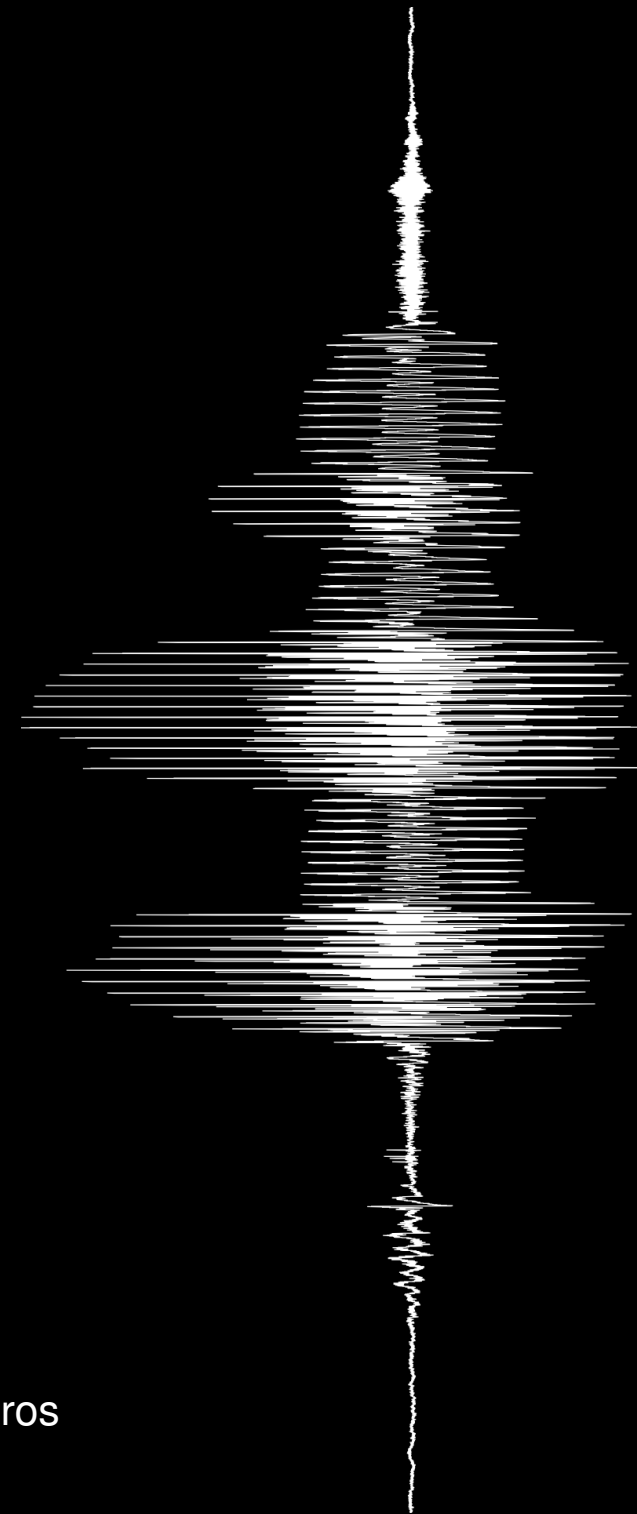


scripting en praat

sesión n.1 fundamentos de programación

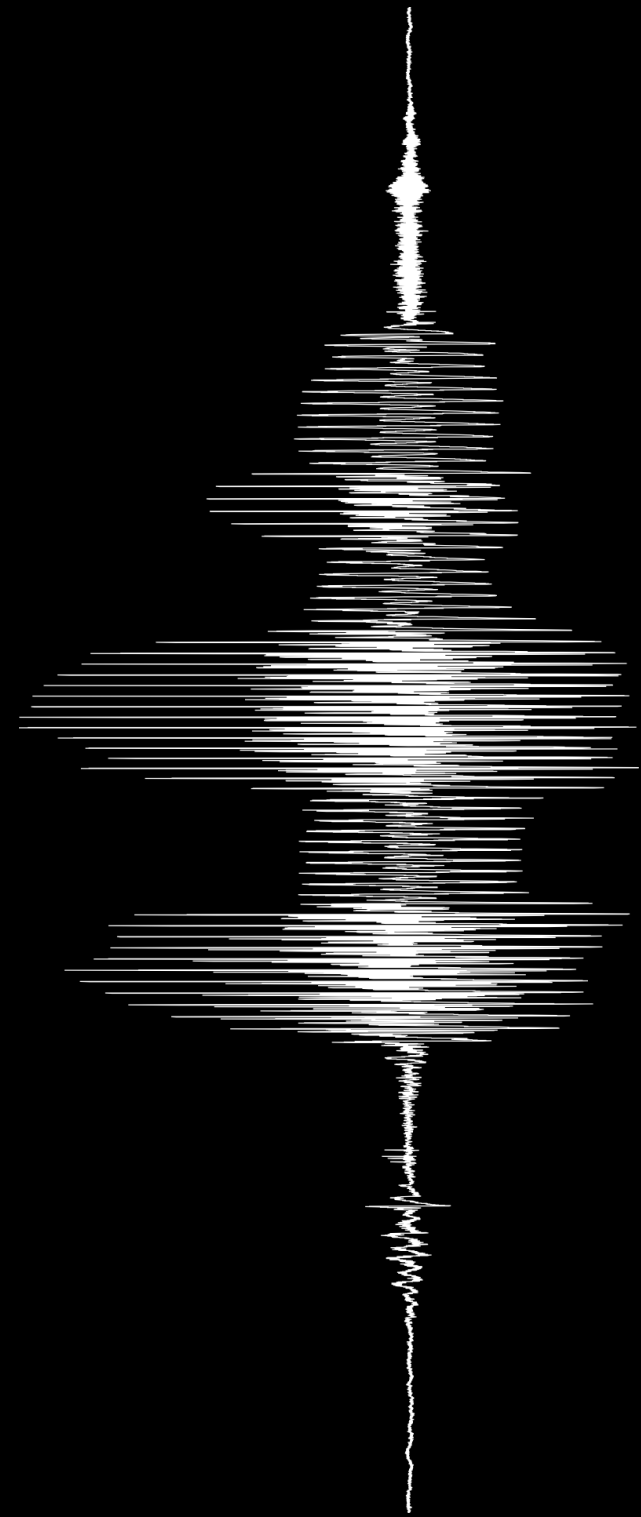


José Joaquín ATRIA Universidad de Tokyo de Estudios Extranjeros

jjatria@gmail.com www.pinguinorodriguez.cl

parte 1

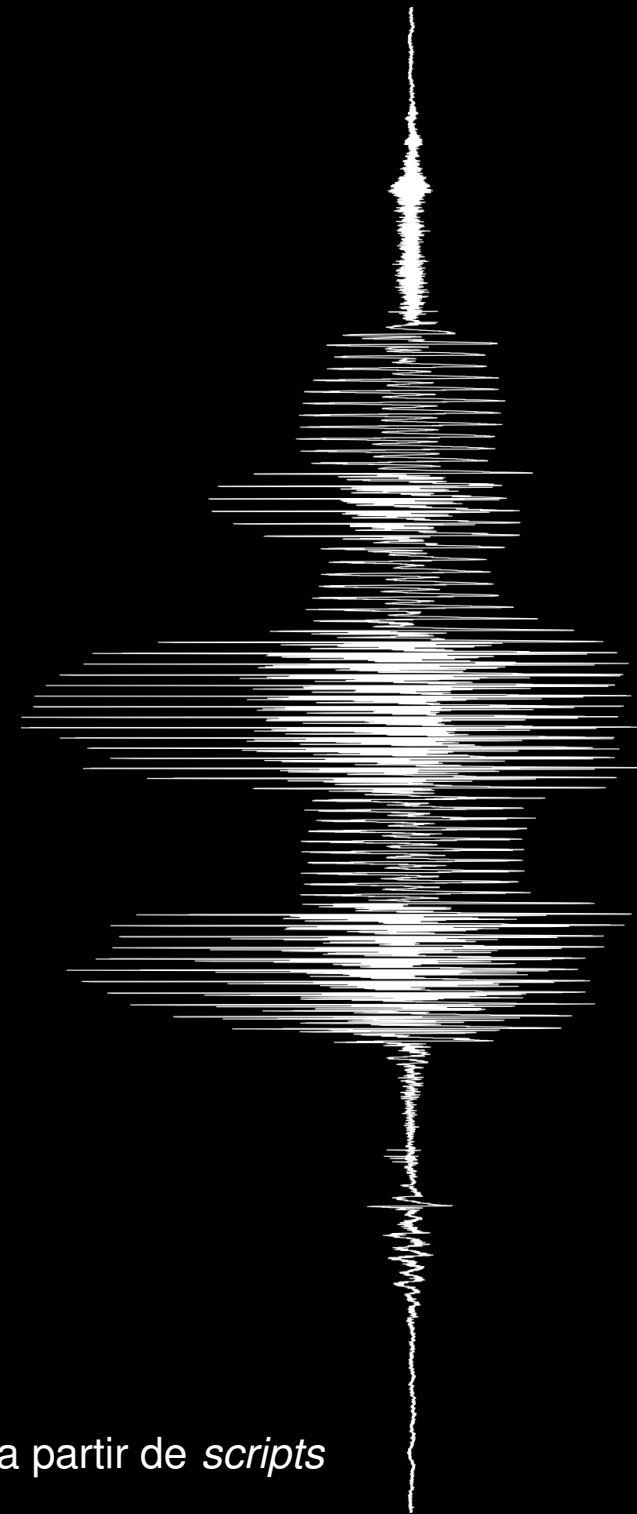
aspectos generales



¿qué es un *script*?

- es una serie de instrucciones
- es un programa
- es interpretado, no compilado*

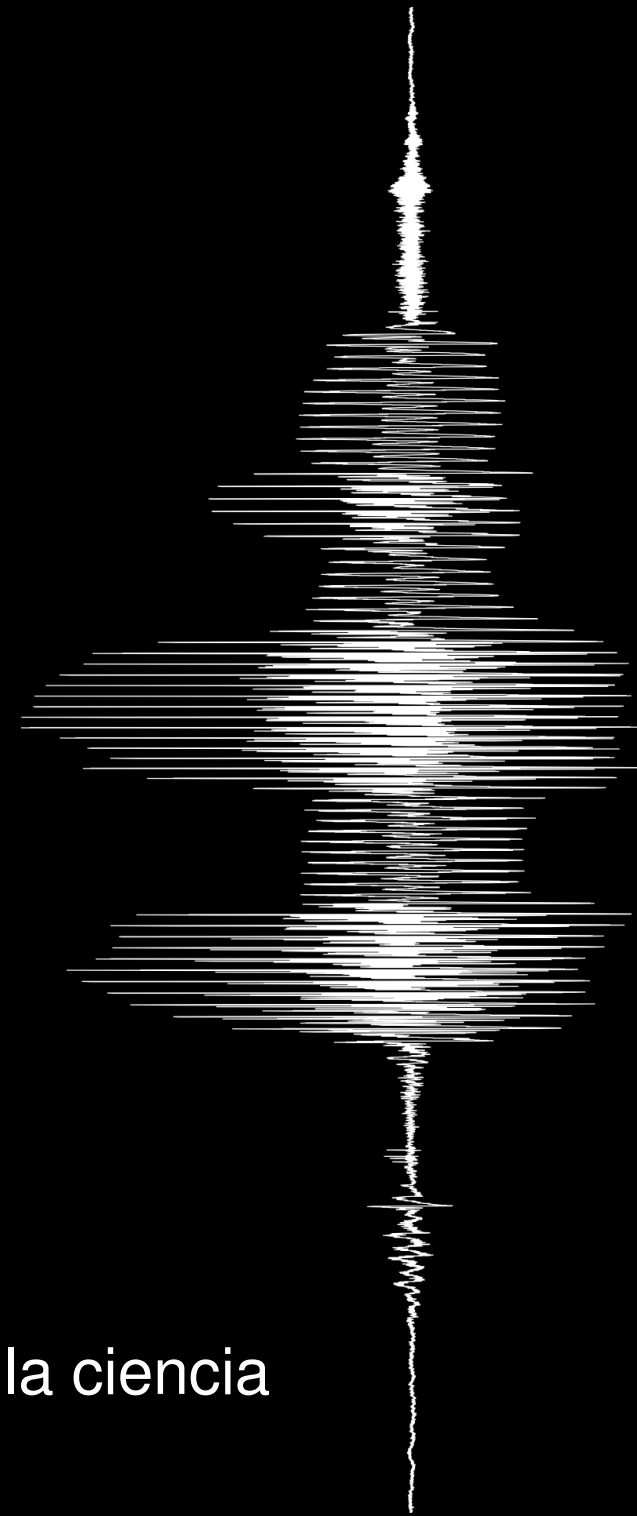
* en versiones recientes de praat ($\geq 5.1.43$), es posible generar programas a partir de *scripts*



¿por qué usar *scripts*?

