

Trigonometric Identi... Natural minor (Aeoli... C++ con Clase - Pro...

The page at www.codeskulptor.org says:

Introducir forma: numerica o cis (para salir (s)):

n|

OK Cancel

Docs

```
12 #Creando clase numero cuyo metodo de inici
13 #una mayor conveniencia de operar para los
14 #de dos numeros complejos en su forma polar es igual al pro
15 #Tambien se incluyen dos metodos dentro de la clase para re
16 #perteneiente a la clase Cis que toma como argumentos modu
17 class Numero:
18
19     def __init__(self, parte_real, parte_imaginaria):
20         self.parte_real=parte_real
21         self.parte_imaginaria=parte_imaginaria
22
23     def __str__(self):
24         return str(self.parte_real)+" "+str(self.parte_ima
25     #método que obtiene el módulo
26     def mod(self):
27         self.m=math.sqrt(math.pow(self.parte_real,2)+math.p
28         return self.m
29     #método que obtiene el ángulo
30     def ang(self):
31         self.angle=math.degrees(math.atan2(self.parte_imagi
32         return self.angle
33
34     def num_a_cis(self):
35         return str(self.m)+"cis"+str(self.angle)
36     #método que efectúa la suma de números complejos
37     def suma(self, a, b):
38         self.parte_real+=a
39         self.parte_imaginaria+=b
40     #método que efectúa la resta de números complejos
41     def resta (self, a, b):
42         self.parte_real-=a
```

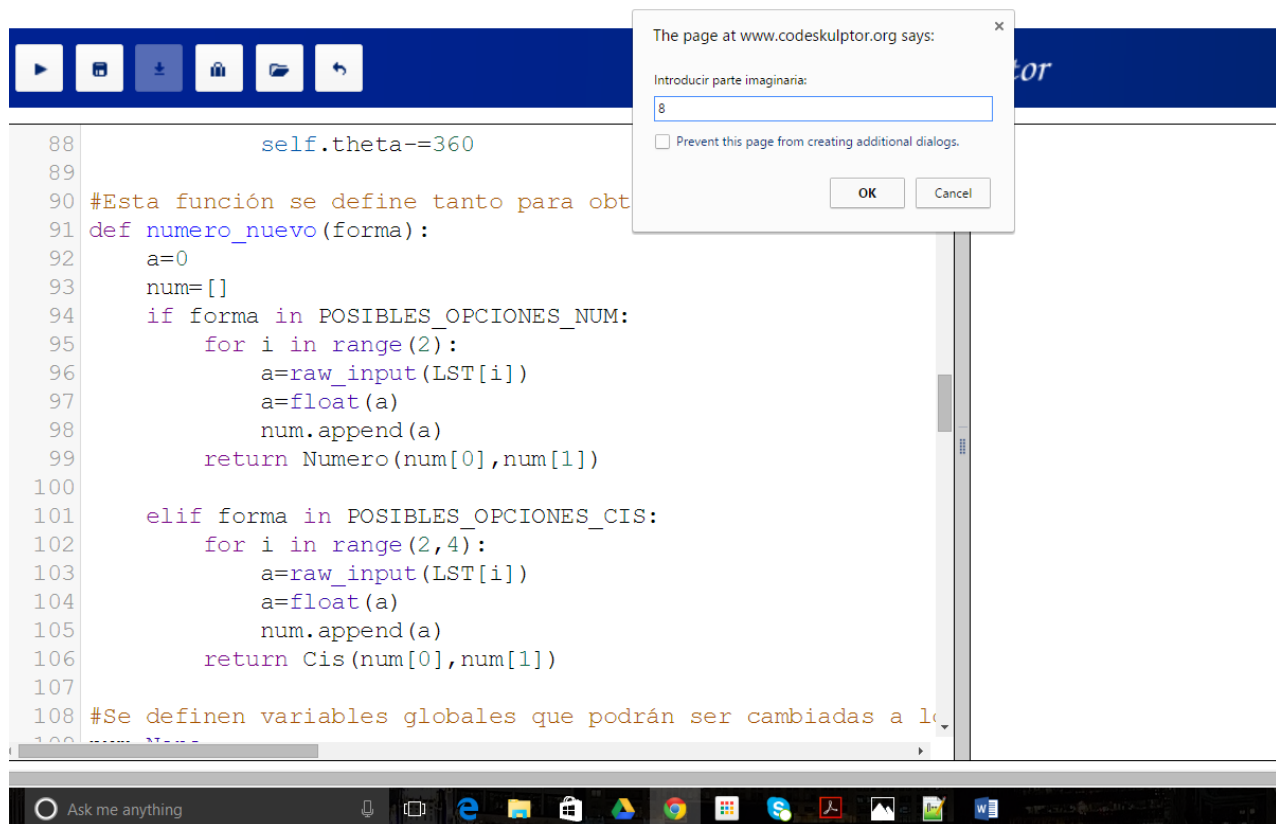
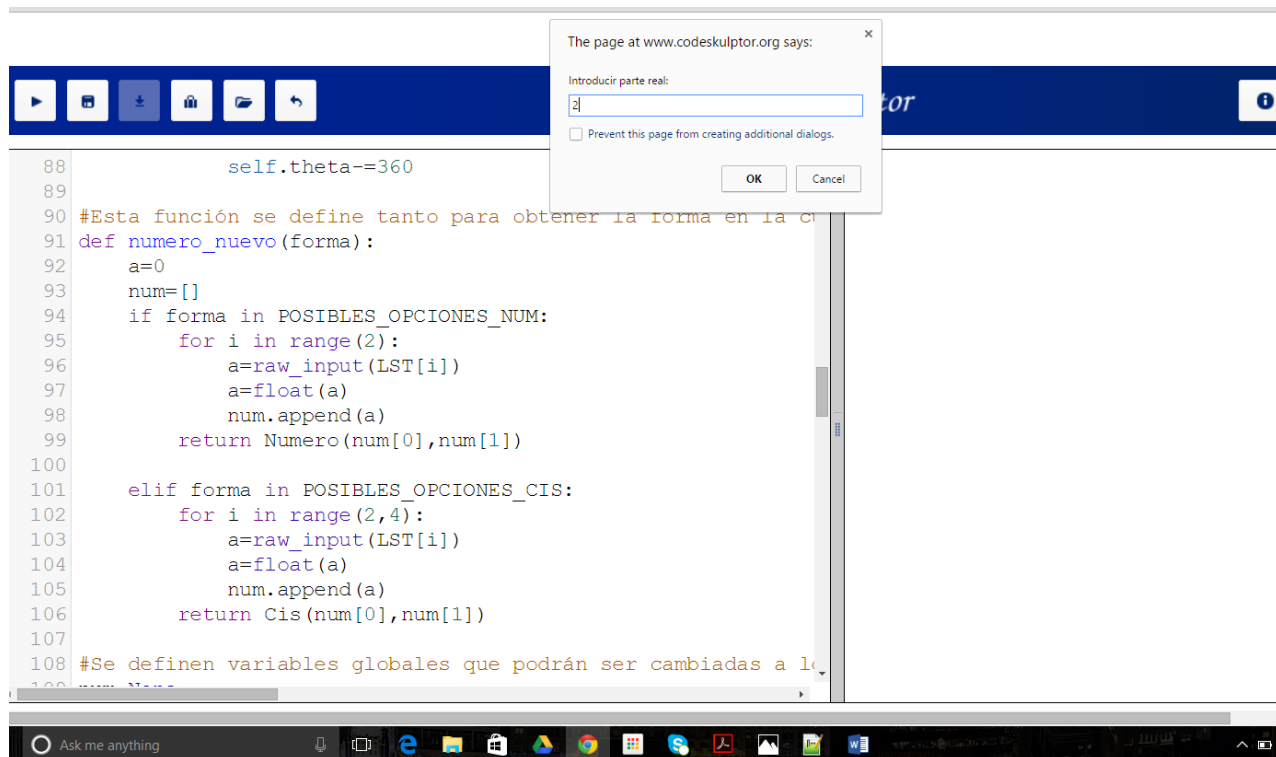


