



Praat Scripting

Introduction

&

Crash course on file management scripting

Goals

Overview of Praat scripting

- Basic fundamentals
- Create a script from your Praat history
- For-Loops
- Boilerplate code

Modify a script!

Goals:

1. Get number of .wav files in a directory
2. Print out individual file names
3. Get duration of each .wav file

Bonus goals:

1. Get total duration of *all* .wav files
2. Copy all .wav files ending in "_3" to a new directory and append "_copy" to the file name
3. Toggle option to "clean up" as you go

Preliminaries: Slack

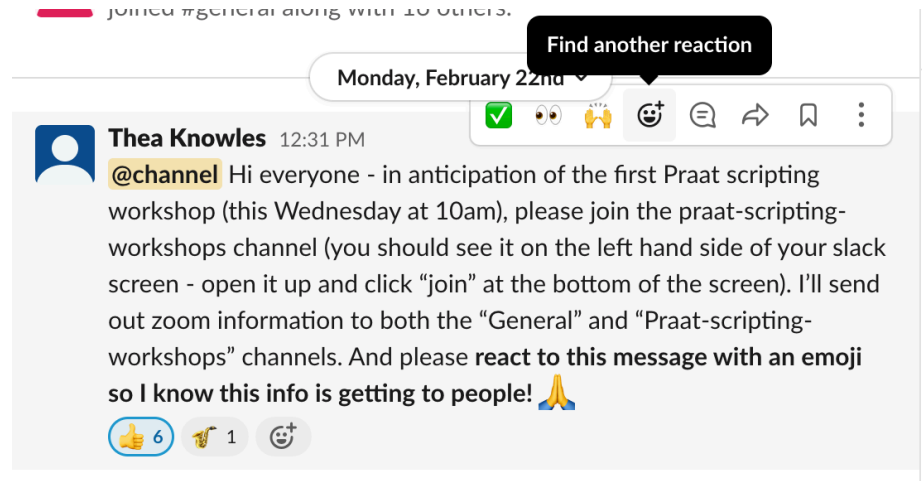
In the workshop:

- Use Slack during the “hands-on” components to ask questions, post screenshots, etc.

Long-term goal:

- Troubleshoot with other UB Praat users!

- “React” to things on Slack:



The screenshot shows a Slack message from Thea Knowles at 12:31 PM. The message text is: "@channel Hi everyone - in anticipation of the first Praat scripting workshop (this Wednesday at 10am), please join the praat-scripting-workshops channel (you should see it on the left hand side of your slack screen - open it up and click “join” at the bottom of the screen). I'll send out zoom information to both the “General” and “Praat-scripting-workshops” channels. And please react to this message with an emoji so I know this info is getting to people! 🙏". Above the message, a date separator reads "Monday, February 22nd". A reaction menu is open over the message, showing a green checkmark, a surprised face emoji, a clapping hands emoji, and a smile with heart eyes emoji. A black tooltip above the menu says "Find another reaction". Below the message, a reaction bar shows a thumbs up emoji with a count of 6, a clapping hands emoji with a count of 1, and a smile with heart eyes emoji.

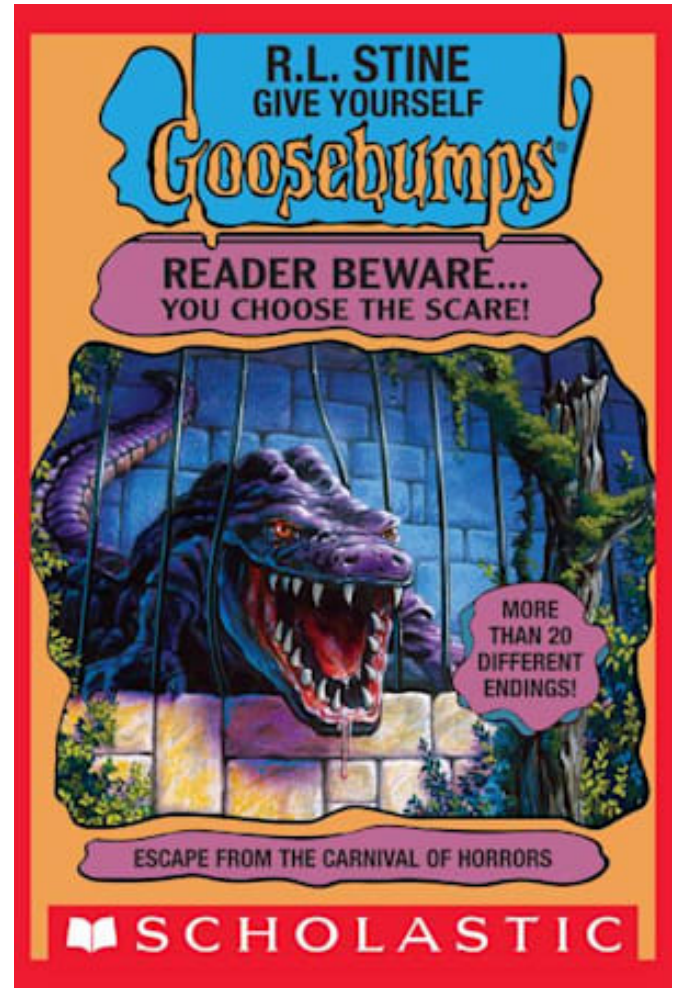
- I will ask you to do this during the hands-on so I know when you’re ready to move on

This is a choose-your-own-adventure workshop

- Fundamentals together
- Hands-on: choose the level of difficulty you want to try

Assumptions for today:

- At least some familiarity with Praat

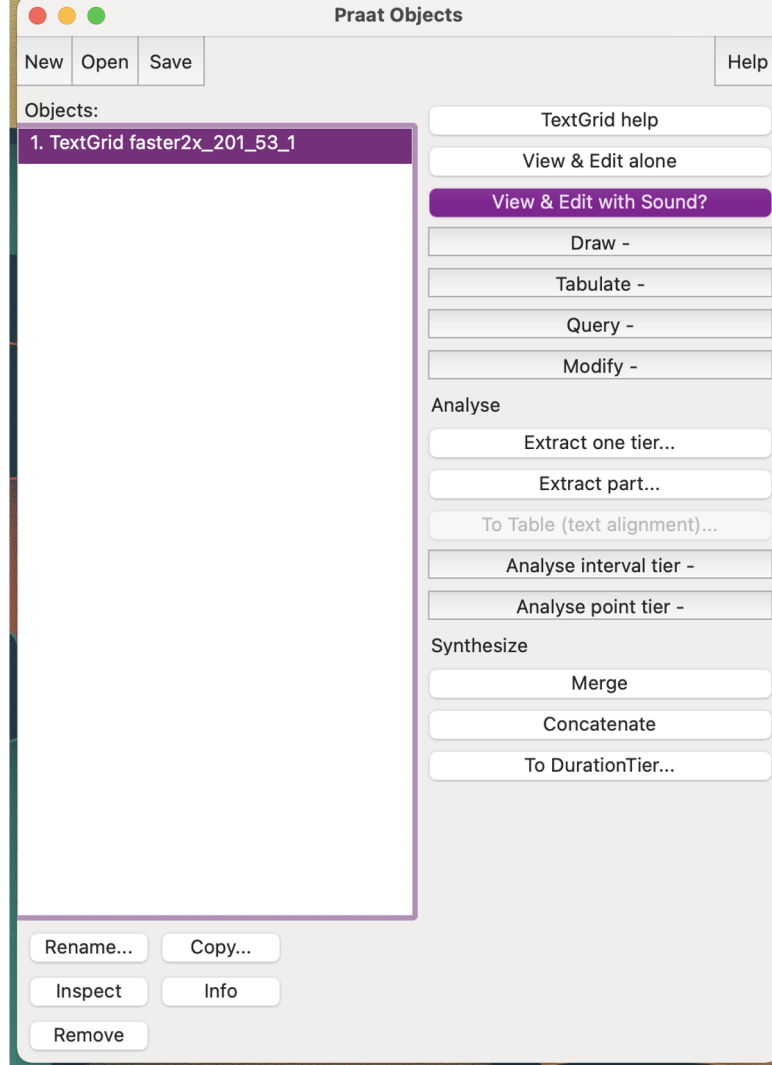


Overview of Praat scripting. Why?