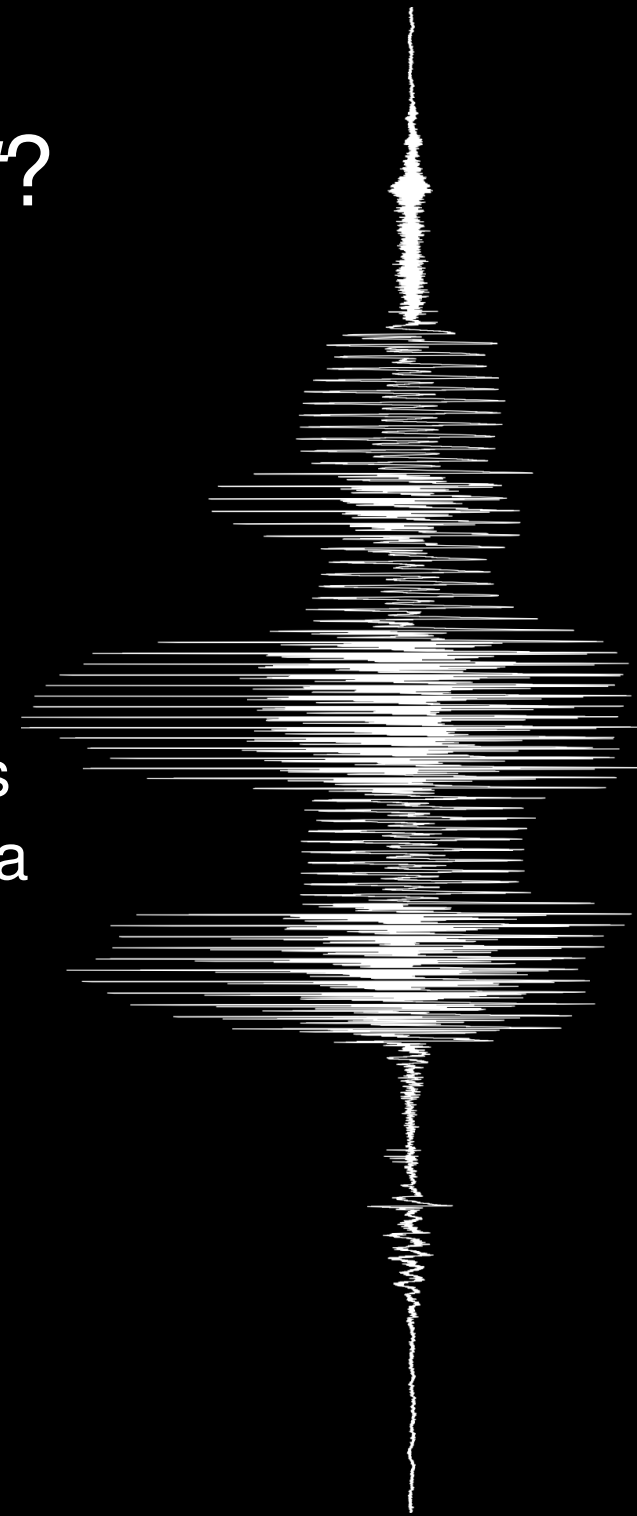
- son precisos
- son reproducibles
- son automáticos
- son portables

... y todo eso los hace los mejores amigos de la ciencia



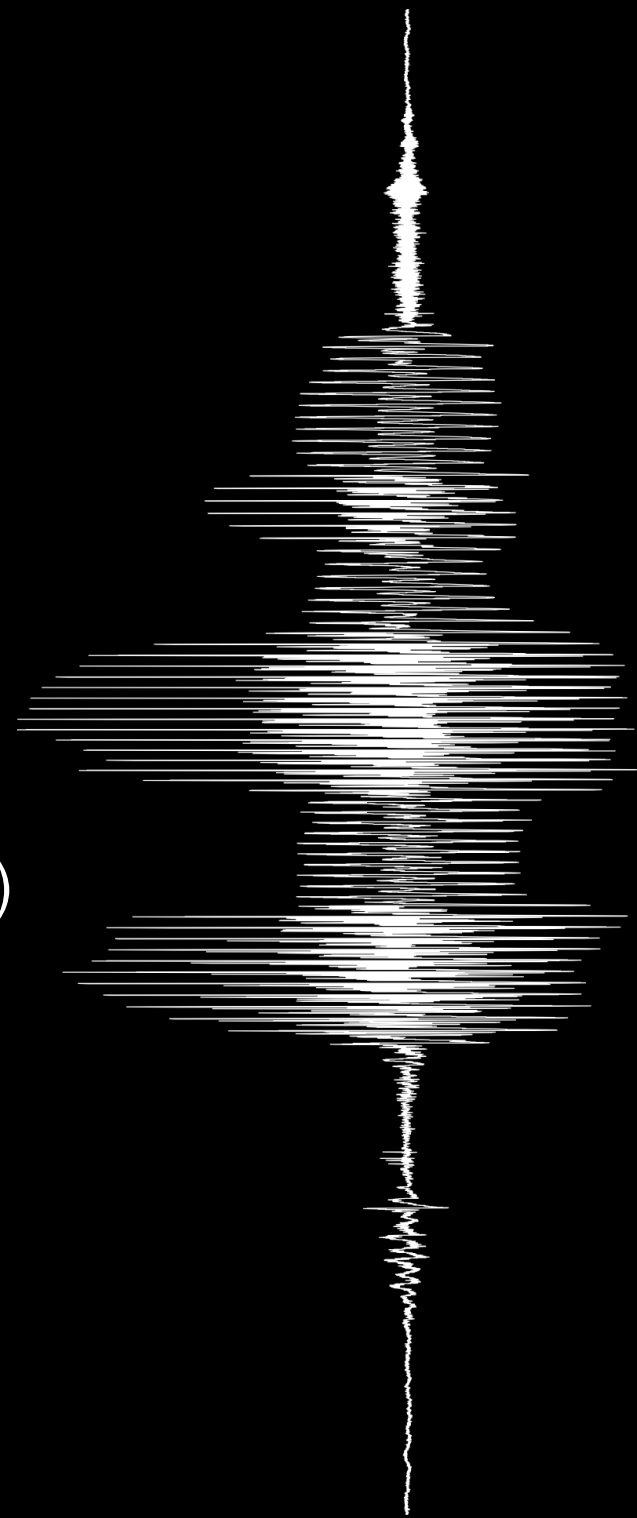
¿qué puedo hacer con un *script*?

- todo lo que puedo hacer sin *scripts*
... y más
- automatizar tareas tediosas
como manipular, analizar y editar grandes cantidades de archivos, todos de la misma manera
- invocar otros *scripts*
- modificar el comportamiento de praat
- etc...



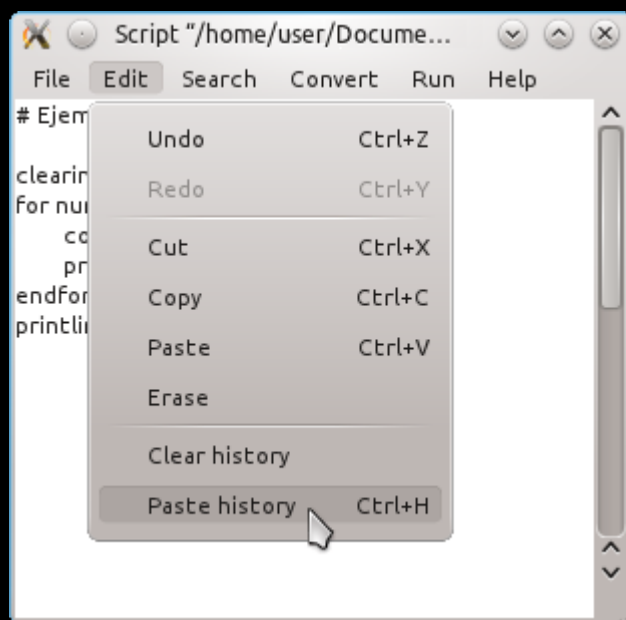
¿cómo se hace un *script*?

- con un editor de texto
 - en Windows, [Notepad++](#) es el mejor
 - en GNU/Linux hay debate (como siempre)
 - en Mac... no sé (sugerencias?)
 - praat incluye uno propio, interno

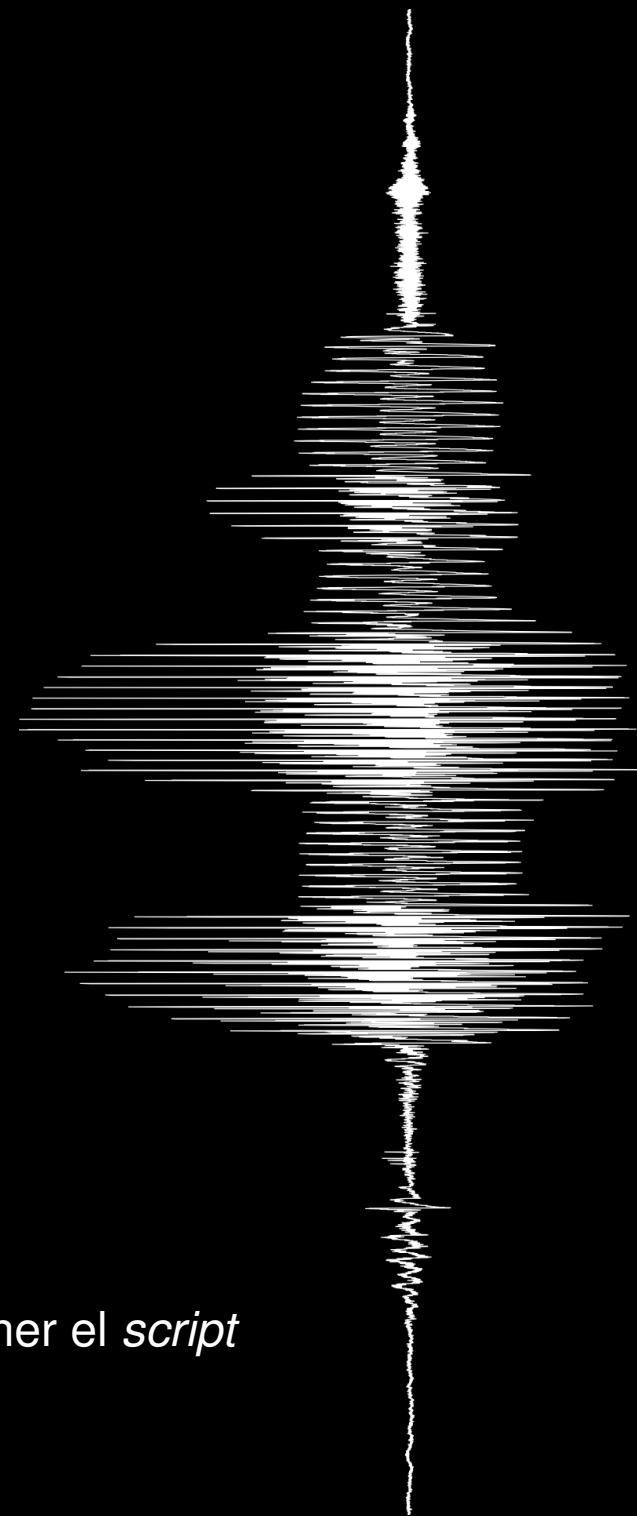


¿cómo se hace un *script*?