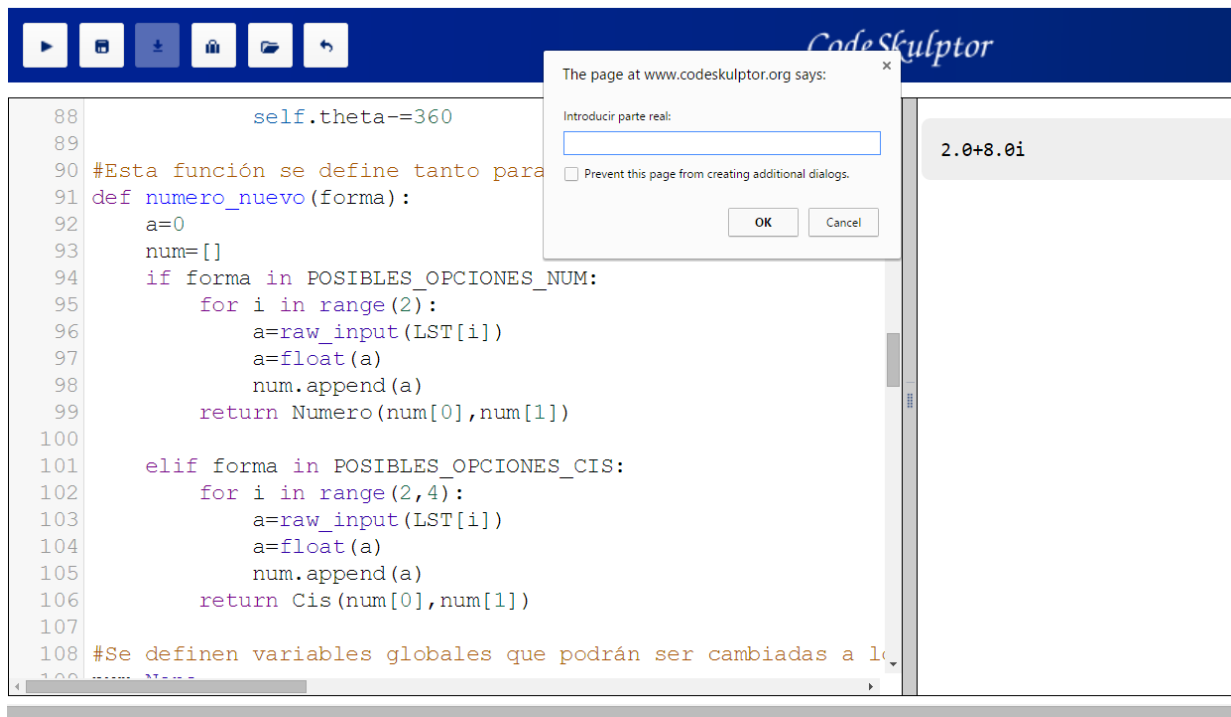
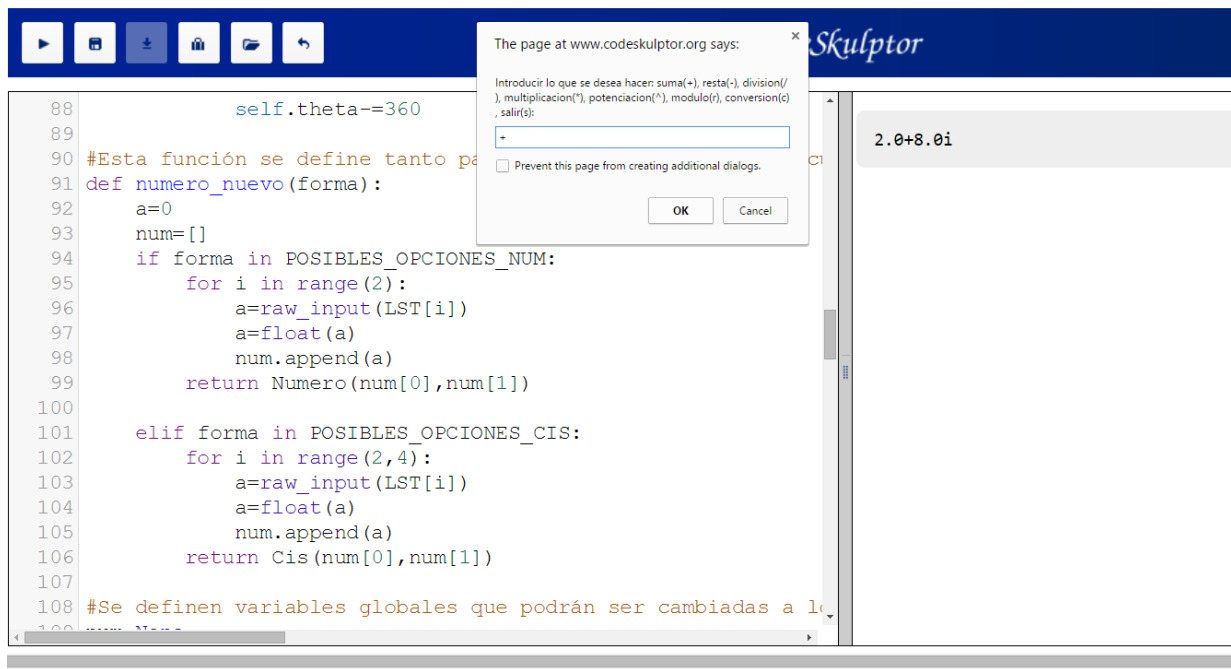
The page at www.codeskulptor.org says:

Introducir forma: numerica o cis (para salir (s)):

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto para obtener la forma en la c
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a l
109 num_Nuevo
```





CodeSkulptor

The page at www.codeskulptor.org says:

Introducir parte real:

5

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto para
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
```

2.0+8.0i

CodeSkulptor

The page at www.codeskulptor.org says:

Introducir parte imaginaria:

7

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto para
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
```

2.0+8.0i

The page at www.codeskulptor.org says:

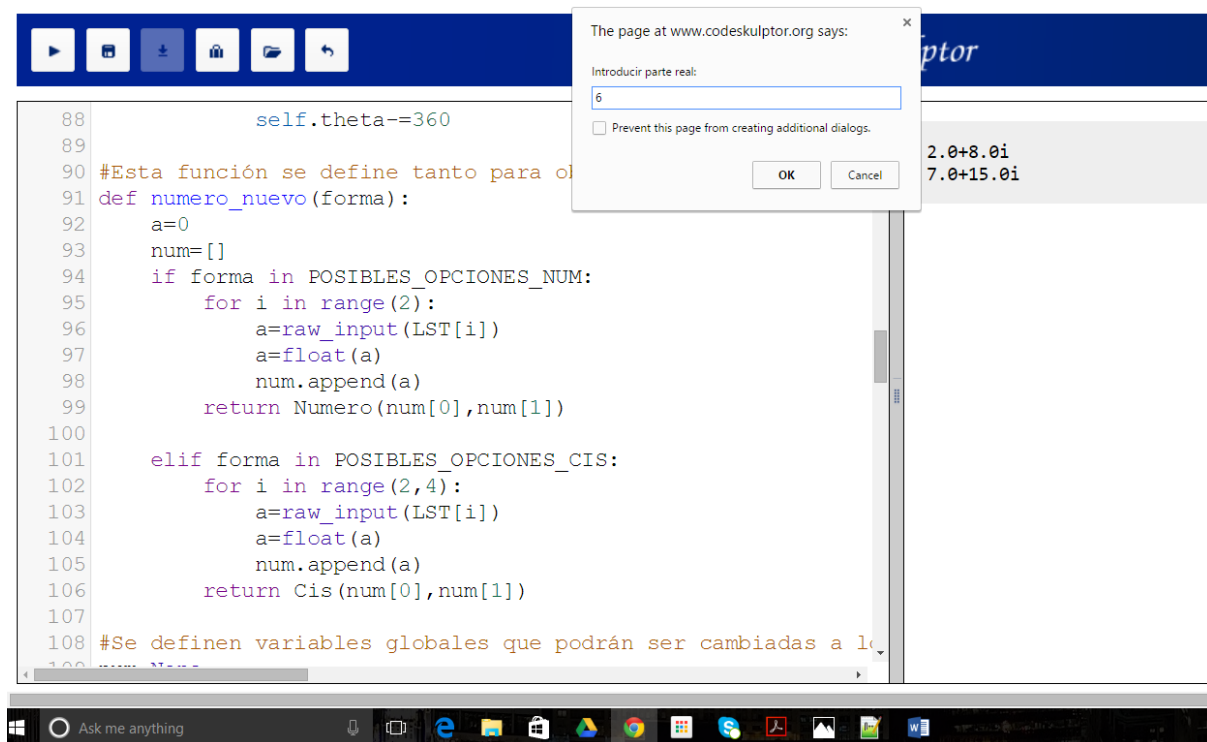
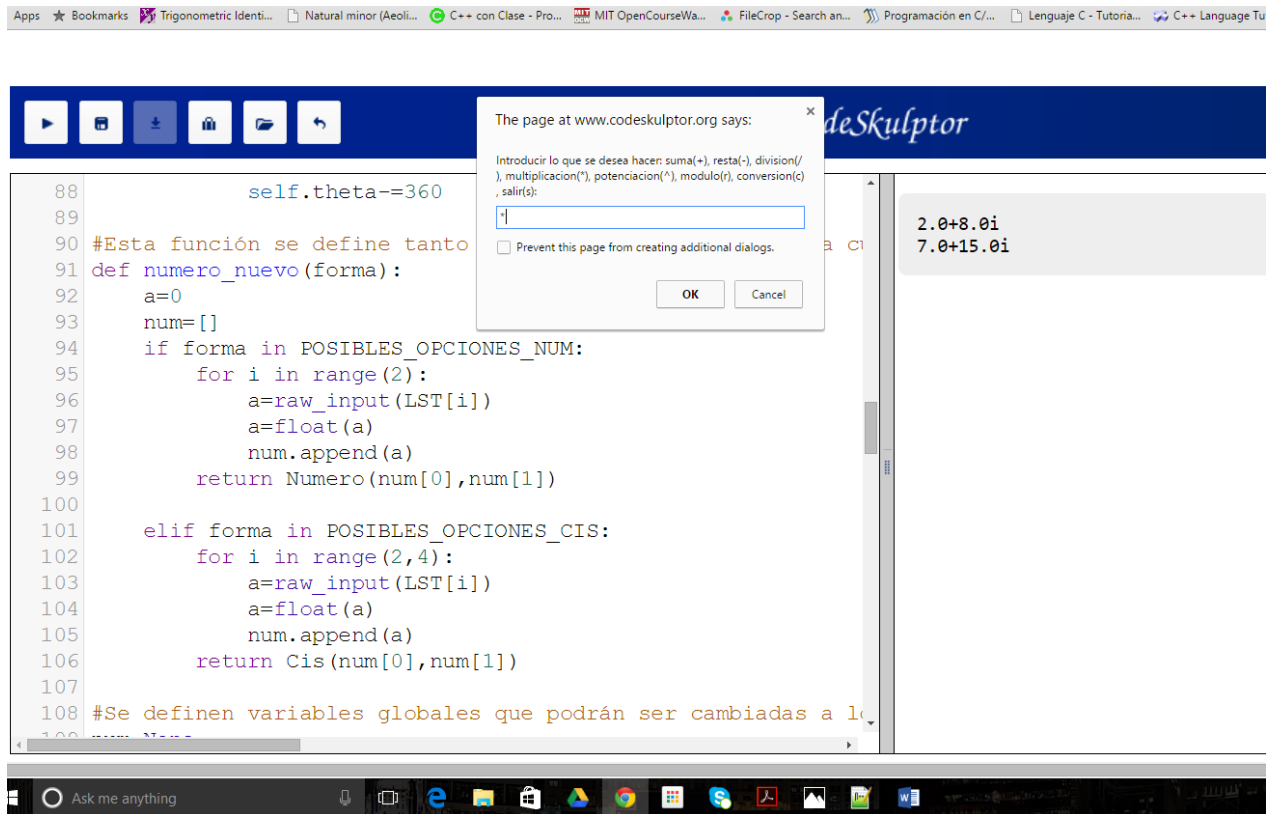