Using Praat

Can accomplish a lot by

- Clicking buttons
- Selecting menu items
- Entering information into forms



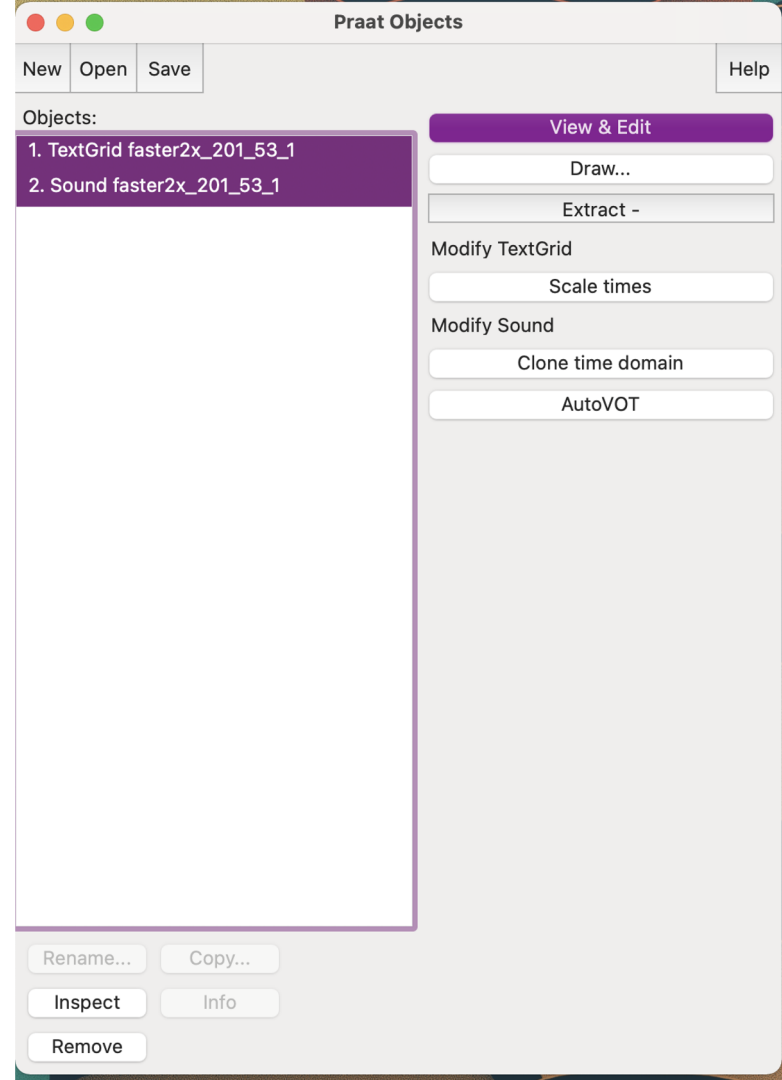
Using Praat

Example of things we do a lot in my lab:

- Find folder with files to annotate
- Open TextGrid/wav pair in Praat
- Select them both
- View & edit together
- Annotate TextGrid
- Save TextGrid

We start off learning to do this by hand

Then we use Praat scripts to do a lot of the repetitive tasks for us



With Praat scripts you can...

- Automate repetitive tasks like...
 - opening/closing/saving files
 - Saves a lot of time (once you get the basics down)!
- Have consistent, replicable measurements
- Minimize human error in data preparation
- Keep a careful log of how you did your analyses
 - Code = instructions

Use your knowledge of Praat to write scripts

A frequent routine of mine:

1. Copy old boilerplate from old script to new script
2. Add comments to top with brief description of script
3. Add some stuff
4. Test (run) script a lot (almost every line)
 1. Force script to crash at relevant points
 2. Lots and lots and lots of printing to the Praat info window as sanity check
5. When I can't remember function/argument structure, try to do it manually in Praat to see what it looks like
 1. Praat history function
 2. (lots of) Googling + Praat scripting manual

Basic fundamentals

Some terminology

Coding: Writing in a language a computer can understand

Scripting: A type of coding that tells a specific program exactly what actions to take

Programming: Writing code that serves to actually create another program (an app, software, etc)

Scripts: Text files containing code.

- Scripting, coding, and programming are sometimes used interchangeably

What is a script?

- Sequentially ordered set of instructions that are given to a program that can interpret & execute them
- Must be written in the program's language (syntax)
 - Praat script
 - R script
 - Etc.

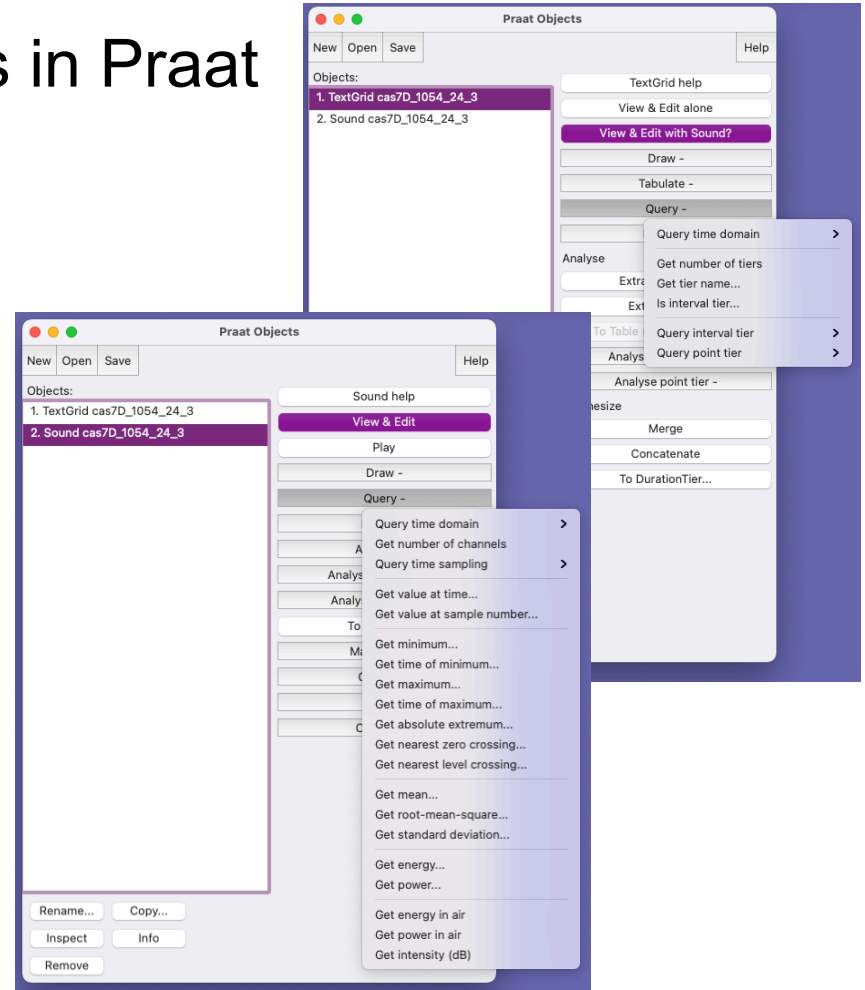
Commands & Built-in Functions in Praat

Commands in Praat: Instructions you give to Praat (aka - the buttons you click!)

