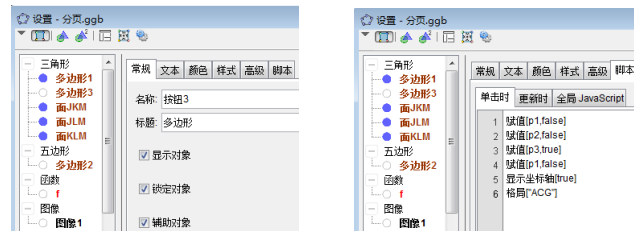
使用“按钮工具”构造按钮，在不同的按钮中设定不同的“标题”和“GeoGebra”脚本。这里输入的标题和脚本，可以在后续的按钮属性中修改。颜色和大小等同样可以在按钮属性中修改。



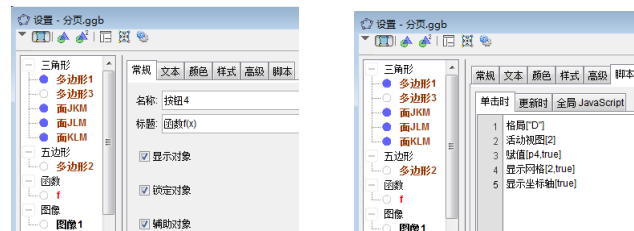
2、“三角形”按钮的标题和脚本。设定“p2=true”，其余真假值为“false”；显示“代数区”和“绘图区”；不显示坐标轴。“P2”是三角形的显示条件。



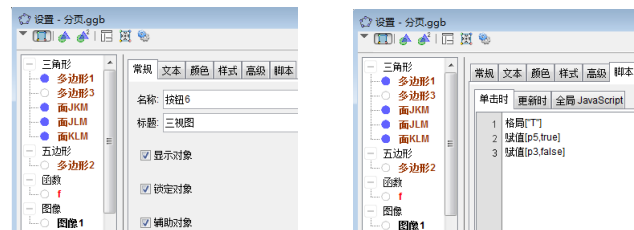
3、“多边形”按钮的标题和脚本。设定“p3=true”，其余真假值为“false”；显示“代数区”、“运算区”和“绘图区”；显示坐标轴。“P3”是多边形的显示条件。



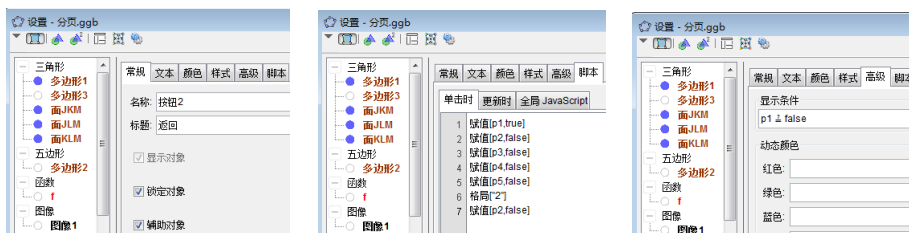
4、“函数 f”按钮的标题和脚本。设定“p4=true”；显示并激活“运算区 II”；显示坐标轴和网格。“P4”是函数图象的显示条件。



5、“三视图”按钮的标题和脚本。设定“p5=true”，“p3=false”；显示“3D 绘图区”。“P4”是棱锥的显示条件。“P3”是棱锥底面的显示条件。（函数图象和棱锥不用真假值显示也可以）



6、“返回”按钮的标题和脚本。只设定“p1=true”，显示无指令栏的“运算区”。按钮条件显示。



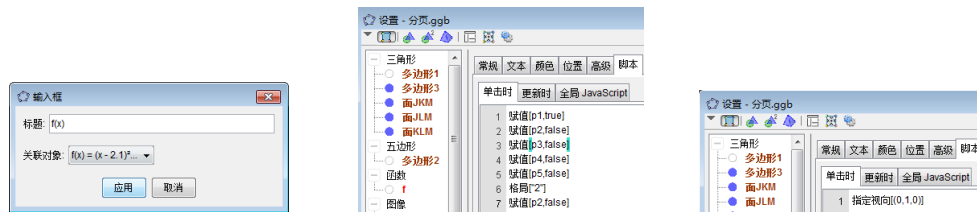
在“运算区 II”还需要给函数建立一个“返回”按钮，方法同上。

四、对象的条件显示。以下面两个多边形显示条件为例，为绘图区中的对象设定相应的显示条件。



五、“函数 f”和“三视图”页的特殊处理。

在“绘图区 II”，使用“输入框”工具，插入了“链接输入框”。链接到了函数 $f(x)$ 。这个输入框内的信息与代数区函数表达式和移动函数图象有交互关联。



在“3D 绘图区”，使用“文本”工具输入文本。“返回”文本的属性设置了“单击时”脚本如上中图。在“正视图”属性中，设置了“单击时”脚本如上右图。“侧视图”和“俯视图”类似设置。

以上只是简单介绍多“页”文档制作，还可以加入更多的脚本指令，各种操作和选项大可优化。实现多“页”显示，还可以使用在指定的图层显示不同对象等其它方法实现。