- con el historial de acciones de praat

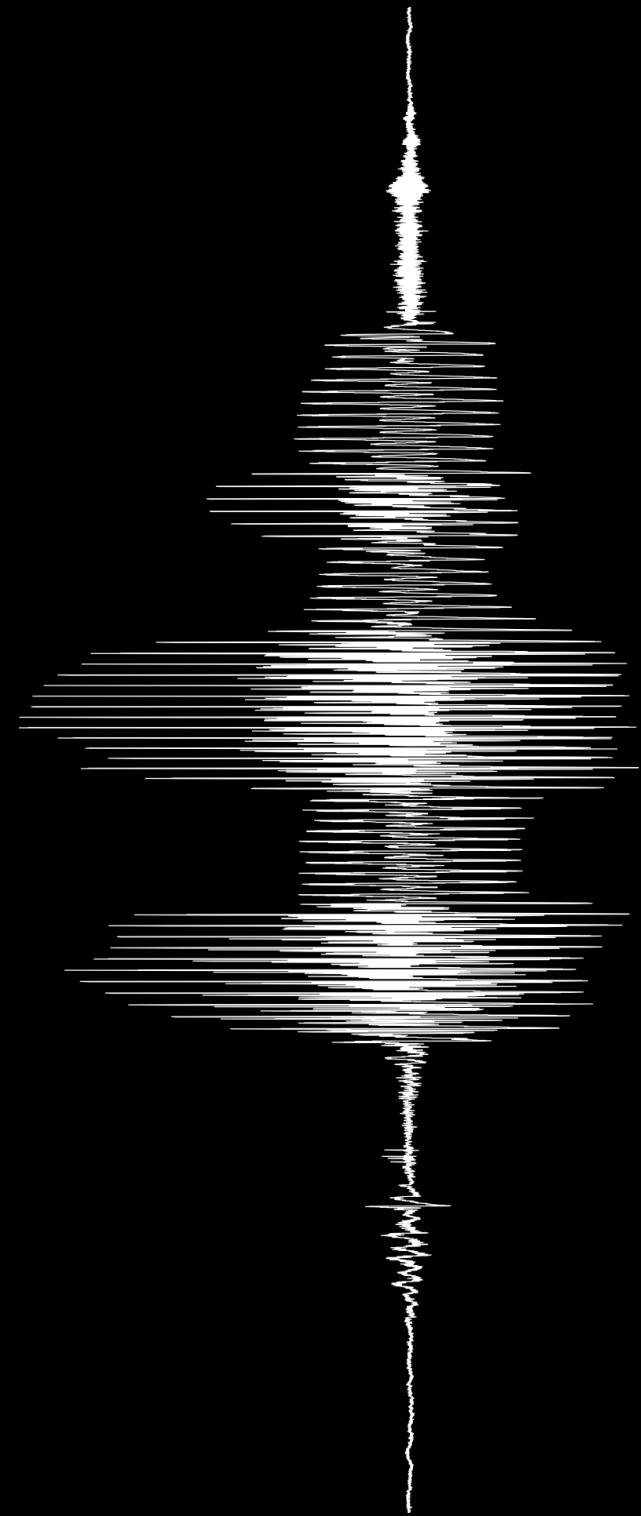


que a veces hace que ni siquiera sea necesario escribir para tener el *script*



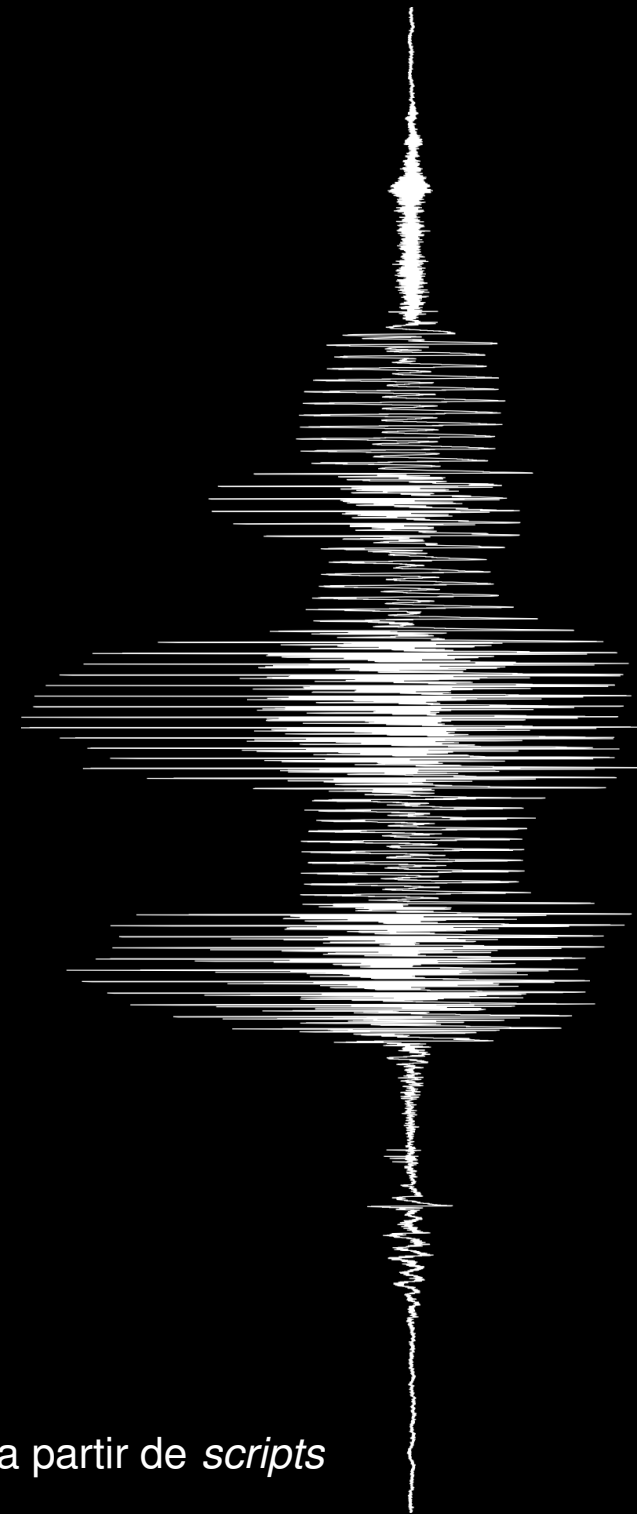
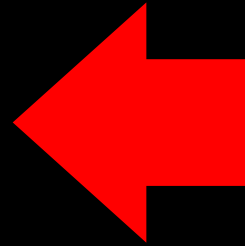
¿cómo se hace un *script*?

¡pero eso NO es todo!



¿qué es un *script*?

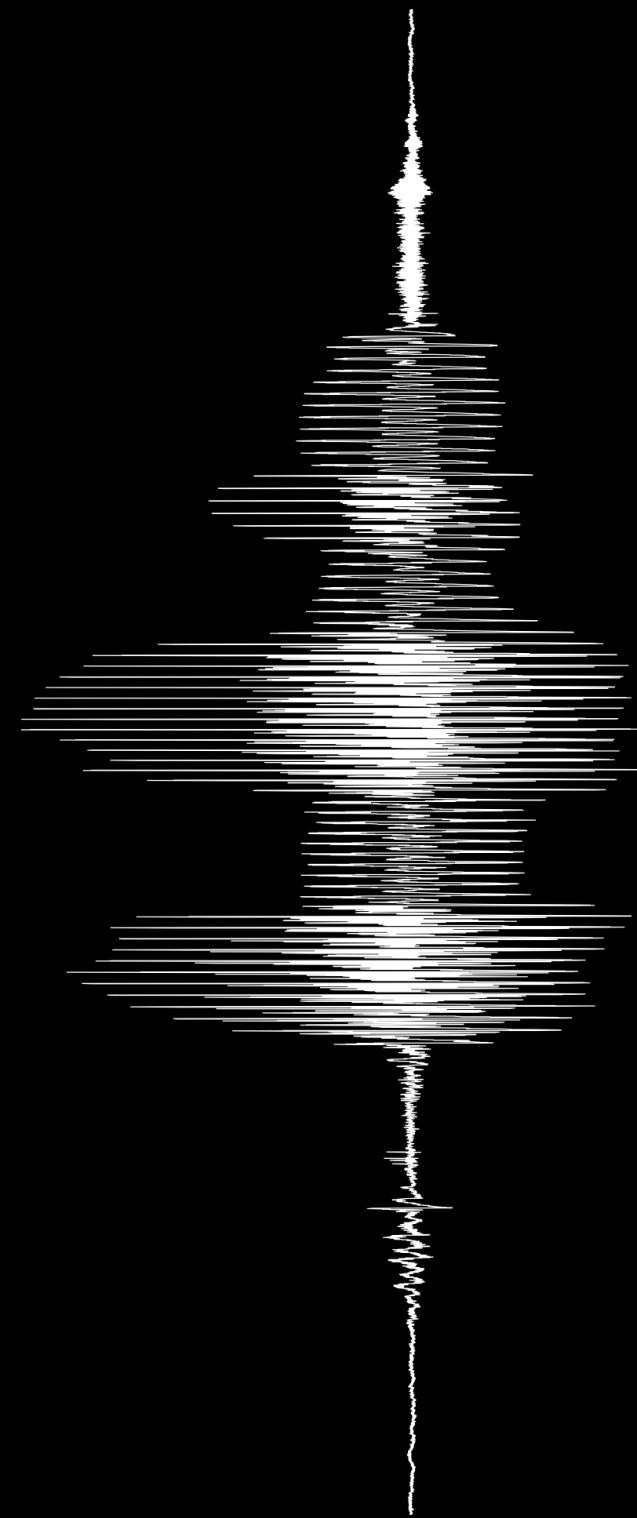
- es una serie de instrucciones
- es un programa
- es interpretado, no compilado*



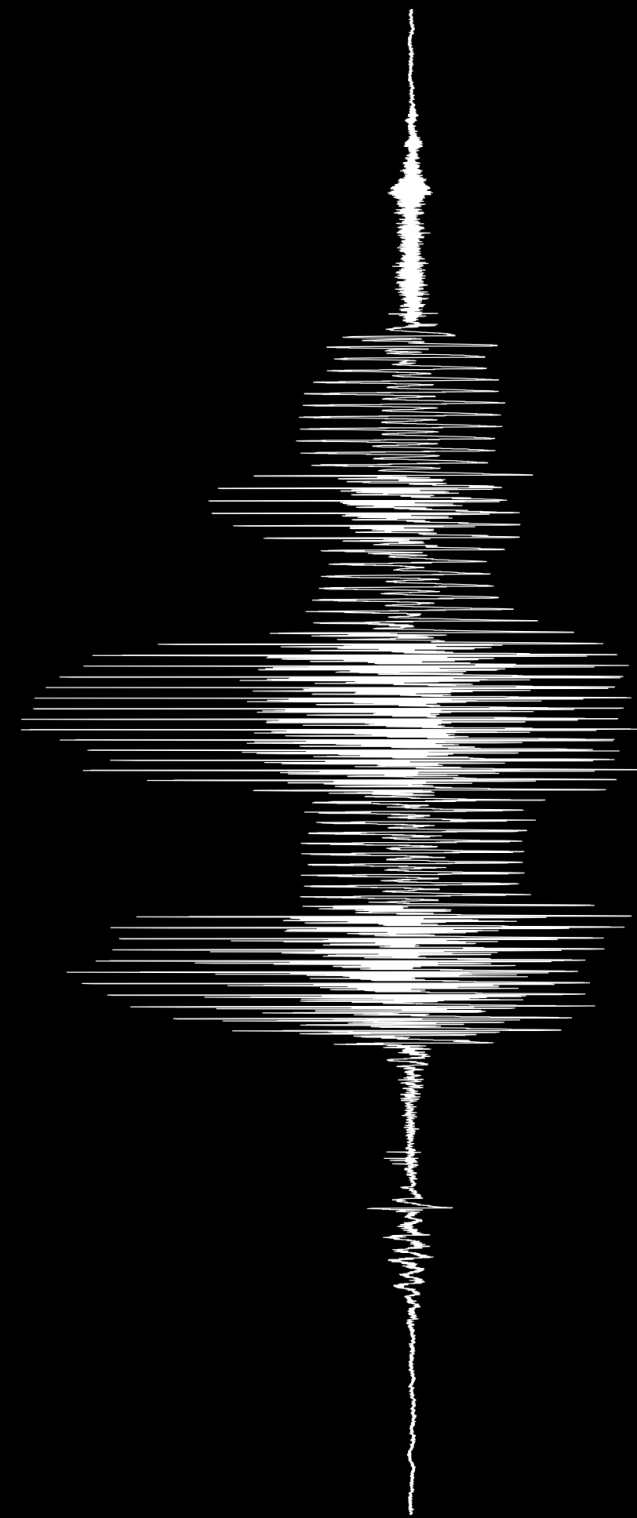
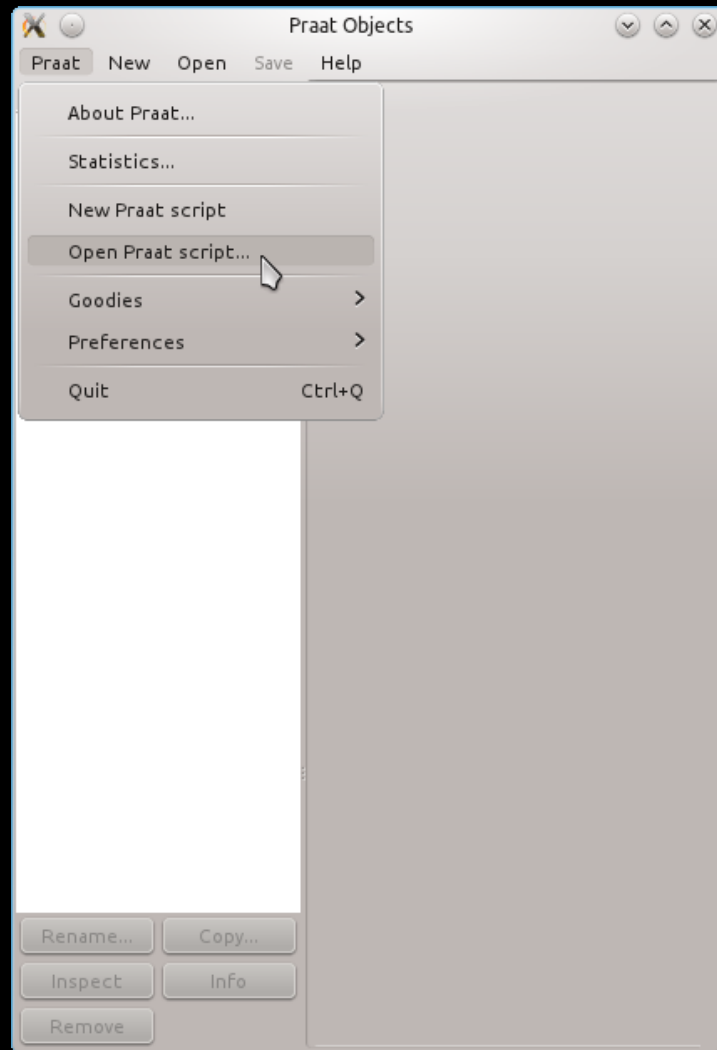
* en versiones recientes de praat ($\geq 5.1.43$), es posible generar programas a partir de *scripts*

la habilidad más importante a la hora de hacer un *script* es la capacidad de tomar una tarea compleja, y reducirla a una sucesión de tareas simples