- Dependent on object type
 - Sound vs. TextGrids vs. Strings, etc
- Written as-appears in Praat
- Start with capital letter
- Arguments given in order you would type them in Praat

Read more:

<https://praatscripting.lingphon.net/commands.html>



Commands & Built-in Functions in Praat

Built-in Functions: Instructions that don't have a corresponding Praat button like `selectObject`, `clearinfo`, `fileReadable()`...

- Human action
- Starts with lower case

Read more:

<https://praascripting.lingphon.net/commands.html>

Variables

- In programming, a variable is a place in the computer's memory where something is stored → **containers**
- It has a name & value
- Value is assigned to the name like this:
 - *name = value*
 - *veggies = broccoli*
 - *proteins = chicken*

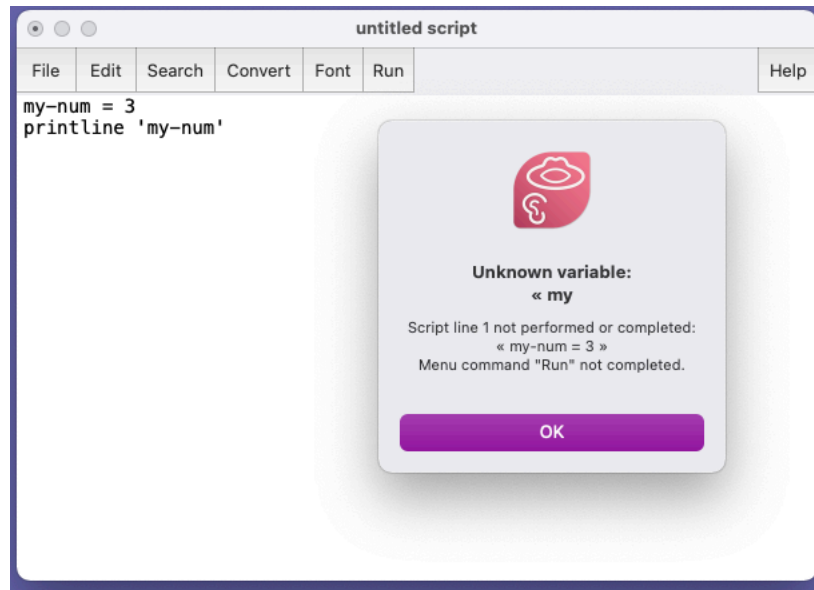


Variables

- Value is assigned to the name like this:
 - *name = value*
 - In Praat, Names must start with lowercase and have only letters, digits, and underscores

Valid variables	Invalid variables
<pre>num = 2 my_num = 2 myNum = 2 n = 2 akjl_123 = 2</pre>	<pre>Num = 2 my num = 2 my-num = 2 _n = 2 2 = 2</pre>

Variable names don't HAVE to be informative, but your life will be easier if they are



Variable types in Praat

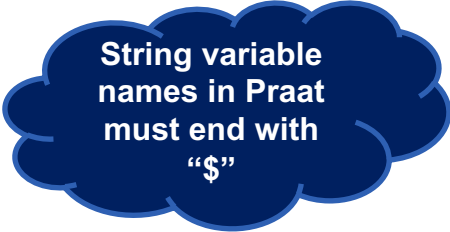
Common types: numeric (number) and string (text)

```
my_number = 42
```


```
my_other_number = 1.0001
```

```
my_text$ = "3"
```

```
my_other_text$ = "hello!"
```



String variable
names in Praat
must end with
"\$"



String variable
values must be in
" "

Learn more:

- <https://praatscripting.lingphon.net/variables-1.html>
- https://www.fon.hum.uva.nl/praat/manual/Scripting_5_1_Variables.html

Praat syntactic structure: Considerations

One complete statement per line

When you run a script, lines are executed in order

If Praat runs into a line it can't interpret, it will crash + output an error

Use Praat GUI to help you learn the syntax

Whitespace ignored, but helpful to you

Comments

Comments:

- notes to user that Praat will ignore (not try to run)
- Statements you want Praat to ignore for now

Must start with “#” and appear on separate lines OR

Start with “;” and

```
# This is a comment and won't be run
```

```
; This is a comment and won't be run
```

```
my_num = 2; This is a comment and won't be run
```

Understanding Praat errors

Description of error + line number where the script crashed

Look at the line in your script: Search >> Go to line... (or cmd + L)

Common sources of error:

- Incorrect capitalization
- Incorrect use of quotes
- Incorrect type of object selected

Types of errors

- Syntactic errors
 - Typos, incorrect use of quotes, etc.
- Runtime errors
 - Impossible for Praat to do as told: file doesn't exist, command doesn't exist for that object type, etc.
 - Object types: Sound, TextGrid, Strings, etc ([see more about Praat Object types](#))
- Semantic errors
 - Praat does what you tell it, but the output is not what you want
 - Sneaky! Doesn't produce an error! Need to frequently test your code to avoid these!

See more: [See https://www.linguisticsociety.org/sites/default/files/e-learning/Day1.pdf](https://www.linguisticsociety.org/sites/default/files/e-learning/Day1.pdf)

println statements in Praat: Get Praat to tell you things

println: special type of function in Praat that prints whatever follows it to the Praat info window

To print...	Do...	Example code
Verbatim text	Just type it!	<code>println hello world</code>
Variables	Use single quotes	<code>println 'filename\$'</code> <code>println The value is 'my_num'</code>
Numeric variables with limited place decimals	Limit decimals with ":"	<code>println The value is 'my_num:4'</code>

NOTE: `writeInfoLine: "some text"` and `appendInfoLine: "more text"` do similar things & reflect more recent syntax:
https://www.fon.hum.uva.nl/praat/manual/Scripting_6_2_Writing_to_the_Info_window.html