Introducir lo que se desea hacer: suma(+), resta(-), division(/), multiplicacion(\*), potenciacion(^), modulo(r), conversion(c), salir(s):

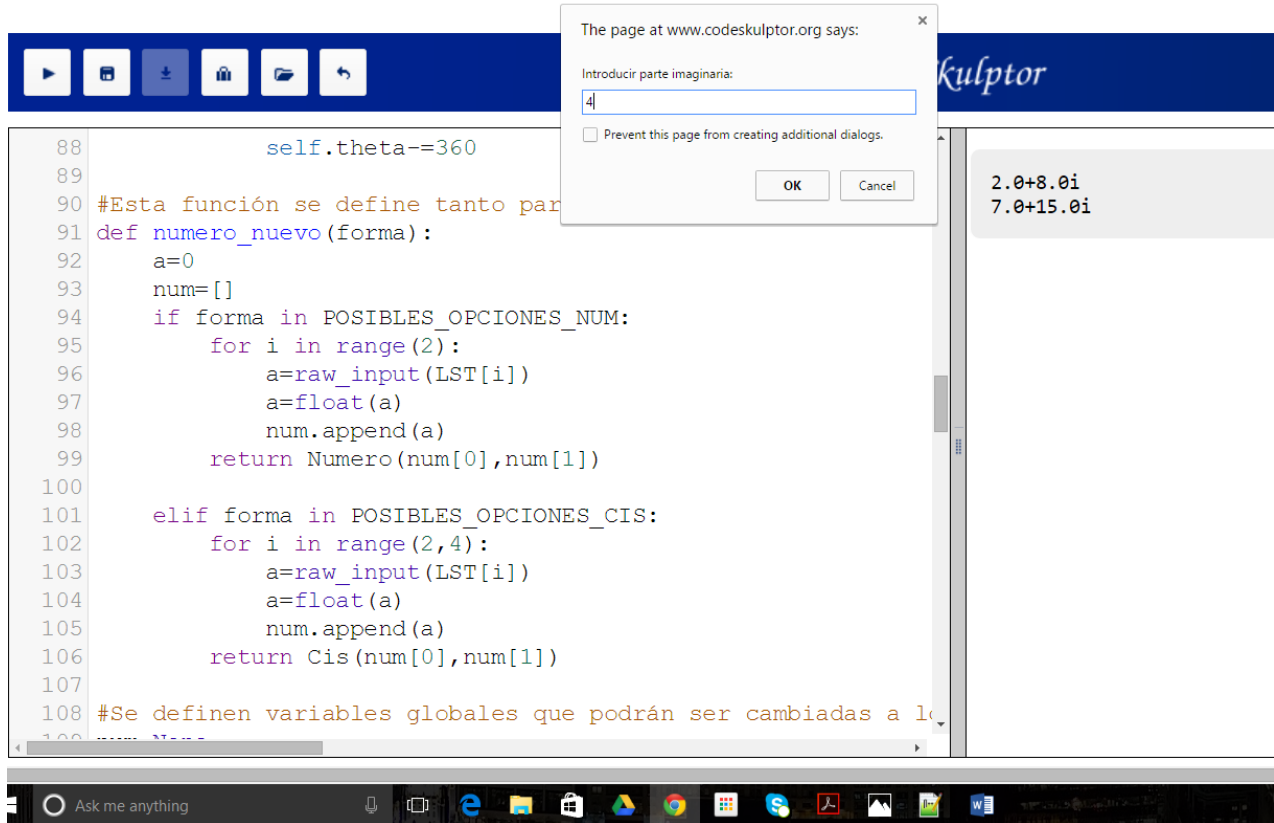
☐ Prevent this page from creating additional dialogs.

OK Cancel

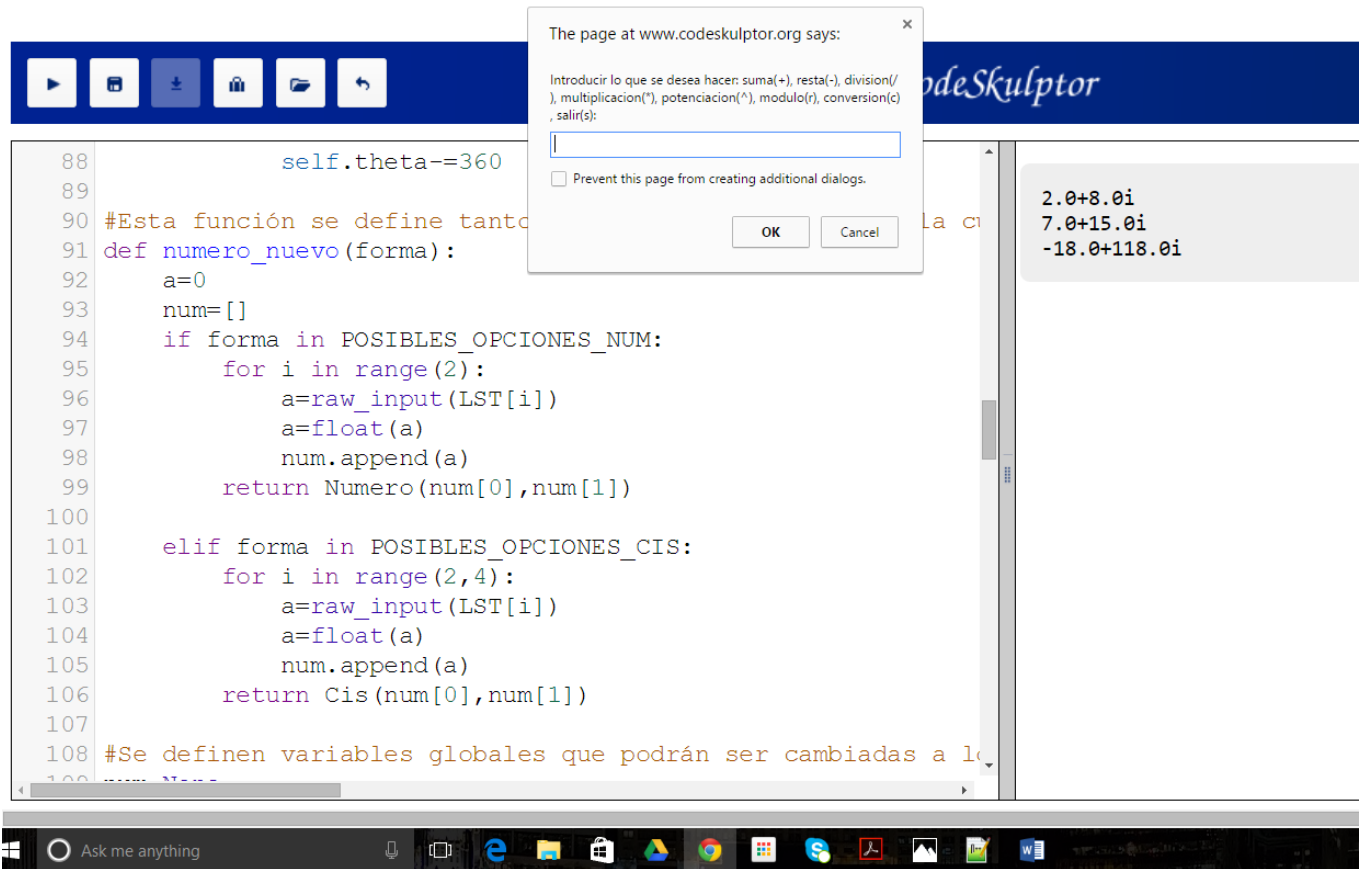
```
88         self.theta-=360
89
90 #Esta función se define tanto
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99         return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106         return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo largo del programa
109 num = Numero(2,0)
110 cis = Cis(7,0)
```

2.0+8.0i  
7.0+15.0i









The page at www.codeskulptor.org says:

Introducir lo que se desea hacer: suma(+), resta(-), division(/), multiplicacion(\*), potenciacion(^), modulo(r), conversion(c), salir(s):

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
109 num_Nuevo
```