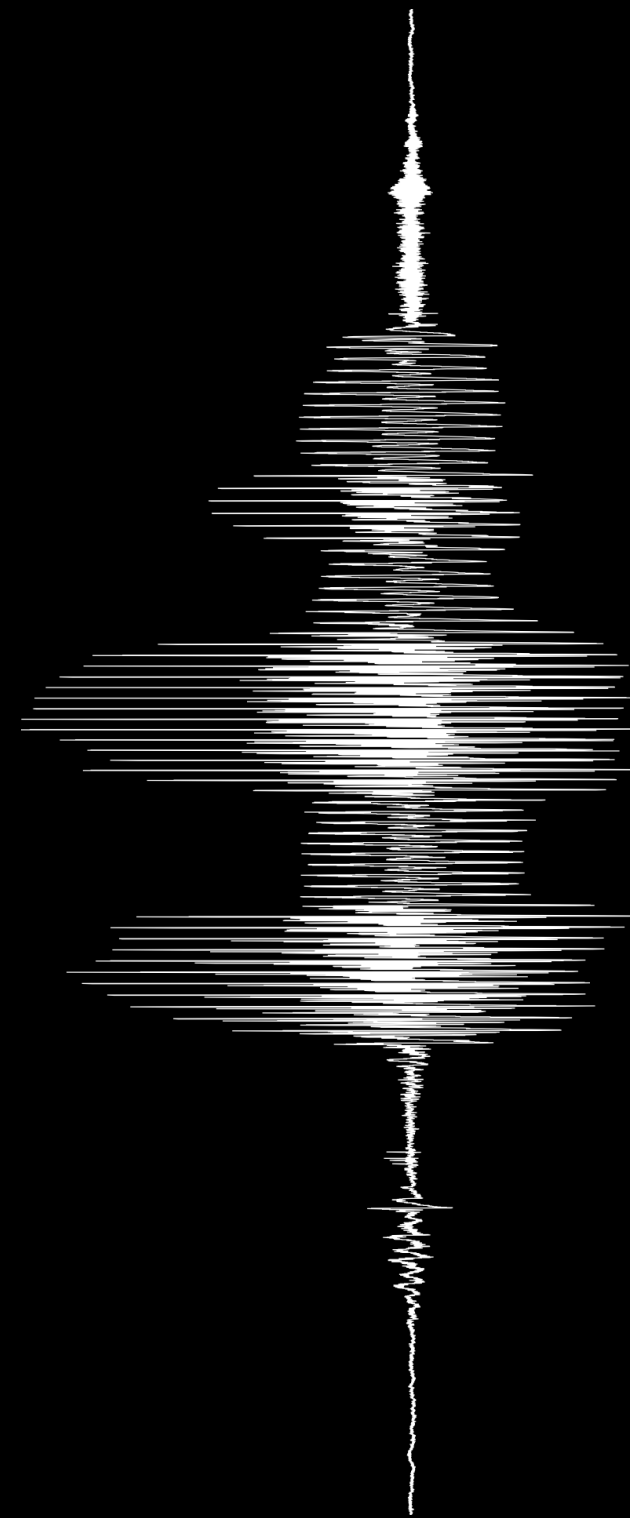
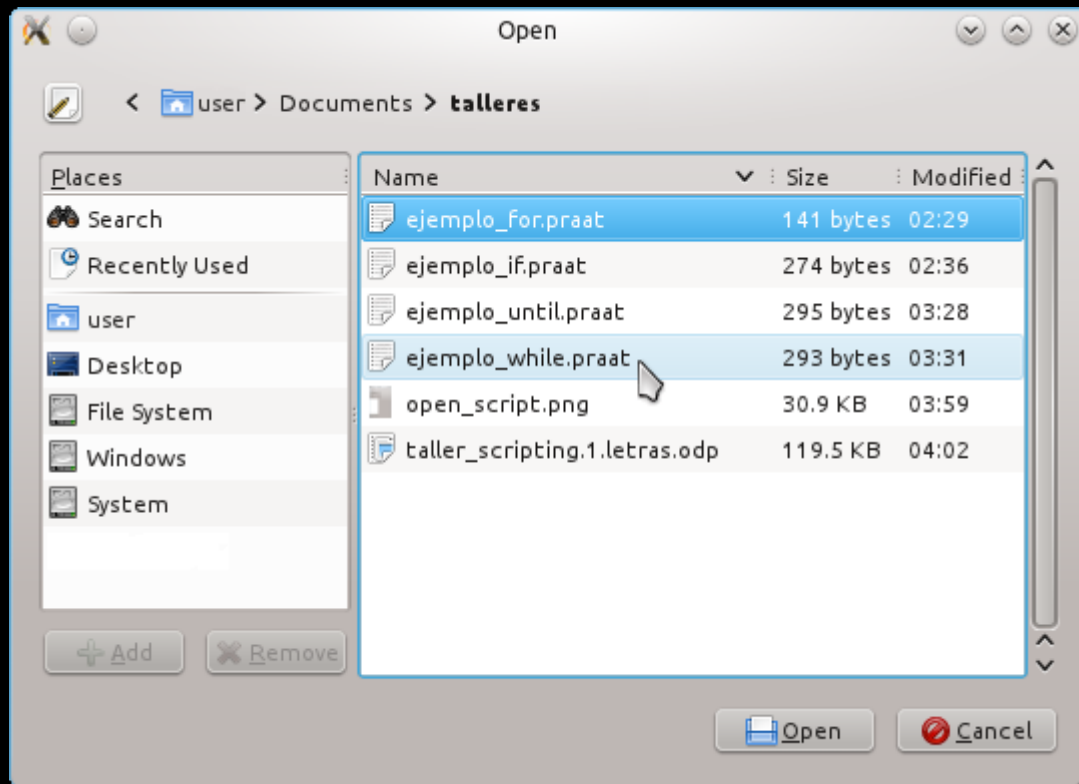
escribir un script es resolver un puzzle



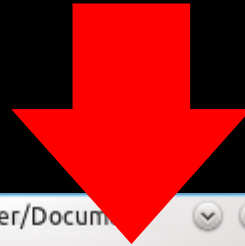
¿cómo se usa un *script*?



¿cómo se usa un *script*?



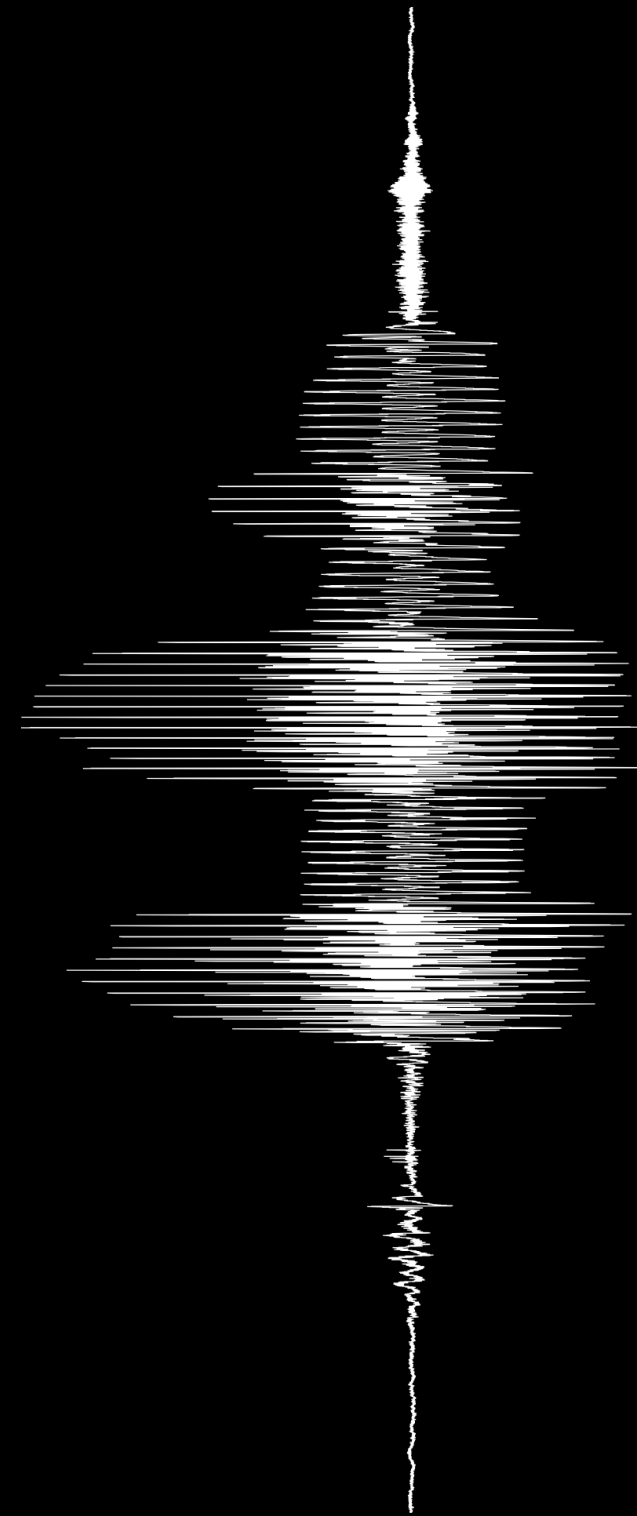
¿cómo se usa un *script*?



```
Script "/home/user/Docum...
File Edit Search Convert Run Help
# Ejemplo de for en praat

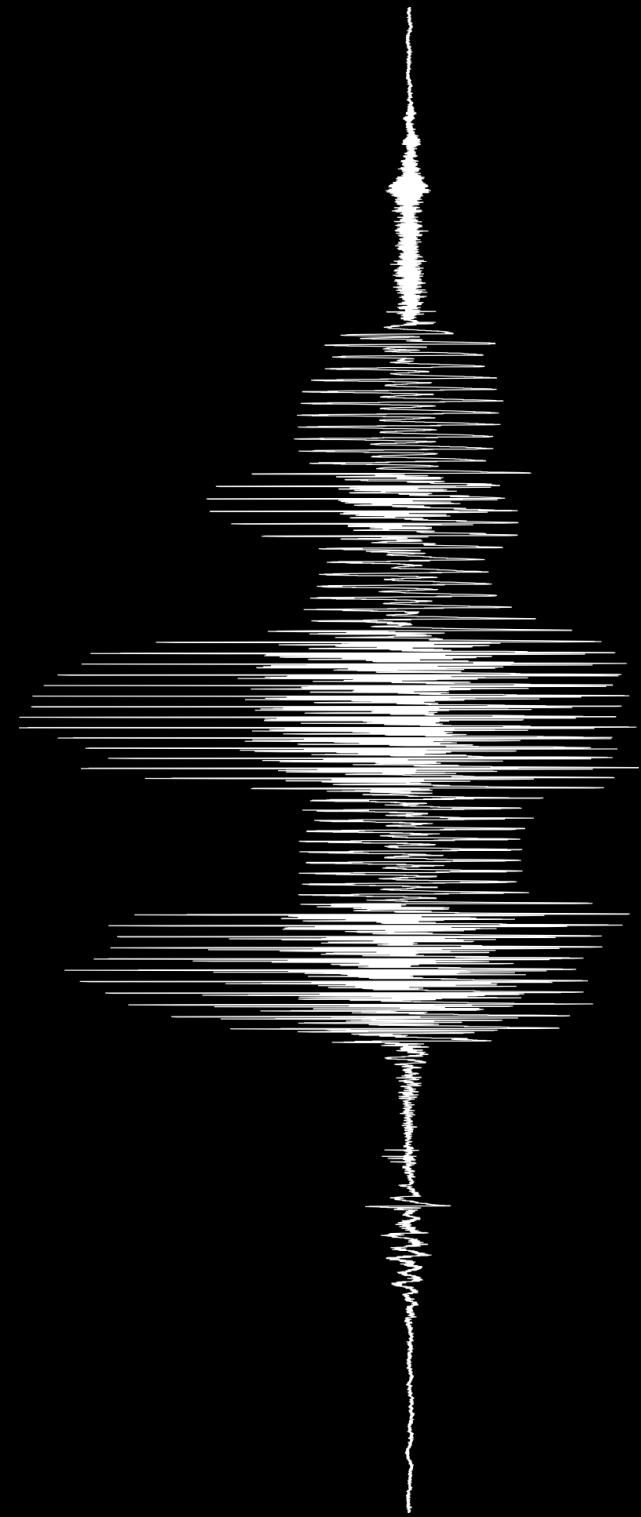
clearinfo
for numero from 0 to 10
  conteo = 10 - numero
  printline 'conteo'...
endfor
printline Despegue!
```