Error example: Unknown variable

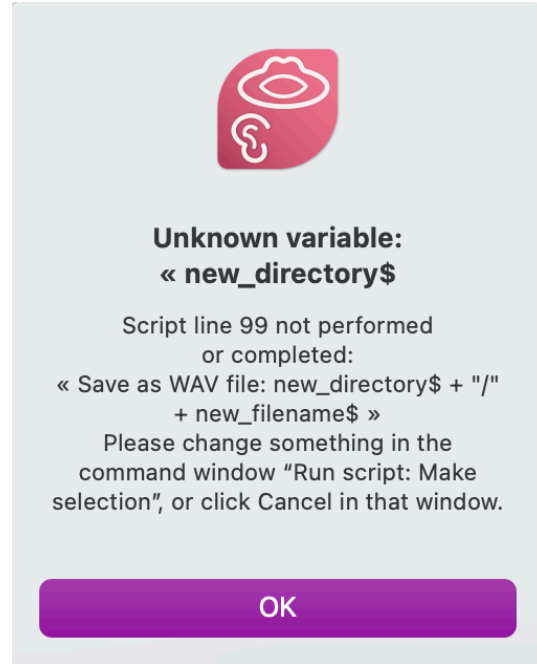
Forgot to assign the variable
“new_directory\$” which Praat
tried to use on line 99

OR

Assigned it but left off the “\$”

OR

Assigned it but with a typo in
the name



How to debug:

Go to line error in Script
(99): Search >> Go to
line

Look further up for
where that variable was
created

Use printline statements
earlier if you're unsure
what it contains

Referring to variables

When referring to variables as arguments in Praat commands, no quotes

```
Read from file: current_file$
```

When referring to variables in built-in Praat functions, single quotes needed

```
selectObject: "Sound" + 'filename$'  
printline 'filename$'
```

Verbatim arguments in double quotes (usually)



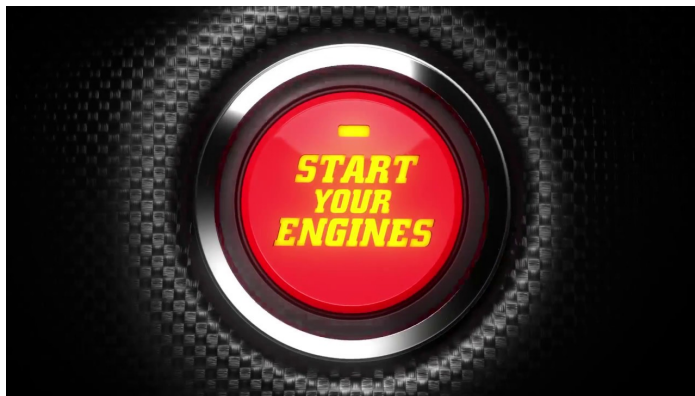
Hands-on

Hands-on

Let's try out some commands

1. Do some stuff in Praat by clicking menu items/buttons
2. Print our history to see how these actions should be coded
3. Turn them into a script

Everybody: Open Praat



Get duration of .wav file

- Open one of the .wav files in data/ in Praat
- Get its duration using the Query menu

The image shows the Praat software interface. The main window is titled 'Praat Objects' and contains a list of objects. The object '548. Sound cas4_28_1_1' is selected. A context menu is open over this object, showing the 'Query' menu. The 'Query' menu is expanded to show the 'Query time domain' submenu, which is further expanded to show the 'Get total duration' option. A separate window titled 'Praat Info' is open, displaying the duration of the selected object: '4.073922902494331 seconds'.

Praat Objects

New Open Save Help

Objects:

548. Sound cas4_28_1_1

Sound help

View & Edit

Play

Draw -

Query -

- Query time domain > Get start time
- Get end time
- Get total duration
- Get number of channels
- Query time sampling >
- Get value at time...
- Get value at sample number...
- Get minimum...
- Get time of minimum...
- Get maximum...
- Get time of maximum...
- Get absolute extremum...
- Get nearest zero crossing...
- Get nearest level crossing...
- Get mean...
- Get root-mean-square...
- Get standard deviation...
- Get energy...
- Get power...
- Get energy in air
- Get power in air
- Get intensity (dB)

Rename... Copy...

Inspect Info

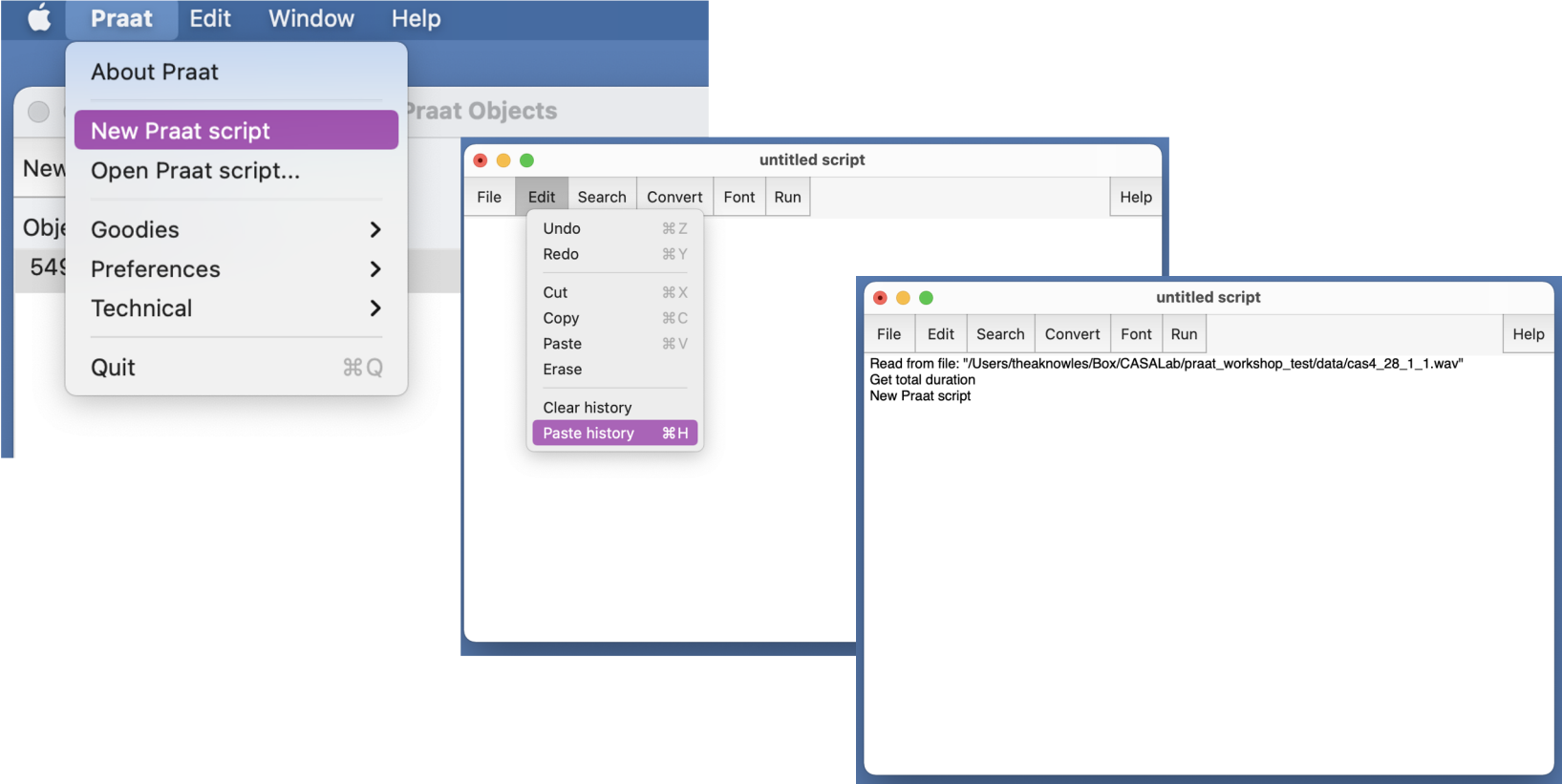
Remove

Praat Info

File Edit Search Convert Font Help

4.073922902494331 seconds

Print command history in new Praat scripting window



Save as test.praat & run!