2.0+8.0i  
7.0+15.0i  
-18.0+118.0i

The page at www.codeskulptor.org says:

Introducir exponente:

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto p
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
109 num_Nuevo
```

2.0+8.0i  
7.0+15.0i  
-18.0+118.0i

The page at www.codeskulptor.org says:

Introducir lo que se desea hacer: suma(+), resta(-), division(/), multiplicacion(\*), potenciacion(^), modulo(r), conversion(c), salir(s):

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
```

2.0+8.0i  
7.0+15.0i  
-18.0+118.0i  
6.21478919485e+28+-1.01717013496e+29i

The page at www.codeskulptor.org says:

Introducir lo que se desea hacer: suma(+), resta(-), division(/), multiplicacion(\*), potenciacion(^), modulo(r), conversion(c), salir(s):

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
```

2.0+8.0i  
7.0+15.0i  
-18.0+118.0i  
6.21478919485e+28+-1.01717013496e+29i

The page at www.codeskulptor.org says:

Introducir lo que se desea hacer: suma(+), resta(-), division(/), multiplicacion(\*), potenciacion(^), modulo(r), conversion(c), salir(s):

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
```

2.0+8.0i  
7.0+15.0i  
-18.0+118.0i  
6.21478919485e+28+-1.01717013496e+29i

The page at www.codeskulptor.org says:

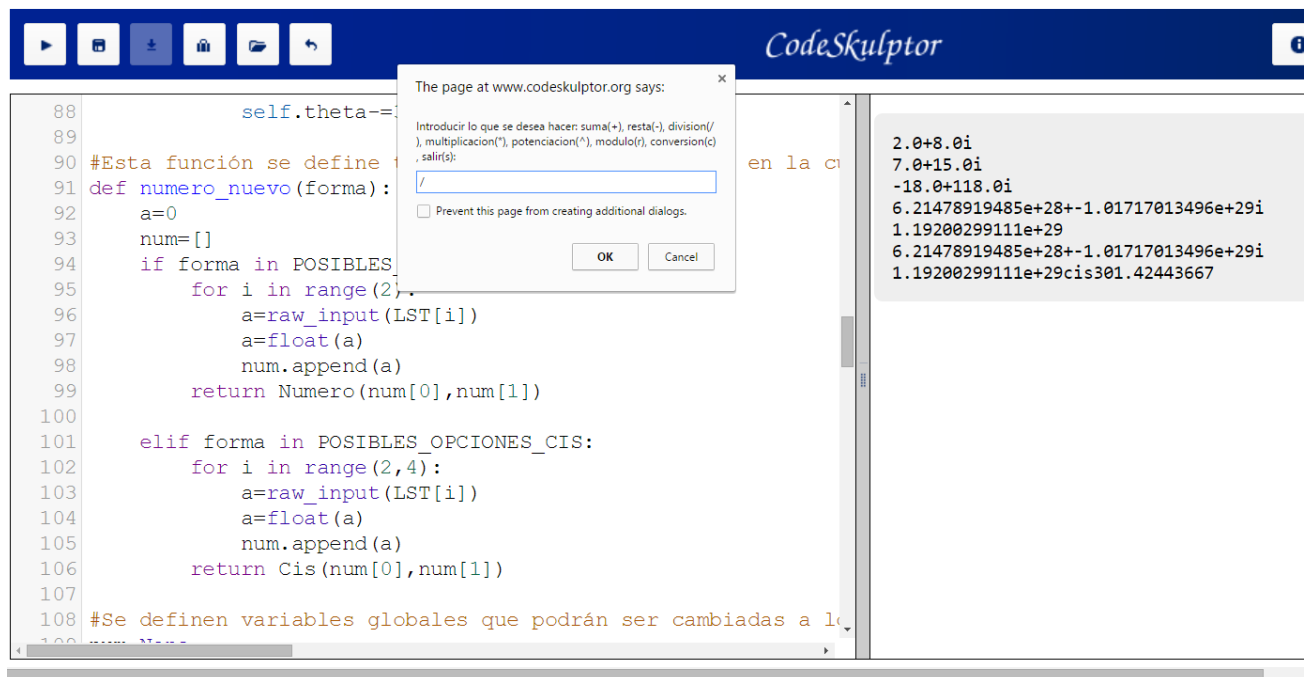
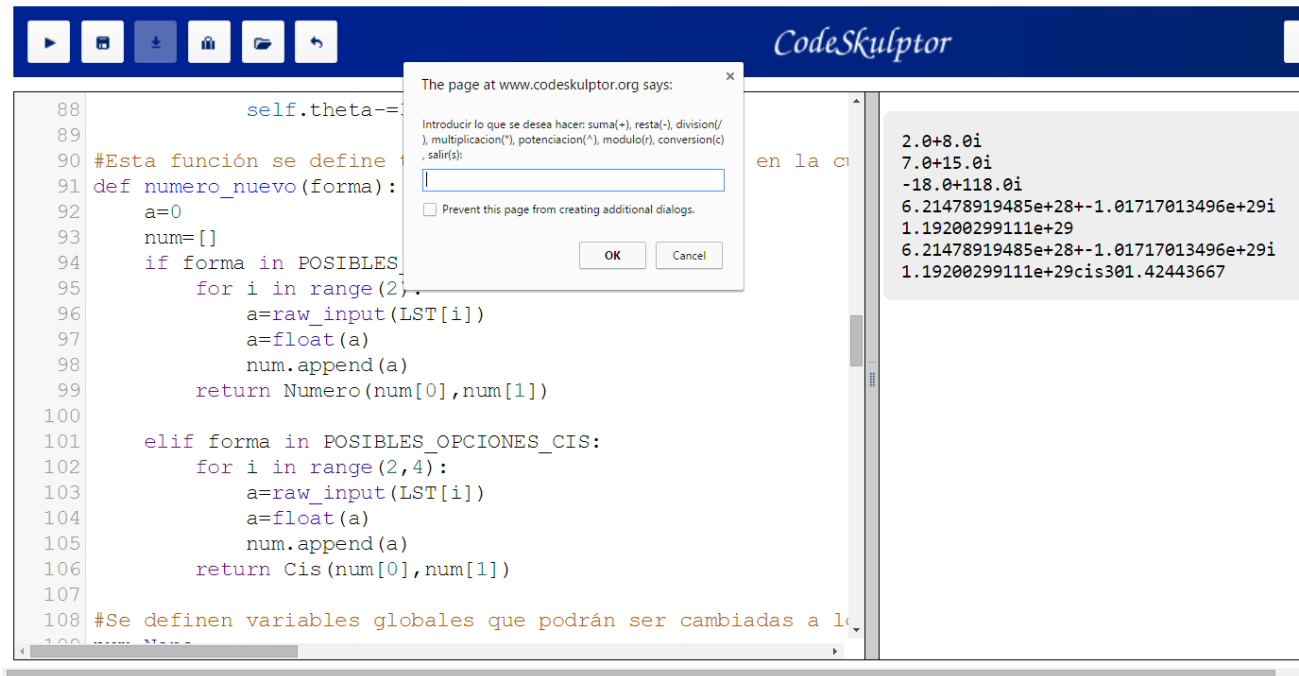
Introducir lo que se desea hacer: suma(+), resta(-), division(/), multiplicacion(\*), potenciacion(^), modulo(r), conversion(c), salir(s):