o apretando CTRL + R



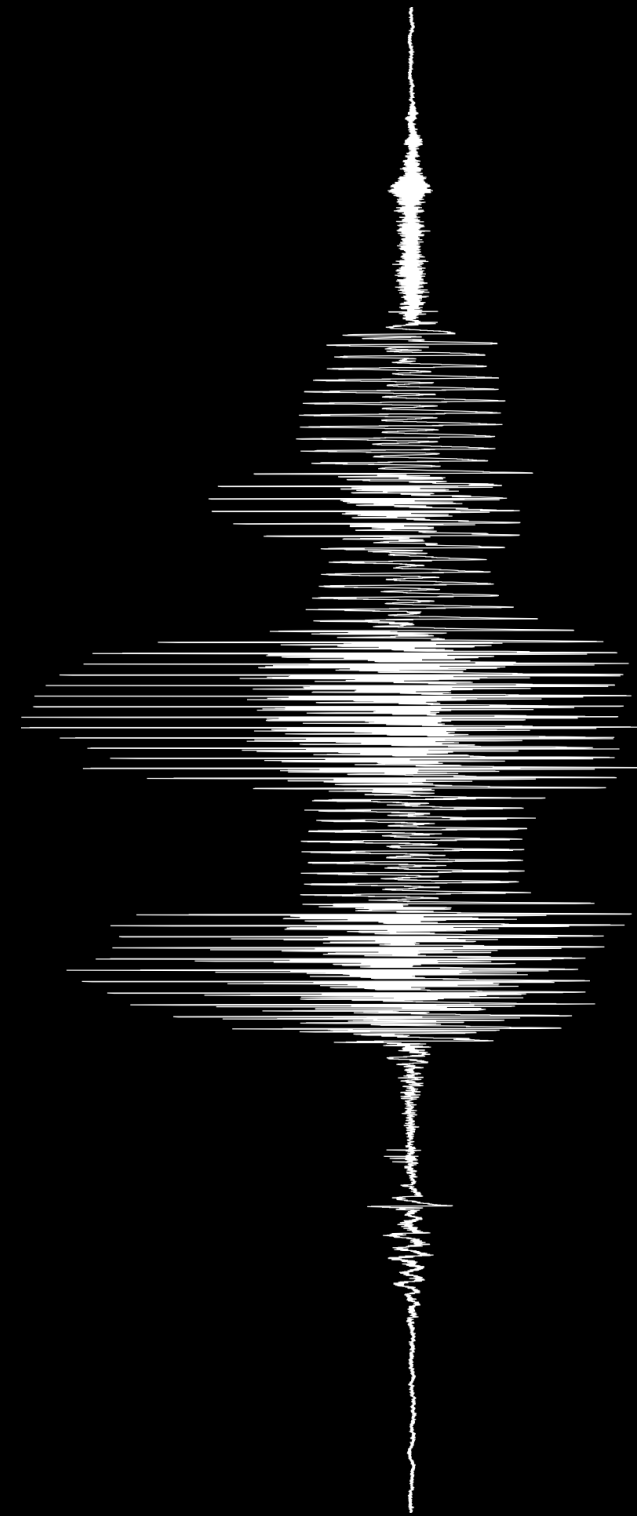
parte 2

el lenguaje de praat



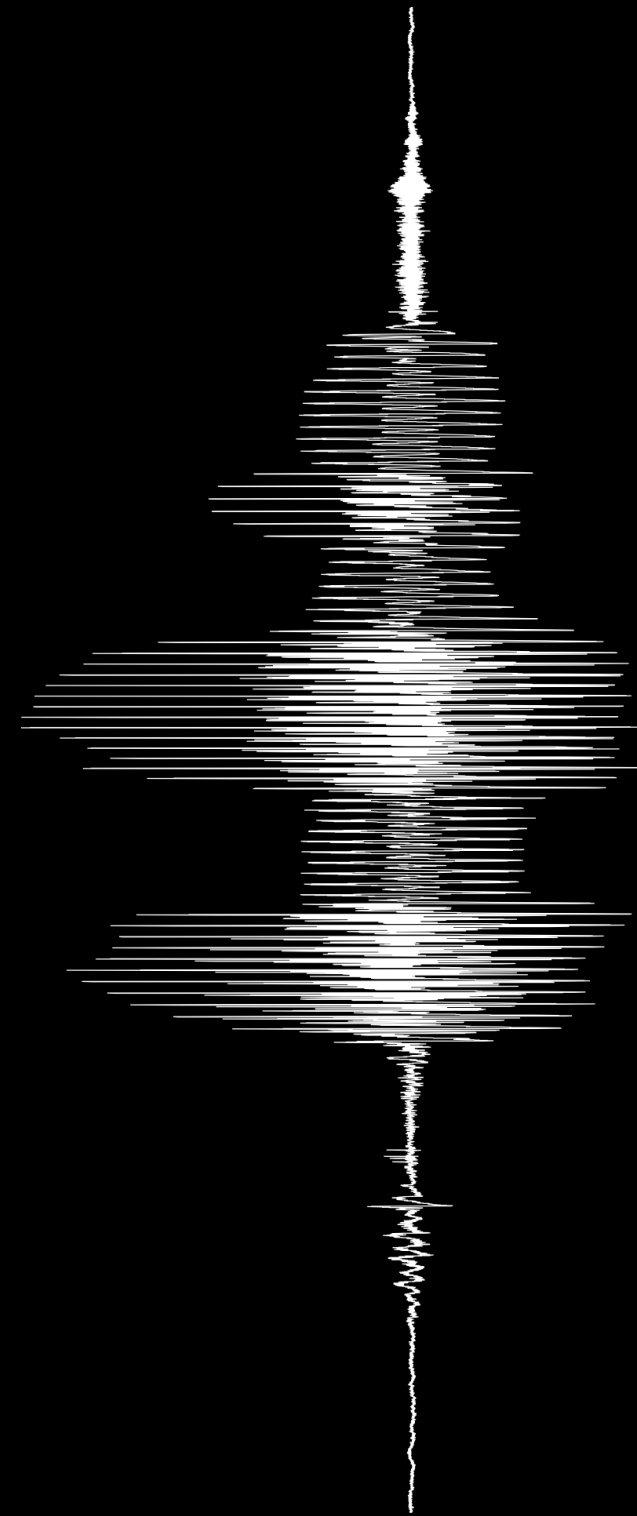
el lenguaje de praat

- objetos
 - los objetos no son archivos
(como saben bien quienes han cerrado praat por accidente)
 - pueden crearse, manipularse, borrarse y guardarse desde *scripts*
 - son parte fundamental de praat



el lenguaje de praat

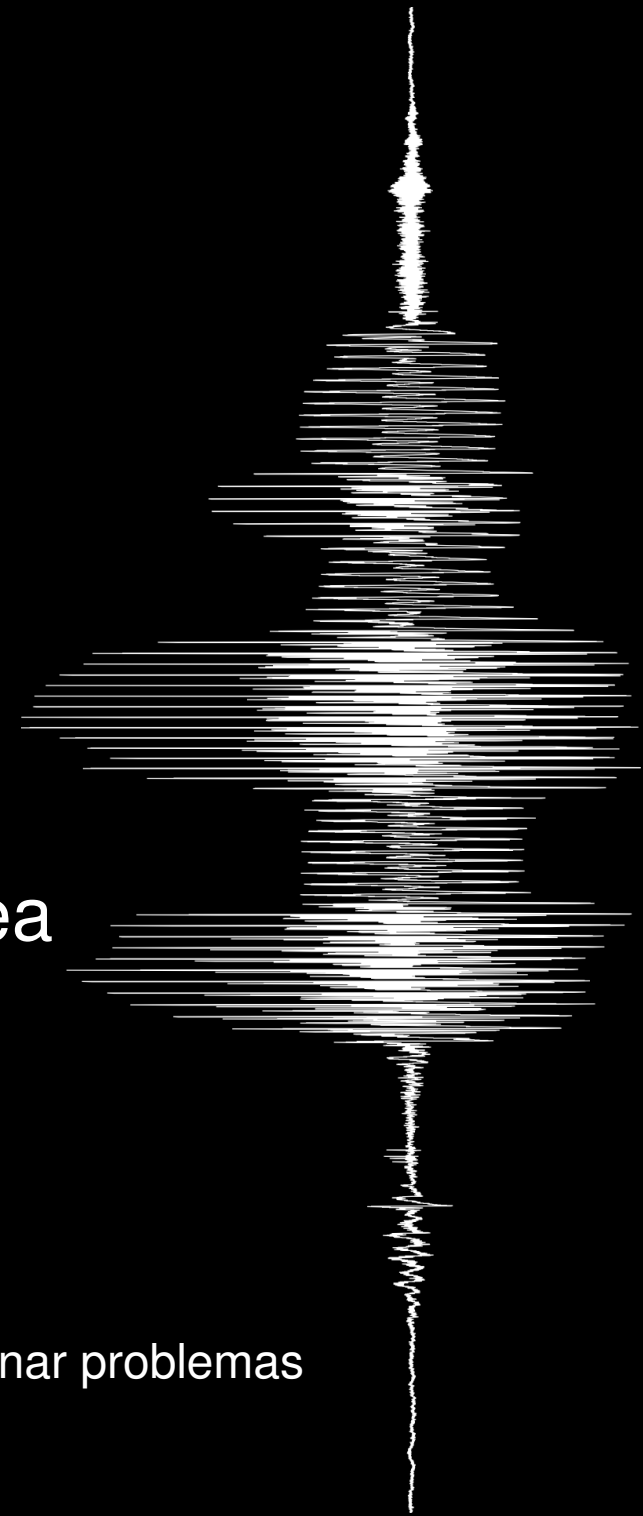
- objetos
 - Sound
 - TextGrid
 - Pitch
 - Table
 - y un larguísimo etcétera...



el lenguaje de praat

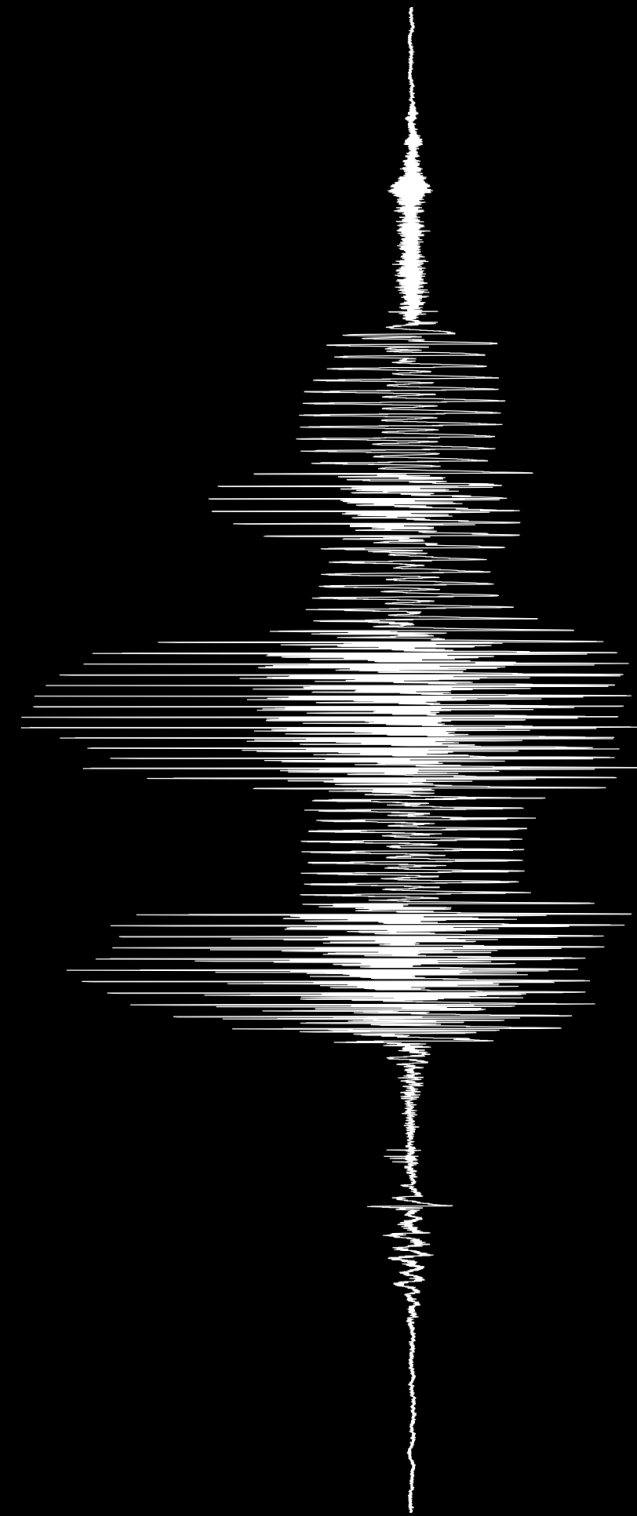
- # comentarios
- clearinfo
limpia la pantalla Info
- printline Imprime una línea a Info
- Echo Limpia Info `_E_` imprime una línea

printline es la principal herramienta para solucionar problemas
... porque problemas va a haber



el lenguaje de praat

- variables
 - `strings$ = "esto es un string"`
 - `variables_numericas = 1337`
 - `arrays []`
- asignaciones
 - como las de arriba
 - por medio de *queries*



