

NOTAS SOBRE COMANDOS JAVASCRIPT PARA GEOGEBRA

A. Martín Dinnbier
<https://www.geogebra.org/u/adrianmartin>

1 Introducción

2 Enlaces útiles

Lista completa de los comandos en inglés: <https://wiki.geogebra.org/en/Category:Commands>

Lista completa de los comandos de Geogebra para Javascript: <https://wiki.geogebra.org/en/Reference:JavaScript>

3 Funciones

Resumen de funciones de Javascript empleadas en construcciones de Geogebra:

```
1 SetPointSize("A", int style)
```

Listing 1: Cambiar el estilo de un punto

Los posibles valores son estos:

- 0 Full dot
- 1 Cross
- 2 Empty dot
- 3 Plus sign
- 4 Full diamond
- 5 Empty diamond
- 6 Triangle north
- 7 Triangle south
- 8 Triangle east
- 9 Triangle west
- 10 Full dot (but with no outline)

```
1 ap.evalCommand("Delete(A_0)");
```

Listing 2: Borrar elementos en Geogebra

```
1 ap.evalCommand("A=(0,1)");
```

Listing 3: Crear puntos nuevos

```
1 ap.evalCommand("SetConditionToShowObject(A,a)");
```

Listing 4: Modificar la condición de visibilidad de un objeto

```
1 ap.evalCommand("ShowLabel(A,"+false+"");
```

Listing 5: Ocultar etiquetas de un elemento

```
1 ap.evalCommand("SetColor(A,1, 0.2, 0.5)");
```

Listing 6: Cambiar el color de un elemento

```
1 ap.evalCommand("SetFilling( A, 0.1)");
```

Listing 7: Cambiar la opacidad de un elemento. La opacidad debe ser un número de 0 a 1

```
1 ap.evalCommand("SetTrace(A,"+false+"");
```

Listing 8: Activar el rastro de un elemento

```
1 ap.evalCommand("SetVisibleInView("+arrayDeNuevos[arrayDeNuevos.length-3]+",1,"+false+"");
```

Listing 9: Ocultar un elemento

```
1 ap.evalCommand("SetLineThickness("+arrayDeNuevos[1]+", "+grosor+"");
```

Listing 10: Cambiar el grosor de línea

```
1 var niveles=ap.getValue("slide");
```

Listing 11: Extraer/tomar el valor de una variable de Geogebra

```
1 var ap=ggbApplet;  
2 var a=ap.getValue("a");
```

Listing 12: Apuntar a un objeto en Geogebra

4 Segmentos de código útiles

4.1 Manejo de cadenas en Javascript

```
1 var validate = ap.getValue("validate");  
2 var stAnswer = ap.getValueString("stAnswerFunc");  
3 stAnswer = "\"" + stAnswer + "\""; //esta línea de código para que se pueda tratar  
    como una string  
4 var numBrackets = 0;  
5  
6 if (validate) {  
7     numBrackets = char_count(stAnswer, "(");  
8     numPluses = char_count(stAnswer, "+");
```

```

9      if (numBrackets == 3 && numPluses == 2) { //hay paréntesis en el nombre de la
        función que también aparecen
10      ap.evalCommand("BoolBrackets=true");
11      } else {
12      ap.evalCommand("BoolBrackets=false");
13      }
14  }
15
16  //La función char_count cuenta el número de caracteres que aparecen en la cadena str
17  function char_count(str, letter) {
18      var letter_Count = 0;
19      for (var position = 0; position < str.length; position++) {
20          if (str.charAt(position) == "(") {
21              letter_Count += 1;
22          }
23      }
24      return letter_Count;
25  }

```

Listing 13: Cambiar el valor de un Booleano en función de la aparición de paréntesis y signos de suma en una función

4.2 Imprimir todos los campos que genera la ejecución de un comando con un bucle

```

1
2  var r=ap.evalCommandGetLabels("Prism("+arrayDeNuevos[0]+","+size+" )");
3  var elementos=r.split(",");
4  for(var cont=0; cont<elementos.length;cont=cont+1){
5      //la primera línea imprime el "objeto", una cadena de texto del navegador
        ilegible para nosotros
6      alert(elementos[cont]);
7      //la segunda línea imprime el nombre del elemento en el espacio de trabajo de
        Geogebra
8      alert("\ "+elementos[cont]+\ " ");
9  }

```

Listing 14: Imprimir todos los campos que genera la ejecución de un comando con un bucle

4.3 Crear un listener para un objeto concreto de Geogebra

```

1 void registerObjectClickListener(String objName, String JSFunctionName)

```

Listing 15: Adjudicar una función a un objeto para que se ejecute al pinchar

De la ayuda de Geogebra:

Registers a JavaScript function as an click listener for a single object. Whenever the object with the given name is clicked, the JavaScript function JSFunctionName is called using the name of the updated object as its single argument. If objName previously had a mapping JavaScript function, the old value is replaced.