- We will be testing out our code a LOT
- Testing:
 - Running code
 - Breaking code (often intentionally)

Save as test.praat and run the “script”

2 minutes

Now we'll edit your script by doing the following...

1. Delete the “New Praat Script” command
2. Store the “total duration” to a variable named “total_dur”
`total_dur = Get total duration`
3. Add a line to print the value of “total_dur” to the Praat info window using printline
`printline 'total_dur'`
4. Modify the printline statement to be more informative and only print 4 digits
`printline The total duration is: 'total_dur:4'`
5. Add a comment somewhere
`# Like this.`

What if we want to get the duration of all the .wav files in our folder?

- This is where being able to write scripts shines
- We can **iterate** our code with a **for-loop**

- For-loops
 - Special syntax
 - Allow us to tell Praat “repeat this code n times”
 - Other kinds of loops: while-loops, until-loops

```
for counter from min to max  
  
    # code to repeat here  
  
endfor
```


Let's level up our script to iterate over all the .wav files in our folder

- But first... boilerplate code

Boilerplate code

i.e. your starting template

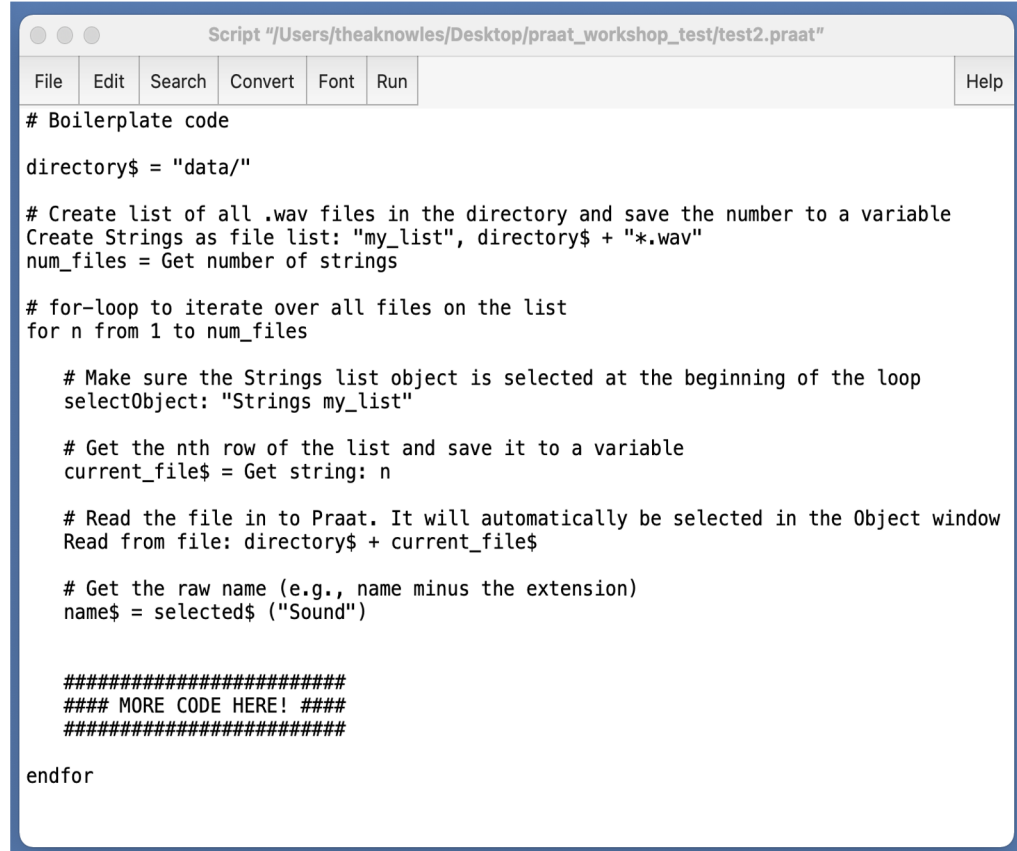
Code that can be used again and again and again
without changing much

Level up: Using a boiler plate structure

Open boilerplate_code1.praat

Goals:

1. Add our code to the boiler plate code and get the duration of all .wav files in our folder
2. Print information to ourselves about the number of files



```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run Help
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n
    # Read the file in to Praat. It will automatically be selected in the Object window
    Read from file: directory$ + current_file$
    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")
    #####
    #### MORE CODE HERE! ####
    #####
endfor
```

Comments

Start with #

Won't get
evaluated by Praat

```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run Help
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
# Make sure the Strings list object is selected at the beginning of the loop
selectObject: "Strings my_list"
# Get the nth row of the list and save it to a variable
current_file$ = Get string: n
# Read the file in to Praat. It will automatically be selected in the Object window
Read from file: directory$ + current_file$
# Get the raw name (e.g., name minus the extension)
name$ = selected$ ("Sound")
#####
#### MORE CODE HERE! ####
#####
endfor
```

Define path to files

Define the directory path

Paths can be:

- Relative to where the script lives
- Absolute

Must be text

- Variable name ends in "\$"
- Value in ""
- Paths must end with backslash /

```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n
    # Read the file in to Praat. It will automatically be selected in the Object
    Read from file: directory$ + current_file$
    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")
    #####
    #### MORE CODE HERE! ####
    #####
endfor
```

Get list of files

Create list of all
.wav files in
directory and count
them

“Strings as file list”
is a special Praat
Strings object used
for this purpose

```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n
    # Read the file in to Praat. It will automatically be selected in the Object window
    Read from file: directory$ + current_file$
    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")
    #####
    #### MORE CODE HERE! ####
    #####
endfor
```

For-loop

Purpose: Iterate over all files on the list and do stuff to each one

Special syntax

```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n
    # Read the file in to Praat. It will automatically be selected in the Object list
    Read from file: directory$ + current_file$
    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")
    #####
    #### MORE CODE HERE! ####
    #####
endfor
```

For-loop structure:

```
for counter from min to max
```

```
...
```

```
endfor
```

n = arbitrary variable name for counter. +1 each time through the loop

1 = starting point

num_files: number of .wav files in our directory

All code within for-loop should be indented: easier to read/debug

```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n
    # Read the file in to Praat. It will automatically be selected in the Object list
    Read from file: directory$ + current_file$
    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")
    #####
    #### MORE CODE HERE! ####
    #####
endfor
```


selectObject:

Equivalent to you clicking on the object in the Praat Object window

Built in Praat function

Argument: object type & name as appears in Object window

Important: You must have the correct object selected in order to issue the right commands!

Otherwise: error

Test: try commenting out and see what happens

```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n
    # Read the file in to Praat. It will automatically be selected in the Object window
    Read from file: directory$ + current_file$
    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")
    #####
    #### MORE CODE HERE! ####
    #####
endfor
```

selectObject: "Strings my_list"
=
select Strings my_list

Get string:

Get the nth row of the list of filenames

Praat command (button) available in Object window when a Strings object is selected

Save it to a variable. Notice...