el lenguaje de praat

- operadores

- = <>

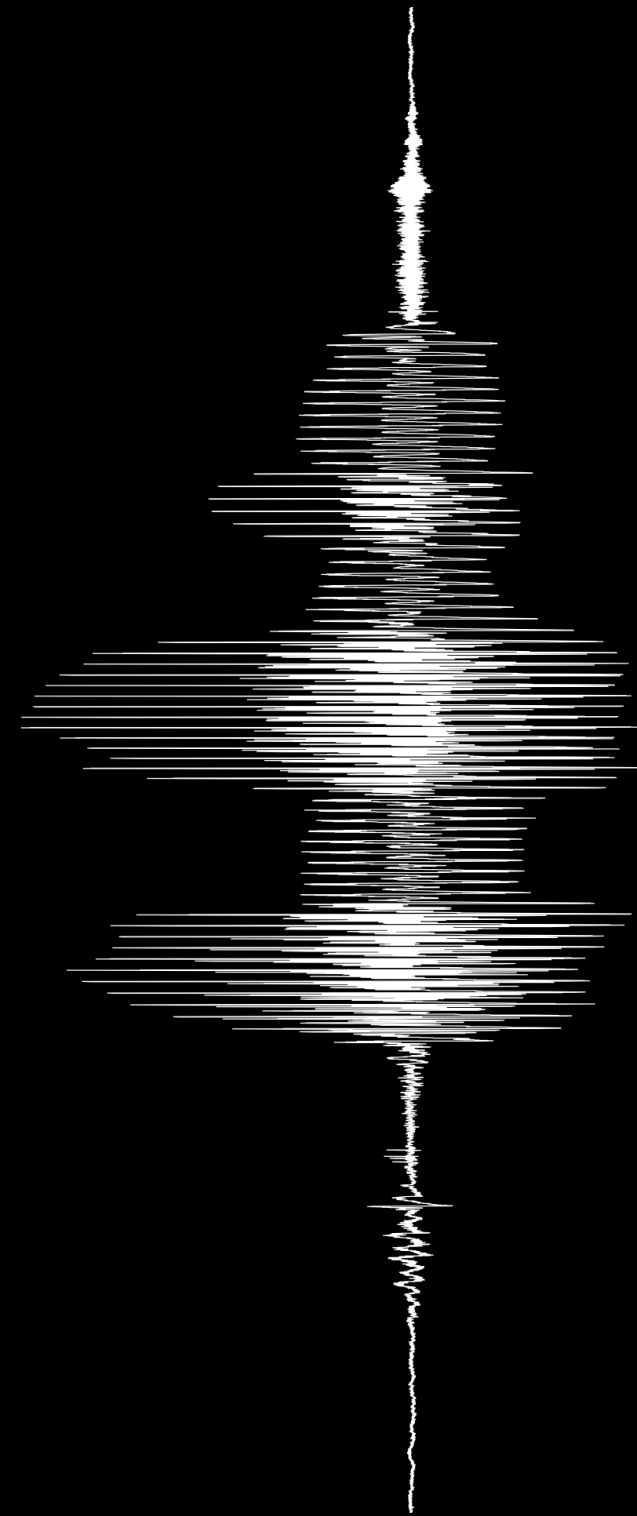
- > >= <= <

- ^ + - * /

- asignaciones

- como las de arriba

- por medio de *queries*

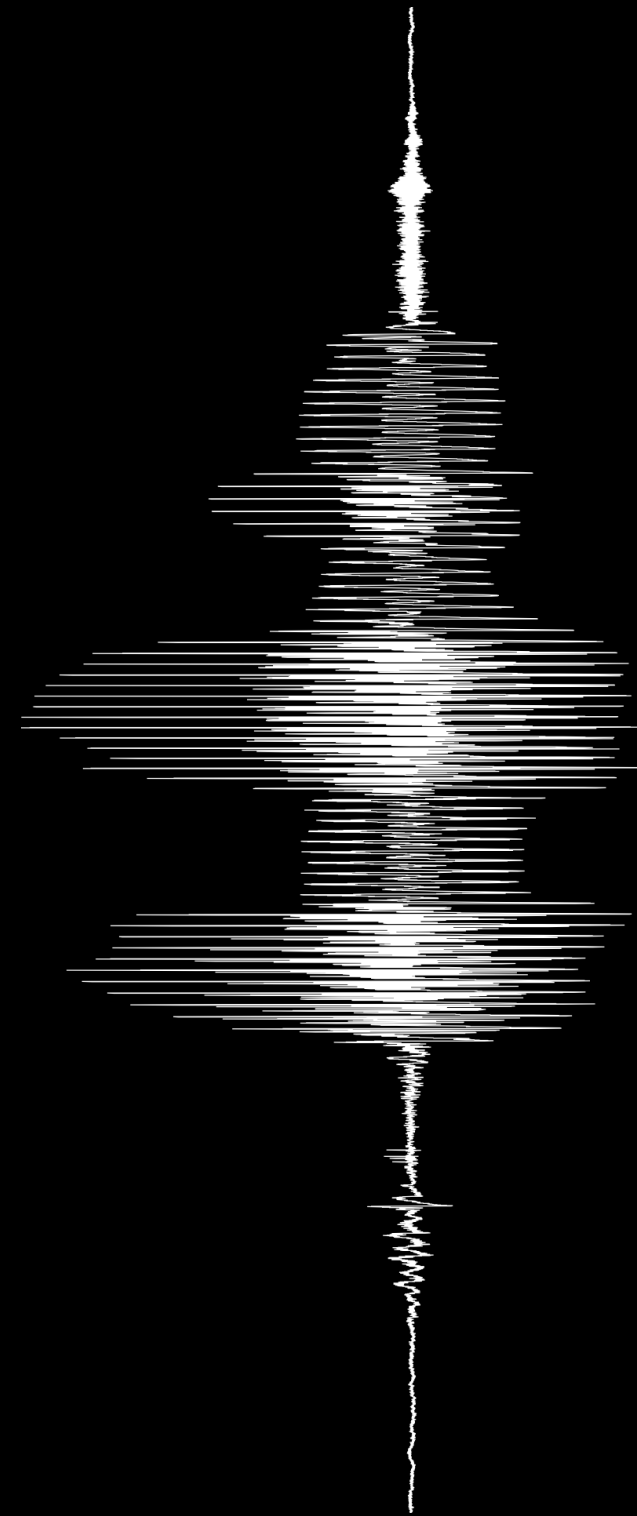


el lenguaje de praat

- estructuras de control
 - for x [from y] to z... endfor

```
# Ejemplo de for en praat  
# Que creen que va a imprimir?
```

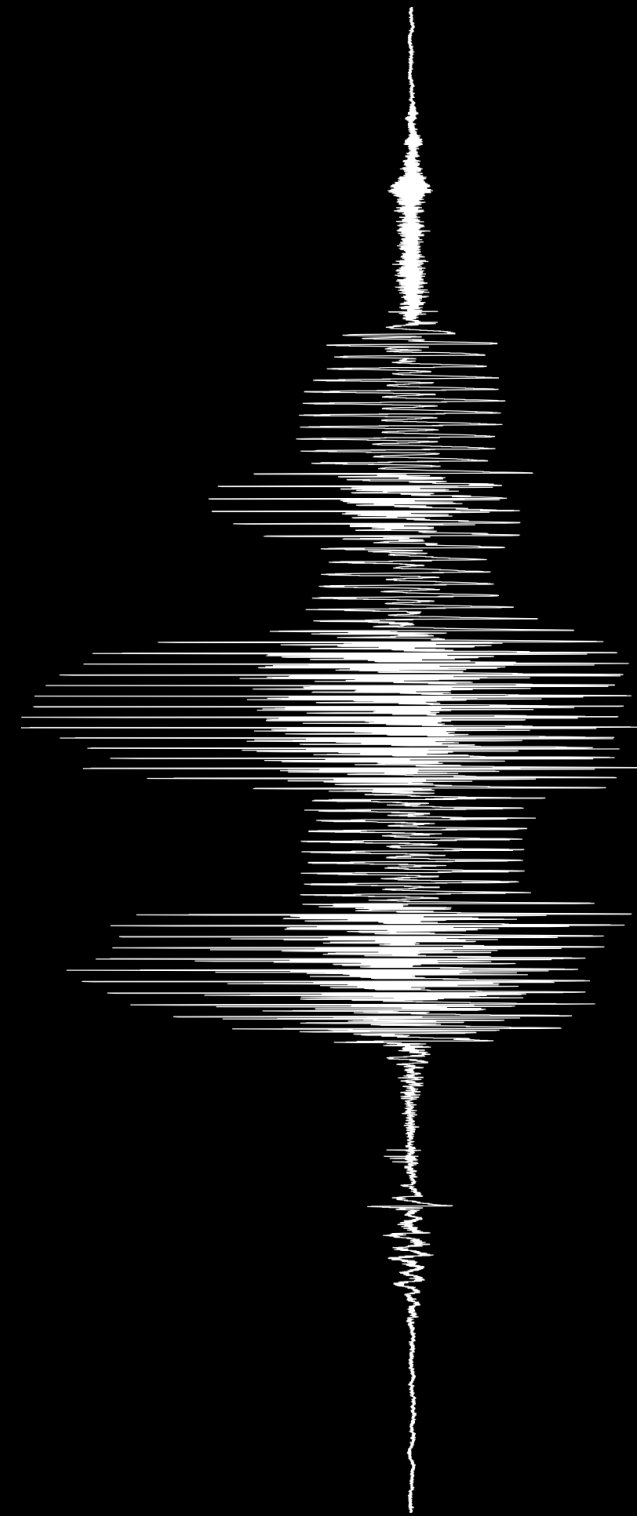
```
clearinfo  
for numero from 0 to 10  
    conteo = 10 - numero  
    printline 'conteo'...  
endfor  
printline Despegue!
```



el lenguaje de praat

- estructuras de control
 - for x [from y] to z... endfor

en praat, *for* siempre incrementa un valor
si $z < y$, el bloque *for* no se ejecuta nunca

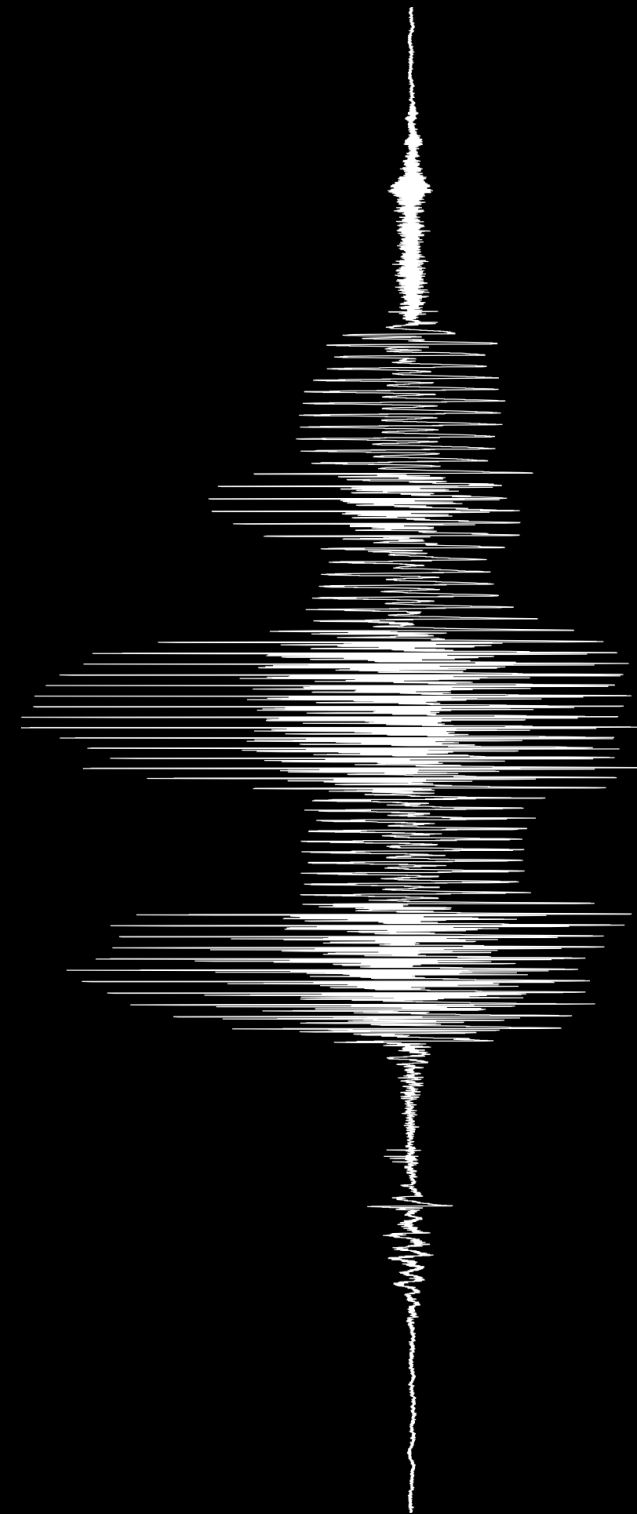


el lenguaje de praat

- estructuras de control
 - if *algo* ... elsif ... else ... endif

```
# Ejemplo de if y for en praat
```

```
clearinfo
println Empezamos!
for numero from 0 to 10
  if numero < 5
    println Antes de la mitad!
  elsif numero > 5
    println Pasamos la mitad!
  else
    println La mitad!
  endif
endfor
println Terminamos!
```