☐ Prevent this page from creating additional dialogs.

OK Cancel

```
88         self.theta-=360
89
90 #Esta función se define tanto
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIONES_NUM:
95         for i in range(2):
96             a=raw_input(LST[i])
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES_CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis(num[0],num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
```

2.0+8.0i  
7.0+15.0i  
-18.0+118.0i  
6.21478919485e+28+-1.01717013496e+29i  
1.19200299111e+29  
6.21478919485e+28+-1.01717013496e+29i



The page at www.codeskulptor.org says:

Introducir modulo:

☐ Prevent this page from creating additional dialogs.

OK

Cancel

88self.theta-=360

89

90#Esta función se define tanto para números como para expresiones complejas

91def numero\_nuevo(forma):

92a=0

93num=[]

94if forma in POSIBLES OPCIONES\_NUM:

95for i in range(2):

96a=raw\_input(LST[i])

97a=float(a)

98num.append(a)

99return Numero(num[0],num[1])

100

101elif forma in POSIBLES OPCIONES\_CIS:

102for i in range(2,4):

103a=raw\_input(LST[i])

104a=float(a)

105num.append(a)

106return Cis(num[0],num[1])

107

108#Se definen variables globales que podrán ser cambiadas a lo largo del programa

109num\_Nuevo=Numero(0,0)

110

2.0+8.0i

7.0+15.0i

-18.0+118.0i

6.21478919485e+28+-1.01717013496e+29i

1.19200299111e+29

6.21478919485e+28+-1.01717013496e+29i

1.19200299111e+29cis301.42443667

The page at www.codeskulptor.org says:

Introducir modulo:

20000000000

☐ Prevent this page from creating additional dialogs.

OK

Cancel

88self.theta-=360

89

90#Esta función se define tanto para números como para expresiones complejas

91def numero\_nuevo(forma):

92a=0

93num=[]

94if forma in POSIBLES OPCIONES\_NUM:

95for i in range(2):

96a=raw\_input(LST[i])

97a=float(a)

98num.append(a)

99return Numero(num[0],num[1])

100

101elif forma in POSIBLES OPCIONES\_CIS:

102for i in range(2,4):

103a=raw\_input(LST[i])

104a=float(a)

105num.append(a)

106return Cis(num[0],num[1])

107

108#Se definen variables globales que podrán ser cambiadas a lo largo del programa

109num\_Nuevo=Numero(0,0)