Note: all object click listeners are unregistered when their object is removed or the construction is cleared, see registerRemoveListener() and registerClearListener(). Example: First, register a listening JavaScript function:

```

1 ggbApplet.registerObjectUpdateListener("A", "myAupdateListenerFunction");

```

Whenever the object A is clicked, the GeoGebra Applet will call the Javascript function

```

1 myAupdateListenerFunction("A");

```

Note: an object update listener will still work after an object is renamed.

4.4 Crear un manejador en Javascript global para controlar eventos

```
1 function ggbOnInit() {
2     ggbApplet.registerClickListener("ClickL");
3     /*alert("initial by ggb")*/
4 }
5
6 function ClickL(nam){
7     /*alert("Click = "+nam);*/
8     ggbApplet.evalCommand('ClickNam="'+nam+'"' );
9     ggbApplet.evalCommand('ClickType="'+ggbApplet.getObjectType(nam)+'"' )
10 }
```

Listing 16: Crear un manejador en Javascript global para controlar eventos

```
1 function getRandomArbitrary(min, max) {
2     return Math.random() * (max - min) + min;
3 }
```

Listing 17: Función para generar un número aleatorio entre dos valores

```
1 //ARBOL
2 //Crea un esquema de fractales a partir de la herramienta definida como "fractal" en
  la interfaz de Geogebra
3
4 function arbol(p1, p2, p3, segmento, n){
5
6     var ap=ggbApplet;
7     var t, seg_reflex, pun_reflex;
8
9
10    t=ap.evalCommandGetLabels("Fractal("+p1+","+p2+","+ p3+")");
11    var arrayDeNuevos=t.split(",");
12
13    //ocultamos las etiquetas de los puntos nuevos
14    ap.evalCommand("ShowLabel("+arrayDeNuevos[0]+", "+false+")");
15    ap.evalCommand("ShowLabel("+arrayDeNuevos[1]+", "+false+")");
16
17    //calculo del grosor en función de la profundidad
18    var grosor=2*n;
19
20
21
22    //Reflejar segmento
23    seg_reflex=ap.evalCommandGetLabels("Reflect("+arrayDeNuevos[1]+", "+segmento+")");
24
25    //Reflejar punto
26    pun_reflex=ap.evalCommandGetLabels("Reflect("+arrayDeNuevos[0]+", "+segmento+")");
27    //Cambio de colores de los nuevos elementos
28    //puntos nuevos
29    ap.evalCommand("SetColor("+arrayDeNuevos[0]+",1, 0.2, 0.5)");
30
31    //cambiamos el grosor de la línea creada
32    ap.evalCommand("SetLineThickness("+arrayDeNuevos[1]+", "+grosor+")");
33    //cambiamos el grosor de la línea nueva
34    ap.evalCommand("SetLineThickness("+seg_reflex+", "+grosor+")");
35
36    n=n-1;
37    if(n==0) return 1;
38    else{
```

```

39     arbol(p2, p3, arrayDeNuevos[0], arrayDeNuevos[1], n);
40     arbol(p2, p3, pun_reflex, seg_reflex, n);
41     return 1;
42 }
43
44
45 }

```

Listing 18: Función árbol para la creación de árboles fractales en Geogebra

```

1
2 function ggbOnInit() {
3     //al arrancar le asignamos al punto A la función "palometear"
4 var ap=ggbApplet;
5 ap.registerObjectClickListener("A", "palometear");
6 alert("todo en orden");
7
8 }
9 //PALOMETEAR
10 //cuando se pulse un objeto, aplicará la herramienta "palometear2" y luego hará que
    los puntos creados
11 //la usen también
12 function palometear(objeto){
13     var ap=ggbApplet;
14     var segment=ap.getValue("segment");
15     var t;
16     t=ap.evalCommandGetLabels("palometas2("+objeto+", "+segment+"");
17     //le aplicamos lo mismo a los nuevos puntos creados
18     var arrayDeNuevos=t.split(",");
19     ap.registerObjectClickListener(arrayDeNuevos[0], "palometear");
20     ap.registerObjectClickListener(arrayDeNuevos[2], "palometear");
21
22     //ap.evalCommand("ShowLabel("+arrayDeNuevos[0]+", "+true+"");
23 //ap.evalCommand("ShowLabel("+arrayDeNuevos[2]+", "+true+"");
24
25     //ap.evalCommand("palometas2("+objeto+", "+segment+"");
26 }

```

Listing 19: Función para crear "Listeners" en objetos recién creados