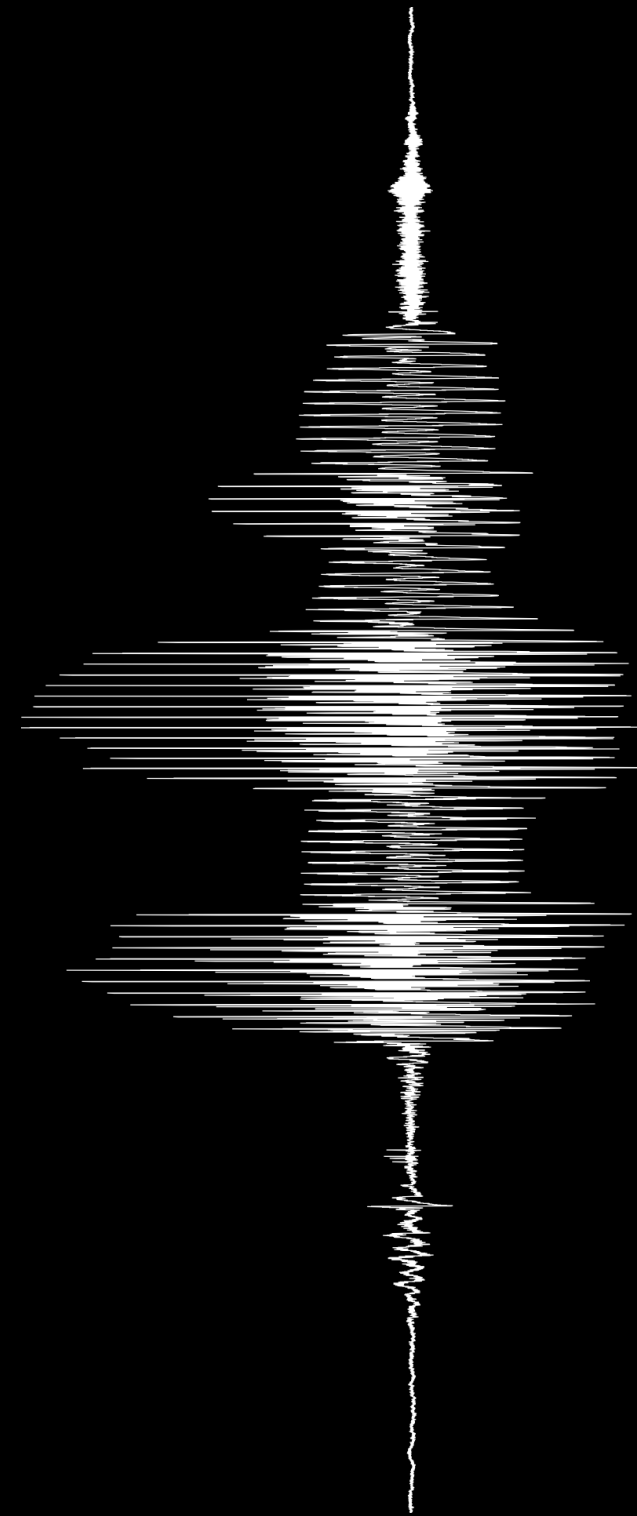


el lenguaje de praat

- estructuras de control
 - if *algo* ... elsif ... else ... endif

otra manera de escribir *elsif* es *elif*

else es muy útil para definir condiciones *default*

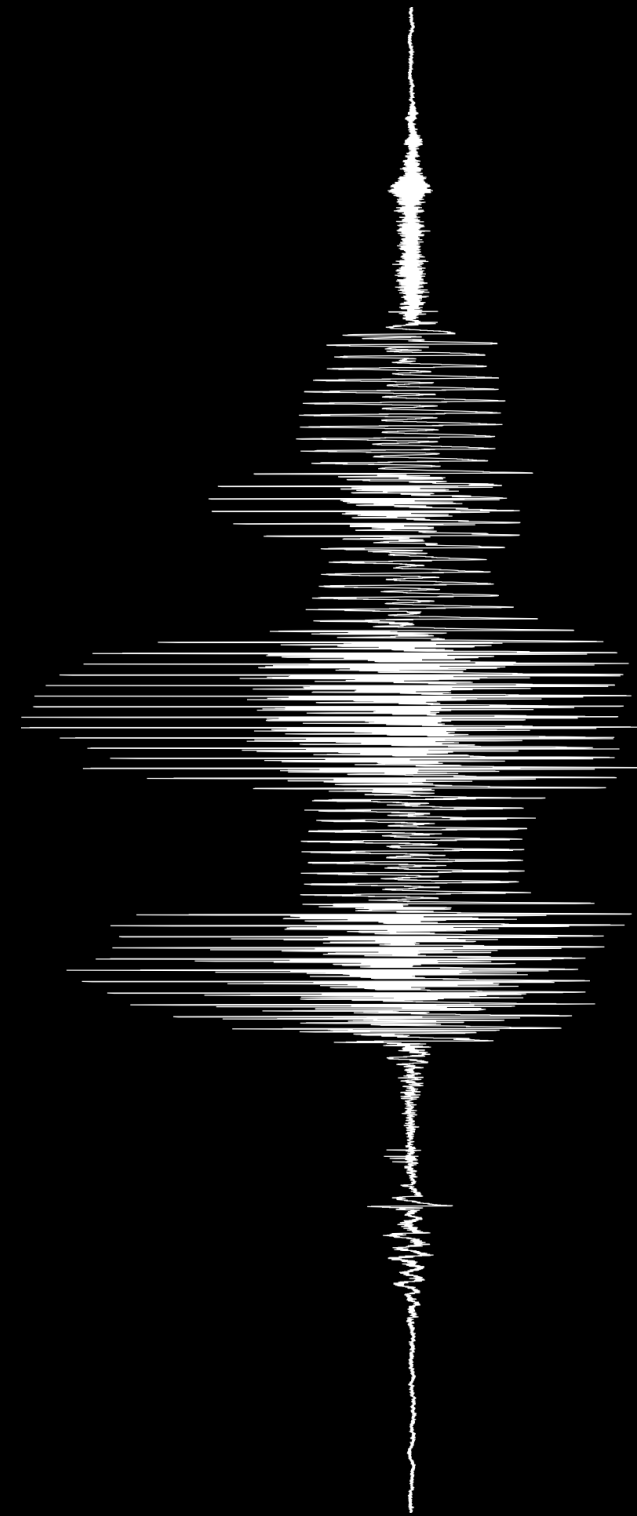


el lenguaje de praat

- estructuras de control
 - repeat ... until *algo*

```
# Ejemplo de repeat en praat

clearinfo
numero = 353467
printline El numero es 'numero'
repeat
  if numero > 10
    numero = numero - numero/2
  elsif numero < 10
    numero = numero + numero/2
  endif
  tmp = round(numero)
  printline ...y ahora es 'tmp'...
until round(numero) = 10
printline Y estamos!
```



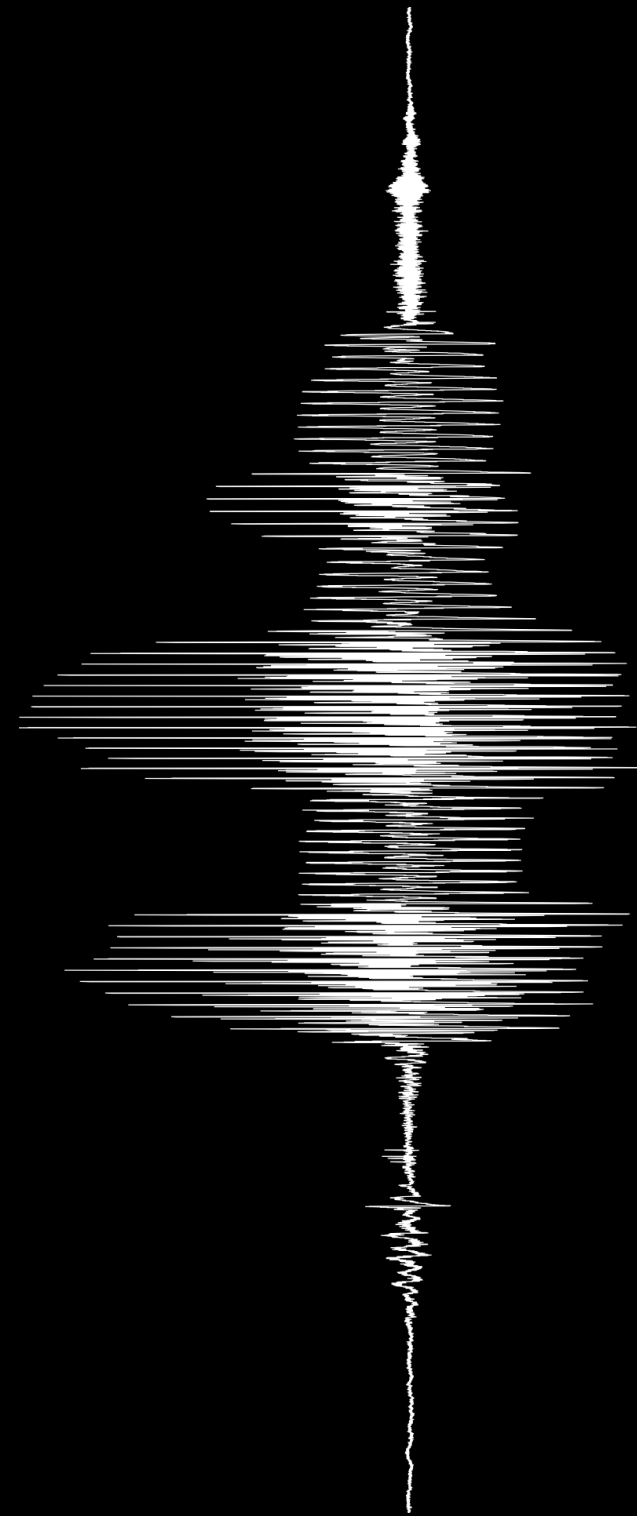
el lenguaje de praat

- estructuras de control
 - repeat ... until *algo*

es posible crear loops infinitos
(¡usar con cuidado!)

el bloque se repite hasta que la condición
sea verdadera

la condición se prueba al llegar al final

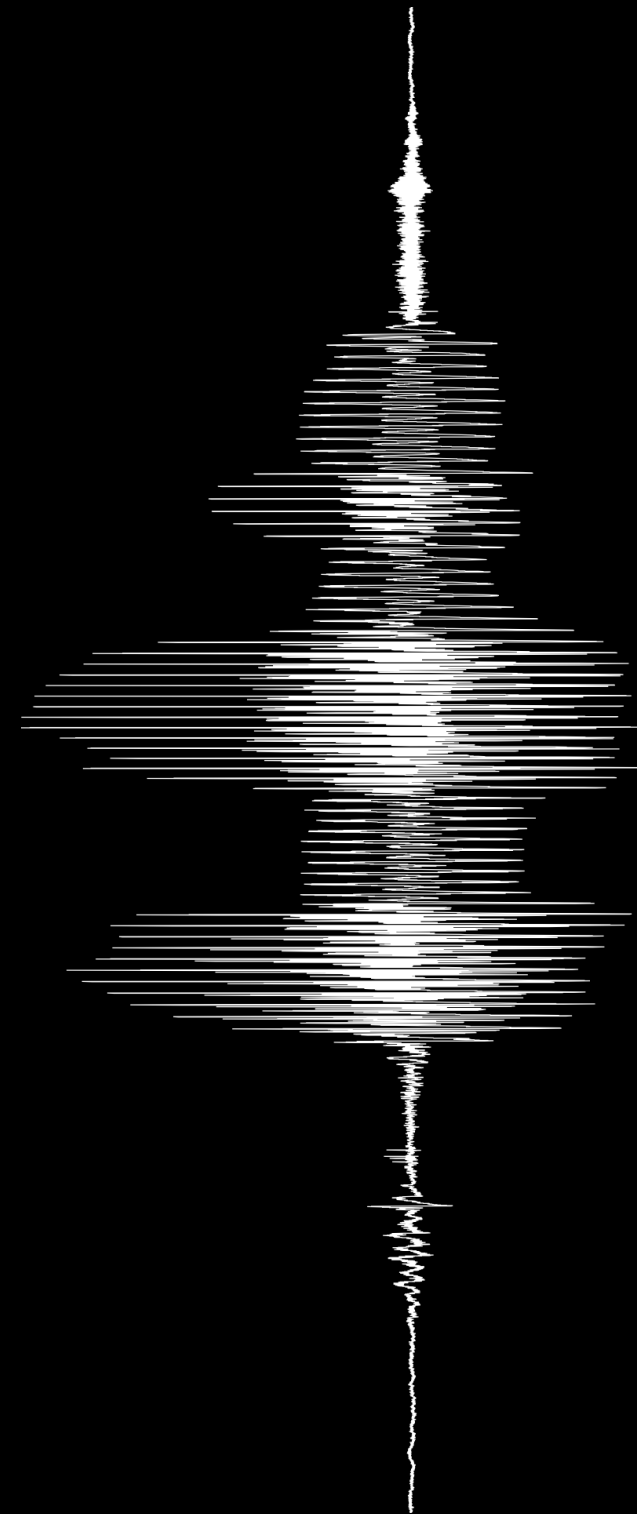


el lenguaje de praat

- estructuras de control
 - while *algo* ... endwhile

```
# Ejemplo de while en praat

clearinfo
numero = 353467
printline El numero es 'numero'
while round(numero) <> 10
    if numero > 10
        numero = numero - numero/2
    elsif numero < 10
        numero = numero + numero/2
    endif
    tmp = round(numero)
    printline ...y ahora es 'tmp'...
endwhile
printline Y estamos!
```



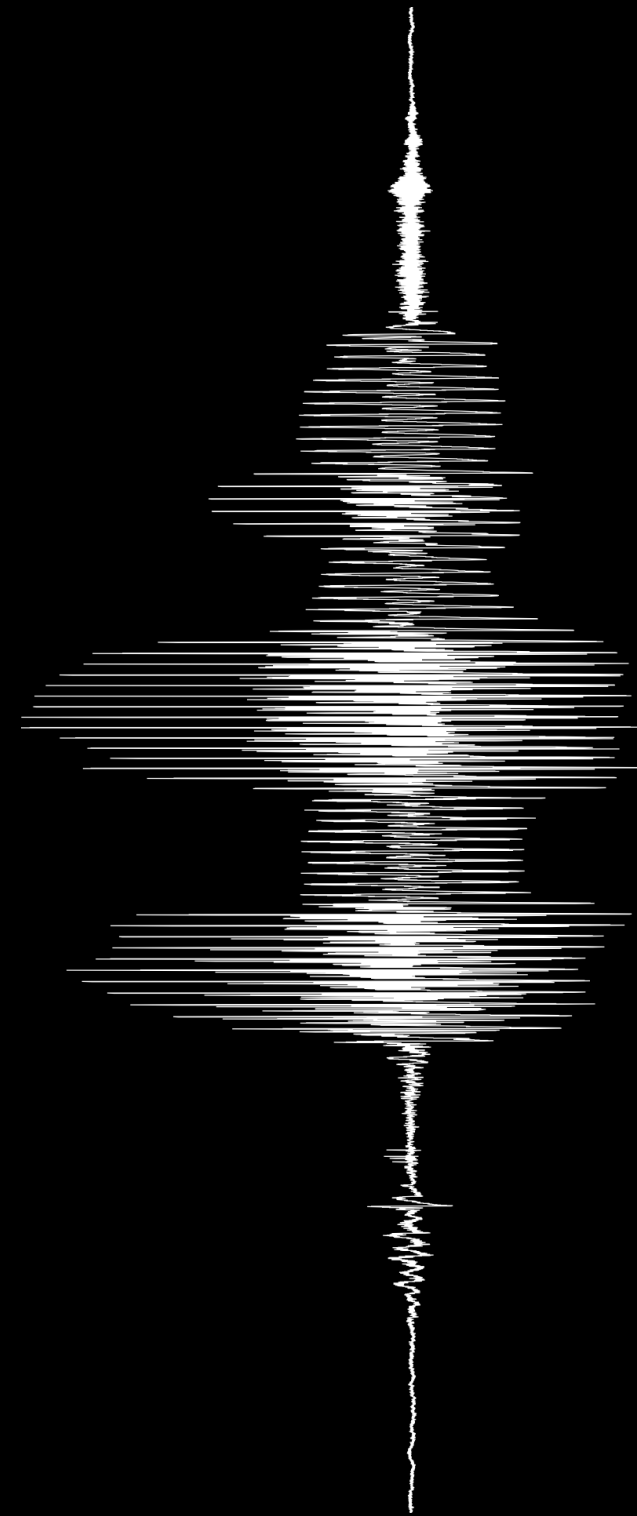
el lenguaje de praat

- estructuras de control
 - while *algo* ... endwhile

es posible crear loops infinitos
(¡usar con cuidado!)