- Because it's text, it is a string (text) variable, so variable name must end in "\$"
- Argument of "Get string:" the position you want to look at in the list (numeric)

- "n" is our counter variable

Script `"/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"`

File Edit Search Convert Font Run

```
# Boilerplate code
directory$ = "data/"

# Create list of all .wav files in the
Create Strings as file list: "my_list",
num_files = Get number of strings

# for-loop to iterate over all files on the list
for n from 1 to num_files

    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"

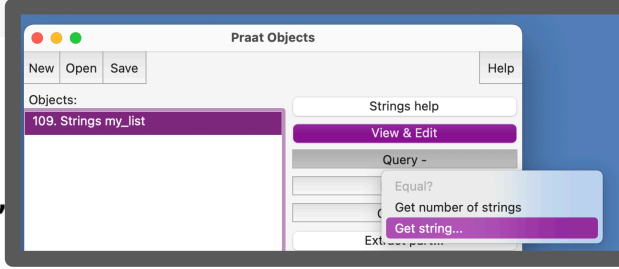
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n

    # Read the file in to Praat. It will automatically be selected in the Object window
    Read from file: directory$ + current_file$

    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")

    #####
    #### MORE CODE HERE! ####
    #####

endfor
```



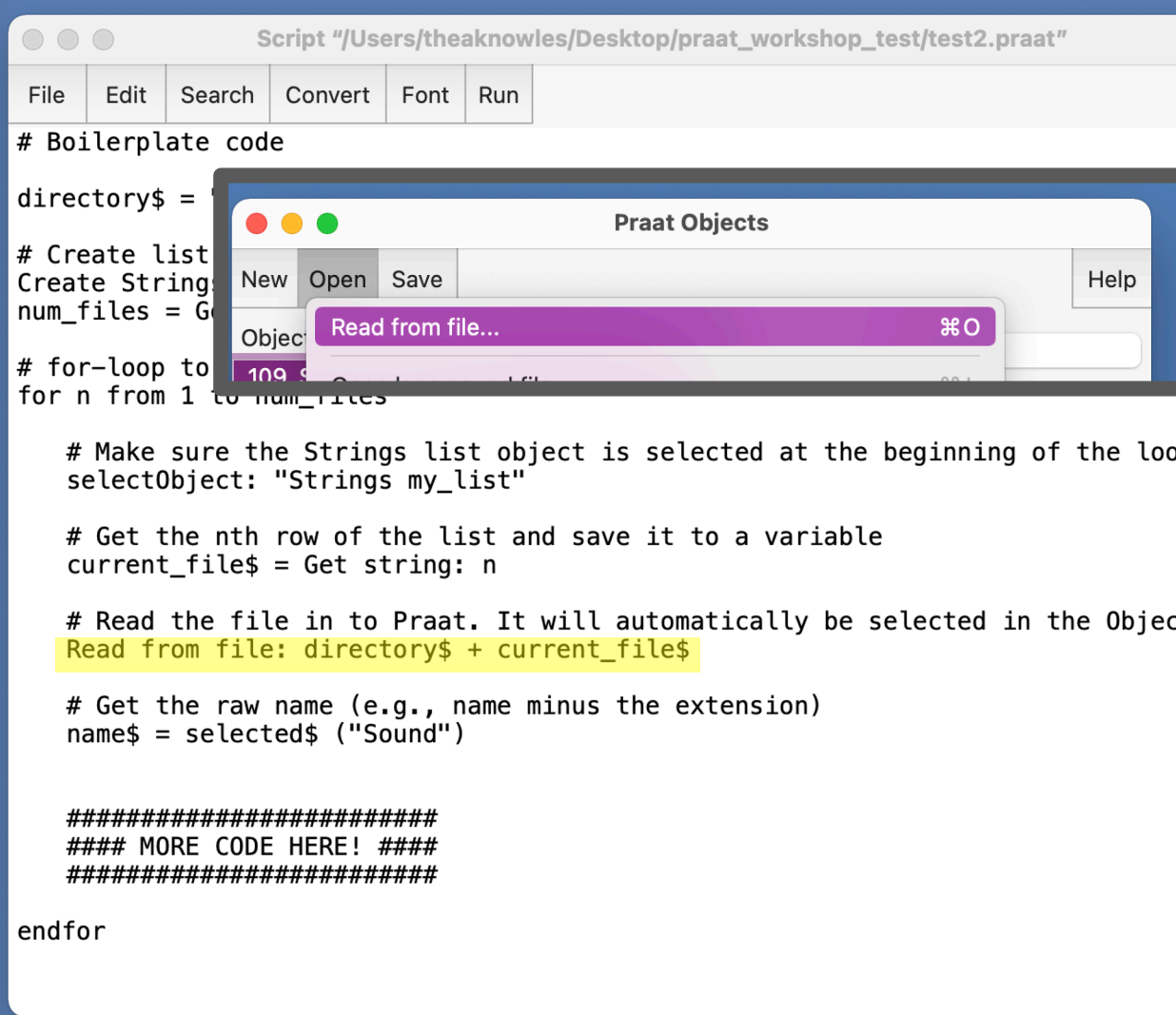
The screenshot shows the Praat Objects window with a list of objects. The object "109. Strings my_list" is selected. A context menu is open over the selected object, showing options like "View & Edit", "Query -", "Equal?", "Get number of strings", and "Get string...". The "Get string..." option is highlighted in purple.

Read from file:

Open the file in Praat

Argument: path to file

- This is the value stored in our variable "directory\$"!



The screenshot shows a Praat script editor window titled "Script '/Users/theaknowles/Desktop/praat_workshop_test/test2.praat'". The menu bar includes File, Edit, Search, Convert, Font, and Run. The script content is as follows:

```
# Boilerplate code
directory$ =
# Create list
Create String
num_files = G
# for-loop to
for n from 1 to num_files

# Make sure the Strings list object is selected at the beginning of the loop
selectObject: "Strings my_list"

# Get the nth row of the list and save it to a variable
current_file$ = Get string: n

# Read the file in to Praat. It will automatically be selected in the Object
Read from file: directory$ + current_file$

# Get the raw name (e.g., name minus the extension)
name$ = selected$ ("Sound")

#####
#### MORE CODE HERE! ####
#####

endfor
```

An "Open" menu is visible, with a "Read from file..." option highlighted in purple. The "Read from file: directory\$ + current_file\$" line in the script is highlighted in yellow.