110

2.0+8.0i

7.0+15.0i

-18.0+118.0i

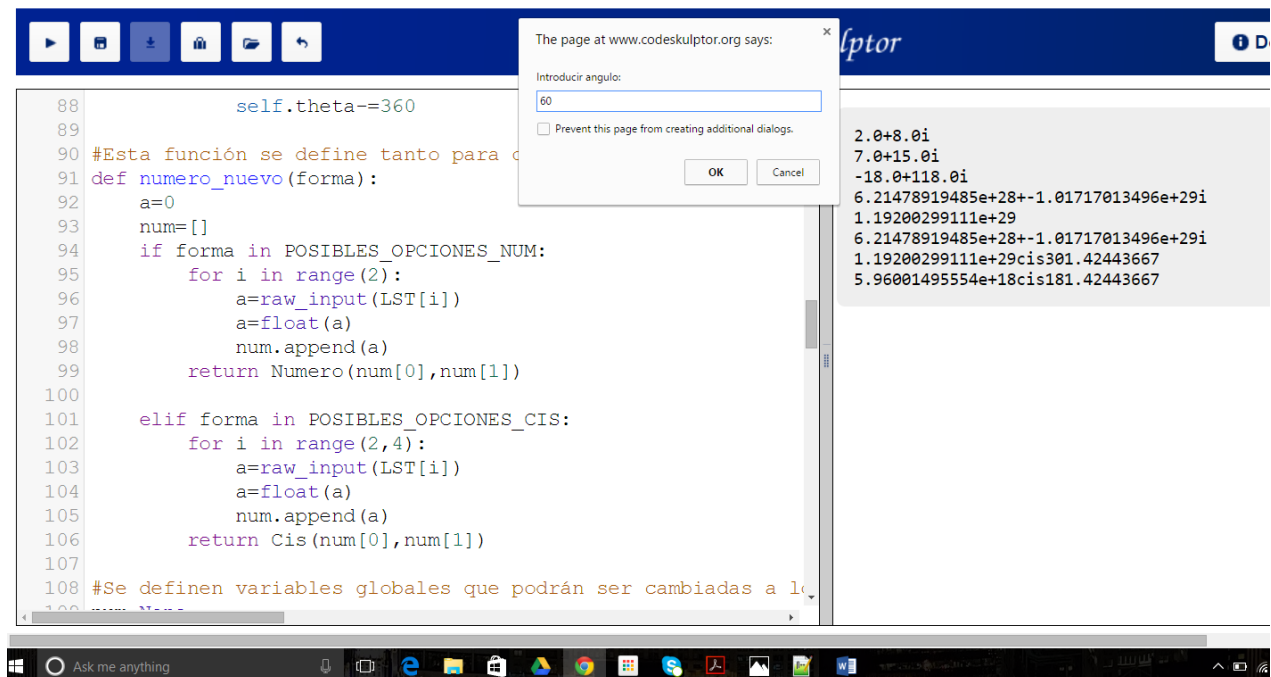
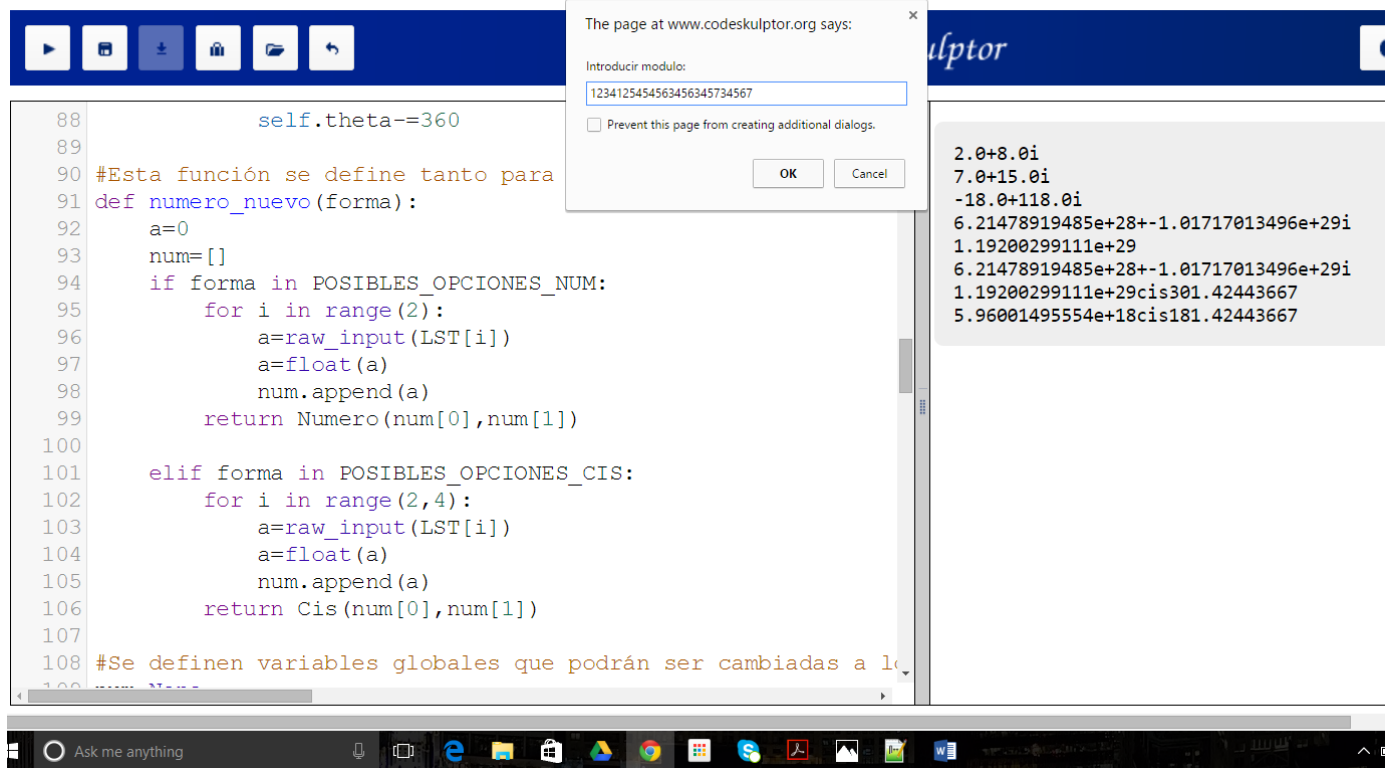
6.21478919485e+28+-1.01717013496e+29i