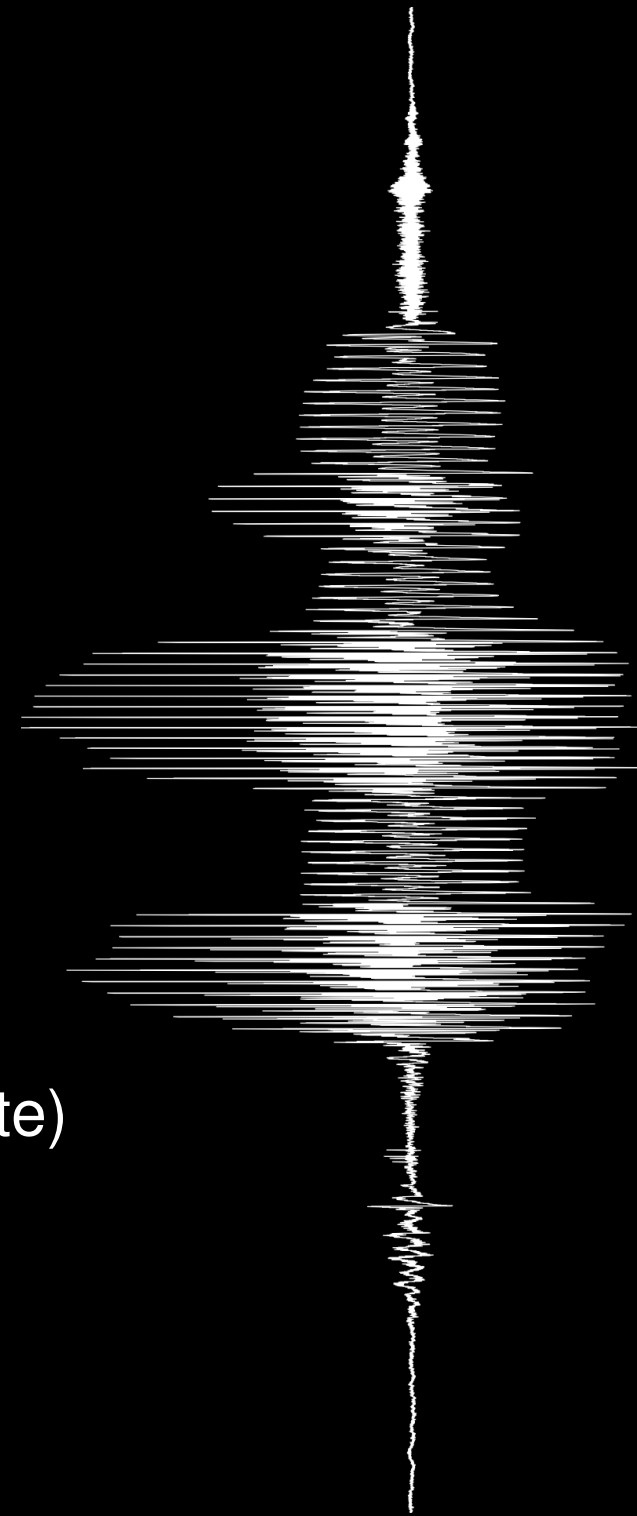
el bloque se repite mientras la condición
sea verdadera

la condición se prueba antes de empezar



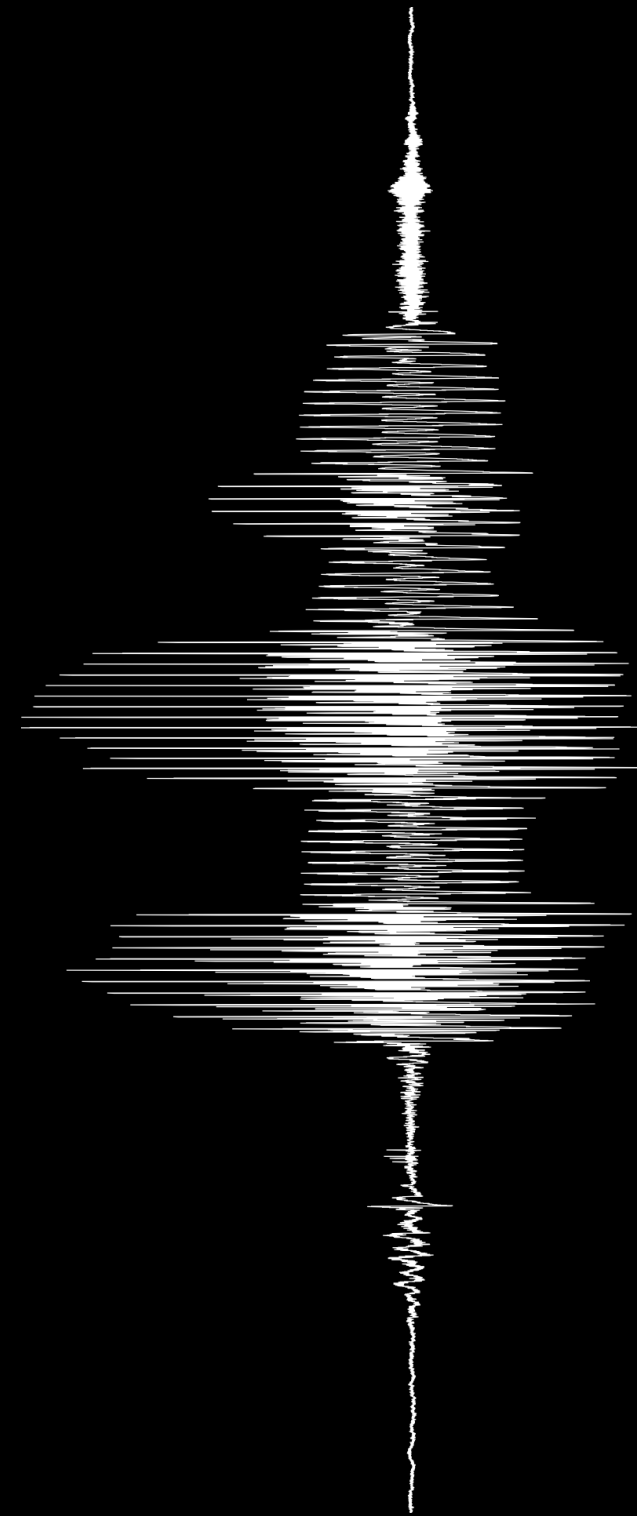
el lenguaje de praat

- estructuras de control
 - for x [from y] to z... endfor
 - if *algo* ... elsif ... else ... endif
 - while *algo* ... endwhile
 - repeat ... until *algo*
- procedures
(que veremos más en detalle más adelante)



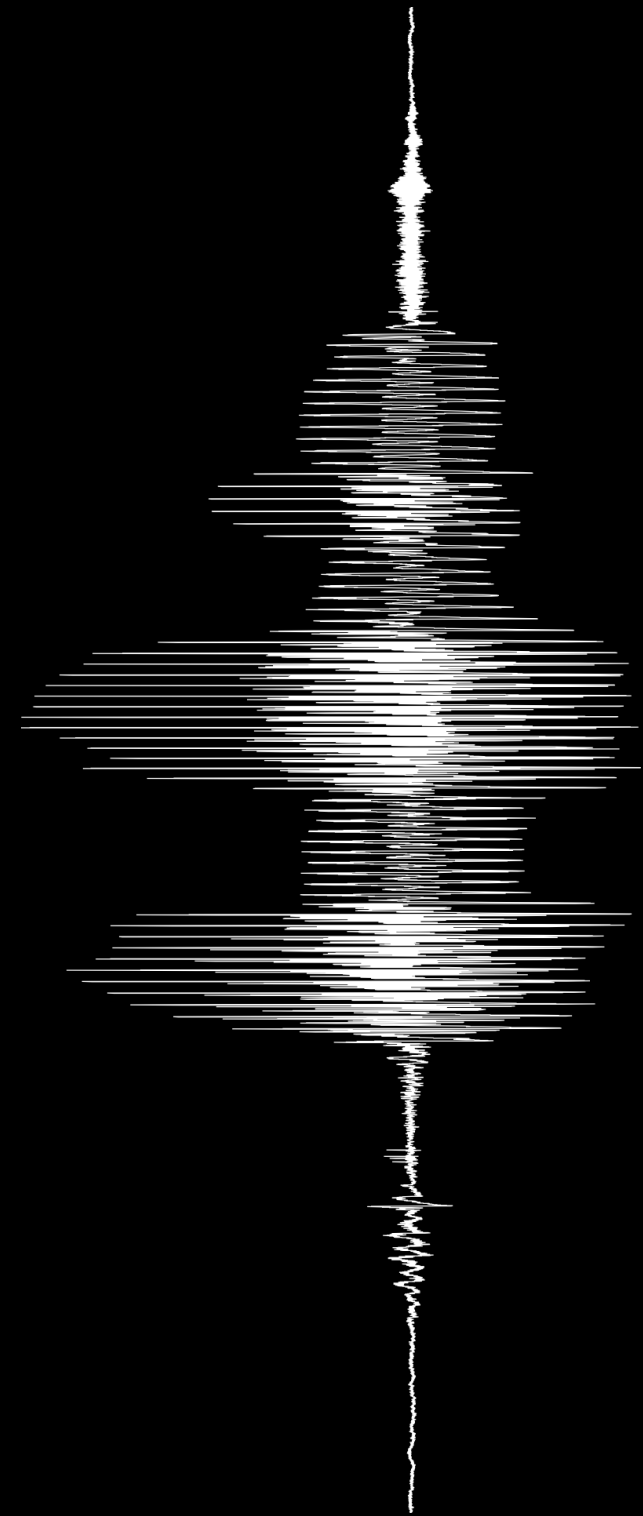
el lenguaje de praat

- operadores lógicos
 - and
 - or
 - not
- funciones
 - de texto
 - numéricas



en la próxima sesión...

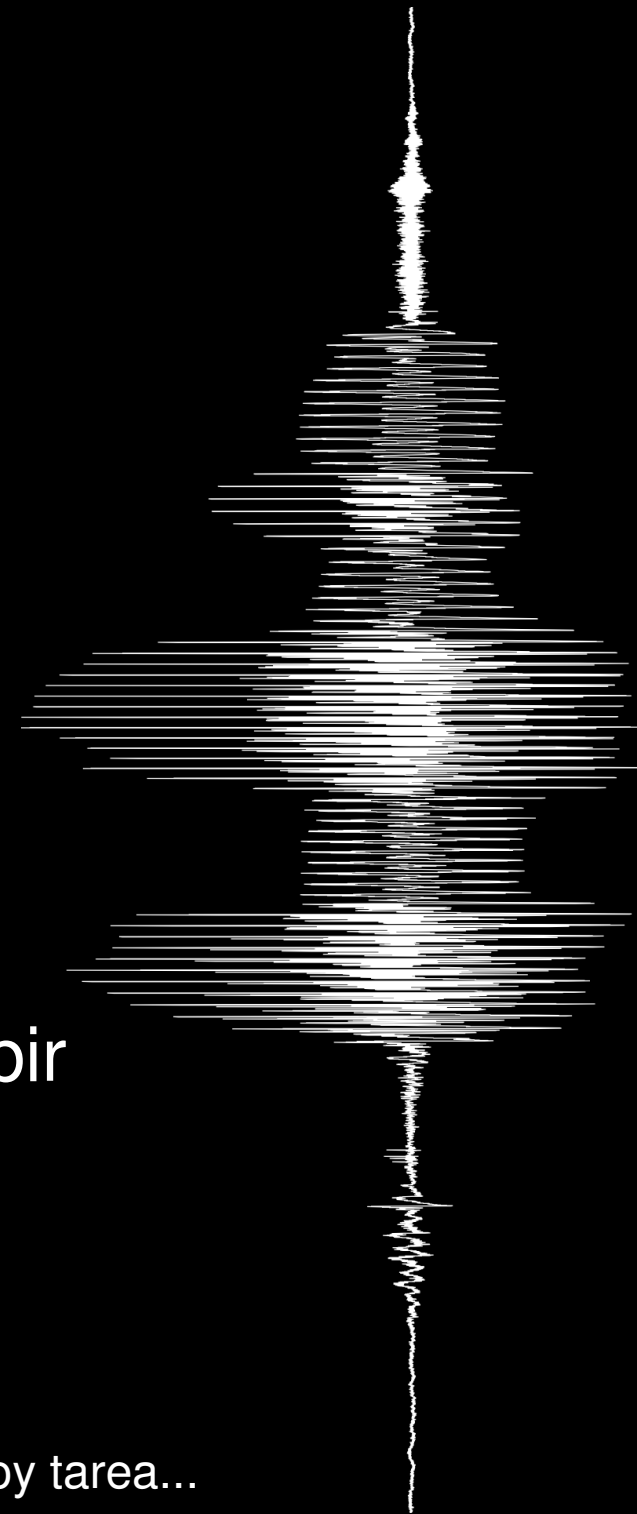
- *queries* numéricos y de texto
- manipulación de objetos
- interacción con el usuario
- buenas prácticas



tareas

- escribir un script que imprima los números pares entre 1 y 200 (pista: usen *mod*)
- escribir un script que imprima los números primos hasta 100
- si no usaron ni `while` ni `repeat`, reescribir el script anterior utilizándolos. ¿qué cambia? ¿qué queda igual?

porque no es un taller si no doy tarea...



otros recursos

los *scripts* de ejemplo de esta y las demás sesiones (así como esta presentación) estarán disponibles en

<http://www.pinguinorodriguez.cl/tallerpraat.html>

la documentación de praat contiene mucha información. pueden revisarla en

<http://www.fon.hum.uva.nl/praat/manual/Scripting.html>

Scott Sadowsky tiene un gran número de recursos útiles en su página

<http://ssadowsky.hostei.com/praat.html>