Get name of Object

Once the file is read into Praat it will have an object *type* and a *name*

- Name is the same as the original filename, minus the extension

```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
    # Make sure the Strings list object is selected at the beginning of the loop
    selectObject: "Strings my_list"
    # Get the nth row of the list and save it to a variable
    current_file$ = Get string: n
    # Read the file in to Praat. It will automatically be selected in the Object window
    Read from file: directory$ + current_file$
    # Get the raw name (e.g., name minus the extension)
    name$ = selected$ ("Sound")
    #####
    #### MORE CODE HERE! ####
    #####
endfor
```



Hands-on

Goals:

Level 1

Modify boilerplate_code1.praat to print the following information to the screen:

1. Get number of .wav files in a directory
2. Print out individual file names
3. Get duration of each .wav file

Cheat sheets/Answers:

- cheat_sheet_level1.praat

Level up: Modify code to:

Level 2: Get total duration of all .wav files

Level 3: Copy all .wav files ending in "_3" to a new directory and append "_copy" to the file name

Level 4: Toggle option to "clean up" as you go

Level 5: Add a form for user input

Cheat sheets:

- boilerplate_code2.praat
- cheat_sheet_workshop1.praat

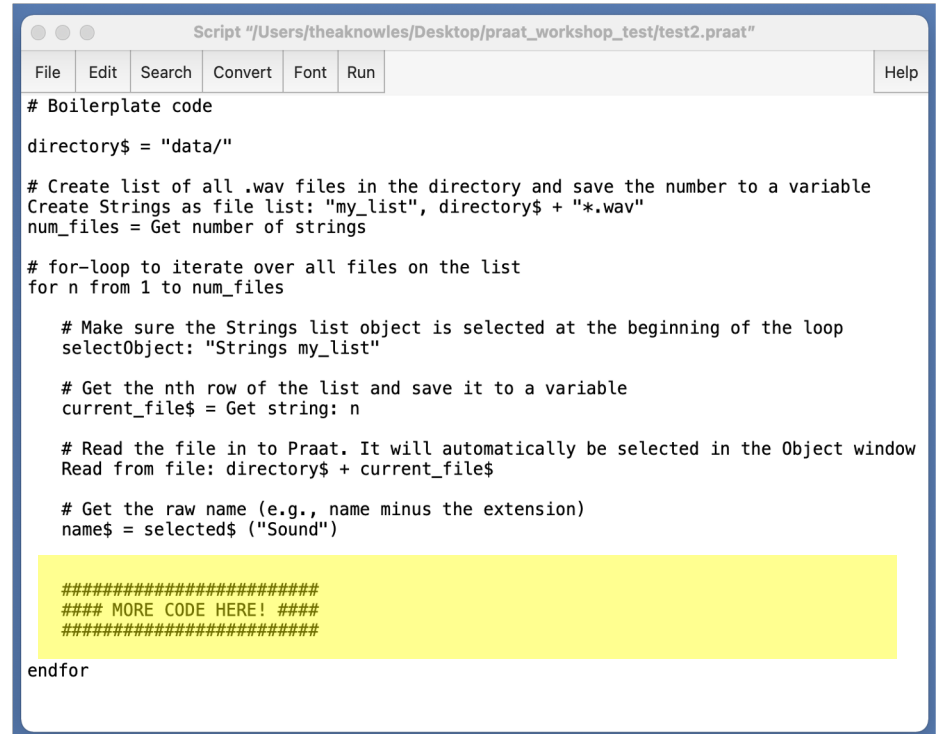
Using boilerplate_code1.praat & your Praat history, get the duration of each .wav file in the directory

Save script as something new (e.g., test.praat) before editing.

Save it in the same location as the “data” folder!

Hint: Get the total duration of the Sound object that’s named ‘name\$’

Use printline to print it to the screen



```
Script "/Users/theaknowles/Desktop/praat_workshop_test/test2.praat"
File Edit Search Convert Font Run Help
# Boilerplate code
directory$ = "data/"
# Create list of all .wav files in the directory and save the number to a variable
Create Strings as file list: "my_list", directory$ + "*.wav"
num_files = Get number of strings
# for-loop to iterate over all files on the list
for n from 1 to num_files
  # Make sure the Strings list object is selected at the beginning of the loop
  selectObject: "Strings my_list"
  # Get the nth row of the list and save it to a variable
  current_file$ = Get string: n
  # Read the file in to Praat. It will automatically be selected in the Object window
  Read from file: directory$ + current_file$
  # Get the raw name (e.g., name minus the extension)
  name$ = selected$ ("Sound")
  #####
  #### MORE CODE HERE! ####
  #####
endfor
```

Debugging

- A few different ways to do it
 - Force crash
 - Pause windows
 - Run selection
- My personal favorites:
 - Include printline statement with variables to inspect & then **force script to crash**
- Other tips:
 - Go to the line number in the script (Search >> Go to line..." or Cmd + L on Mac)
 - Google the literal error

Test: Force Praat to crash in order to inspect your code

- Useful prinline statement of variables I want to inspect followed "x"
 - Praat will get to the "x" and crash
 - At this point I can see what's in the Object window, what the most recent variable is... etc.

The screenshot shows the Praat software interface. The main window displays a script titled "Script '/Users/theaknowles/Desktop/praat_workshop_test/test2.praat'". The script contains the following code:

```
# Boilerplate code
directory$ = "data/"

# Create list of all .wav files in the directory
Create Strings as file list: "my_list"
num_files = Get number of strings

# for-loop to iterate over all files
for n from 1 to num_files

  # Make sure the Strings list object is selected
  selectObject: "Strings my_list"

  # Get the nth row of the list and store it in a variable
  current_file$ = Get string: n

  # Read the file in to Praat. It will automatically load the file
  Read from file: directory$ + current_file$

  # Get the raw name (e.g., name minus the extension)
  name$ = selected$ ("Sound")

  prinline The current file name is: 'name$'
  x
endfor
```

An error dialog box is displayed in the center, titled "Unknown variable:", with the message "Script line 25 not performed or completed: « x »" and "Menu command 'Run' not completed." Below the dialog box, the Praat Info window shows the following output:

```
4.073922902494331 seconds
The total duration is: 4.0739
The current file name is: cas4_1367_10_3
```

The line `prinline The current file name is: 'name$'` in the script is highlighted in yellow, and the corresponding output line in the Praat Info window is highlighted in green.

What kind of error is this?

Goals:

5 minutes

Level 1

Modify boilerplate_code1.praat to print the following information to the screen:

1. Get number of .wav files in a directory
2. Print out individual file names
3. Get duration of each .wav file

Cheat sheets/Answers:

- [cheat_sheet_level1.praat](#)

Level up: Modify code to:

Level 2: Get total duration of all .wav files

Level 3: Copy all .wav files ending in "_3" to a new directory and append "_copy" to the file name

Level 4: Toggle option to "clean up" as you go

Level 5: Add a form for user input

Cheat sheets:

- [boilerplate_code2.praat](#)
- [file_management.praat](#)

“Answers”

(i.e., there's way more than 1 "correct" way to do these things,
but here's what I did)

Review cheat_sheet_level1.praat together

Level 2: Get total duration of all .wav files

Hint: set up a dummy numeric variable (e.g., `total_dur`) as a counter and add to it on each loop

Think: Where in relationship to for-loop should this happen?

```
# LEVEL 2 (part 1): set up empty numeric variable to sum .wav duration
total_dur = 0

for n from 1 to num_files
  ...
  # LEVEL 2 (part 2): Get total duration of files
  total_dur = total_dur + dur

endfor

# LEVEL 2 (part 3): print with 4 decimal places
printline 'total_dur:4'
```

Level 3: Copy all .wav files ending in "_3" to a new directory and append "_copy" to the file name

New ingredients:

- String functions to get information from file name
 - https://www.fon.hum.uva.nl/praat/manual/Formulas_6_String_functions.html
- Condition (if-statement)
 - <https://praascripting.lingphon.net/conditionals-1.html>
- Save as WAV file