1.19200299111e+29

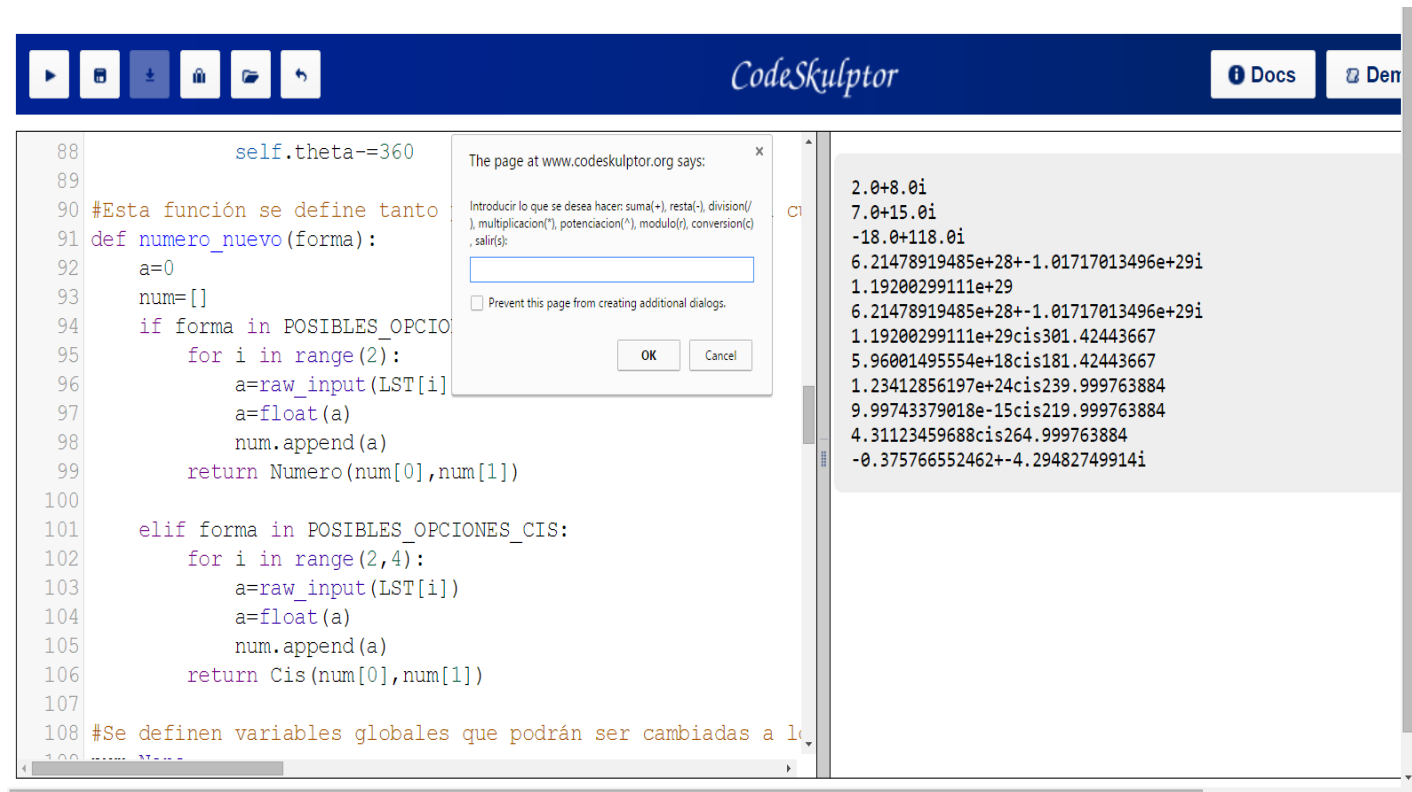
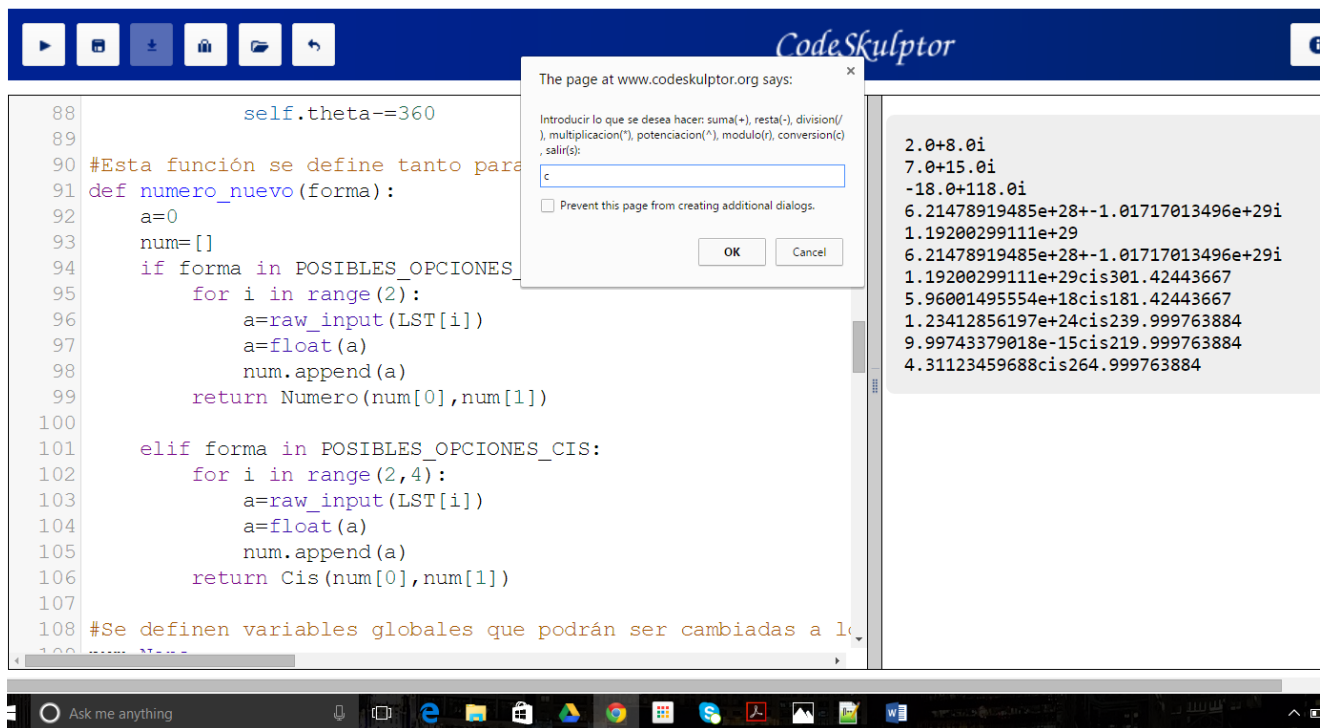
6.21478919485e+28+-1.01717013496e+29i

1.19200299111e+29cis301.42443667









The screenshot shows the CodeSkulptor web interface. The left pane contains Python code for a calculator. The right pane shows the output of the code. A dialog box is open in the center.

```
88         self.theta-=360
89
90 #Esta función se define tanto
91 def numero_nuevo(forma):
92     a=0
93     num=[]
94     if forma in POSIBLES OPCIO
95         for i in range(2):
96             a=raw_input(LST[i]
97             a=float(a)
98             num.append(a)
99             return Numero(num[0],num[1])
100
101     elif forma in POSIBLES OPCIONES CIS:
102         for i in range(2,4):
103             a=raw_input(LST[i])
104             a=float(a)
105             num.append(a)
106             return Cis (num[0], num[1])
107
108 #Se definen variables globales que podrán ser cambiadas a lo
109 num = None
```

The page at www.codeskulptor.org says:

Introducir lo que se desea hacer: suma(+), resta(-), division(/), multiplicacion(\*), potenciacion(^), modulo(r), conversion(c), salir(s):

☐ Prevent this page from creating additional dialogs.

OK Cancel

2.0+8.0i  
7.0+15.0i  
-18.0+118.0i  
6.21478919485e+28+-1.01717013496e+29i  
1.19200299111e+29  
6.21478919485e+28+-1.01717013496e+29i  
1.19200299111e+29cis301.42443667  
5.96001495554e+18cis181.42443667  
1.23412856197e+24cis239.999763884  
9.99743379018e-15cis219.999763884  
4.31123459688cis264.999763884  
-0.375766552462+-4.29482749914i

Enlace para entrar al programa en el intérprete en línea

[http://www.codeskulptor.org/#user40\\_pjo576dBJ5\\_1.py](http://www.codeskulptor.org/#user40_pjo576dBJ5_1.py)

Bonus:

Juego efectuado para clase en línea de programación interactiva:

[http://www.codeskulptor.org/#user40\\_pjo576dBJ5\\_2.py](http://www.codeskulptor.org/#user40_pjo576dBJ5_2.py)

Si quieres divertirte un rato programando juegos, se pueden crear muchas cosas con codeskulptor.org