Level 3: Copy all .wav files ending in "_3" to a new directory and append "_copy" to the file name

```
new_directory$ = directory$ + new_folder_name$
...
for n from 1 to num_files
  ...

  # LEVEL 3: Copy all .wav files ending in _3 to new directory & rename
  # Get last character of filename
  # Source: https://www.fon.hum.uva.nl/praat/manual/Formulas\_6\_\_String\_functions.html
  suffix$ = right$ (filename$,1)

  if suffix$ == "3"
    select Sound 'filename$'
    new_filename$ = filename$ + "_copy.wav"

    # Sanity check by using printline
    printline 'filename$' suffix: 'suffix$'

    # Save file to new location
    Save as WAV file: new_directory$ + "/" + new_filename$
  endif
endfor
```

Level 4: Toggle option to "clean up" as you go

- New ingredients
- Boolean variables (TRUE/FALSE or 1/0)
- Conditional (if-statement)
- Built-in functions:
 - select all, minus, clearinfo
- “Remove” command

Level 4: Toggle option to "clean up" as you go

First, without toggling

```
clearinfo

for n from 1 to num_files
    ...

    select all
    minus Strings my_list
    Remove
endfor

select all
Remove
printline All finished!
```

Toggle option challenge:

- Create Boolean variable
 clean_up = 1
- Nest your clean up code in if-statements
(note syntax)
 if clean_up == 1
 clearinfo
 endif

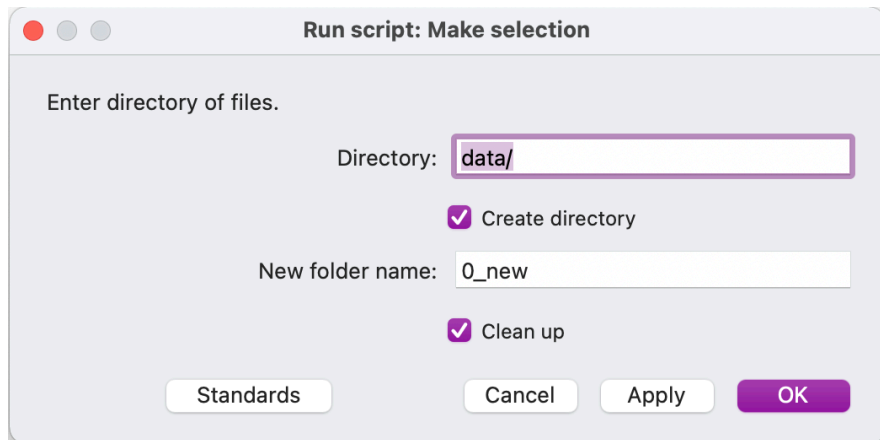
Level 5: Add a form for user input

- Forms: special code chunks that will generate a user input form for the user to provide arguments to the script
- Special syntax & behavior
 - Must declare variable type, name, value differently than in script
 - https://www.fon.hum.uva.nl/praat/manual/Scripting_6_1_Arguments_to_the_script.html
 - <https://praascripting.lingphon.net/simpleinput-1.html>
- My habits:
 - I usually include a form at the beginning of my script to enter the directory information
 - For scripts I run myself, I usually don't do much with the form, but I like to have it as a secondary buffer between me and the rest of the script (“Did I actually mean to Run this?”)
 - For scripts other people run, I include many other form variables
 - Annotator ID, task option, verbose option...

Level 5: Add a form for user input

```
# LEVEL 5: Forms
# Forms are special kinds of code
blocks in Praat
form Make selection
    comment Enter directory of files.
    sentence Directory data/
    boolean Create_directory 1
    sentence New_folder_name 0_new
    boolean Clean_up 1
endform
```

This is what pops up when you run the script:



Additional tips

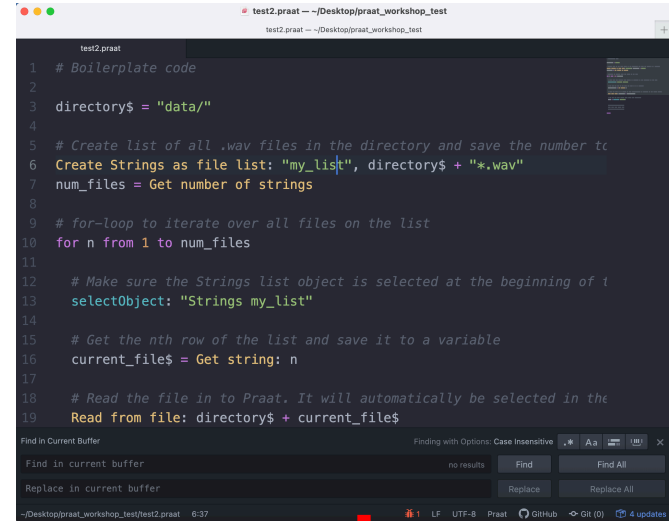
Script editors

Options I like:

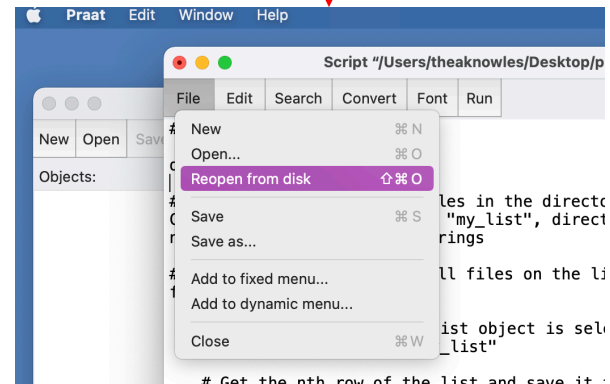
- [Atom](#) + [Praat syntax](#)
- [BBEdit](#) + [Praat syntax](#)

My routine:

- Edit in text editor of choice (have script also open in Praat)
- Switch to Praat script in Praat, and do “Reopen from disk”
- Run



```
test2.praat
1 # Boilerplate code
2
3 directory$ = "data/"
4
5 # Create list of all .wav files in the directory and save the number to
6 Create Strings as file list: "my_list", directory$ + "*.wav"
7 num_files = Get number of strings
8
9 # for-loop to iterate over all files on the list
10 for n from 1 to num_files
11
12 # Make sure the Strings list object is selected at the beginning of 1
13 selectObject: "Strings my_list"
14
15 # Get the nth row of the list and save it to a variable
16 current_file$ = Get string: n
17
18 # Read the file in to Praat. It will automatically be selected in the
19 Read from file: directory$ + current_file$
```



Other scripting resources

- <https://praascripting.lingphon.net/>
- <https://eleanorchodroff.com/tutorial/PraatScripting.pdf>
- http://www.mauriciofigueroa.cl/04_scripts/Praat_scripting_manual_0.1.8_04.pdf
- <https://www.linguisticsociety.org/sites/default/files/e-learning/Day1.pdf>
- Subscribe to Praat Users group: <https://groups.io/g/Praat-Users-List>

Other people's scripts

- <http://www.acsu.buffalo.edu/~cdicianio/scripts.html>
- <https://lennes.github.io/spect/>
- https://github.com/stylerw/styler_praat_scripts
- <http://www.mattwinn.com/praat.html>
- <https://github.com/thealk/PraatScripts>