UK-South Korea Prosody Research Network

4 July 2022
1:30-17:30

Praat Scripting Basic

Tae-Jin Yoon
Dept. of English Language and Literature
Sungshin Women's University

Objectives

1. How to write and read Praat scripting
2. How to extract acoustic features

강의안내

01 Basic

02 Formants

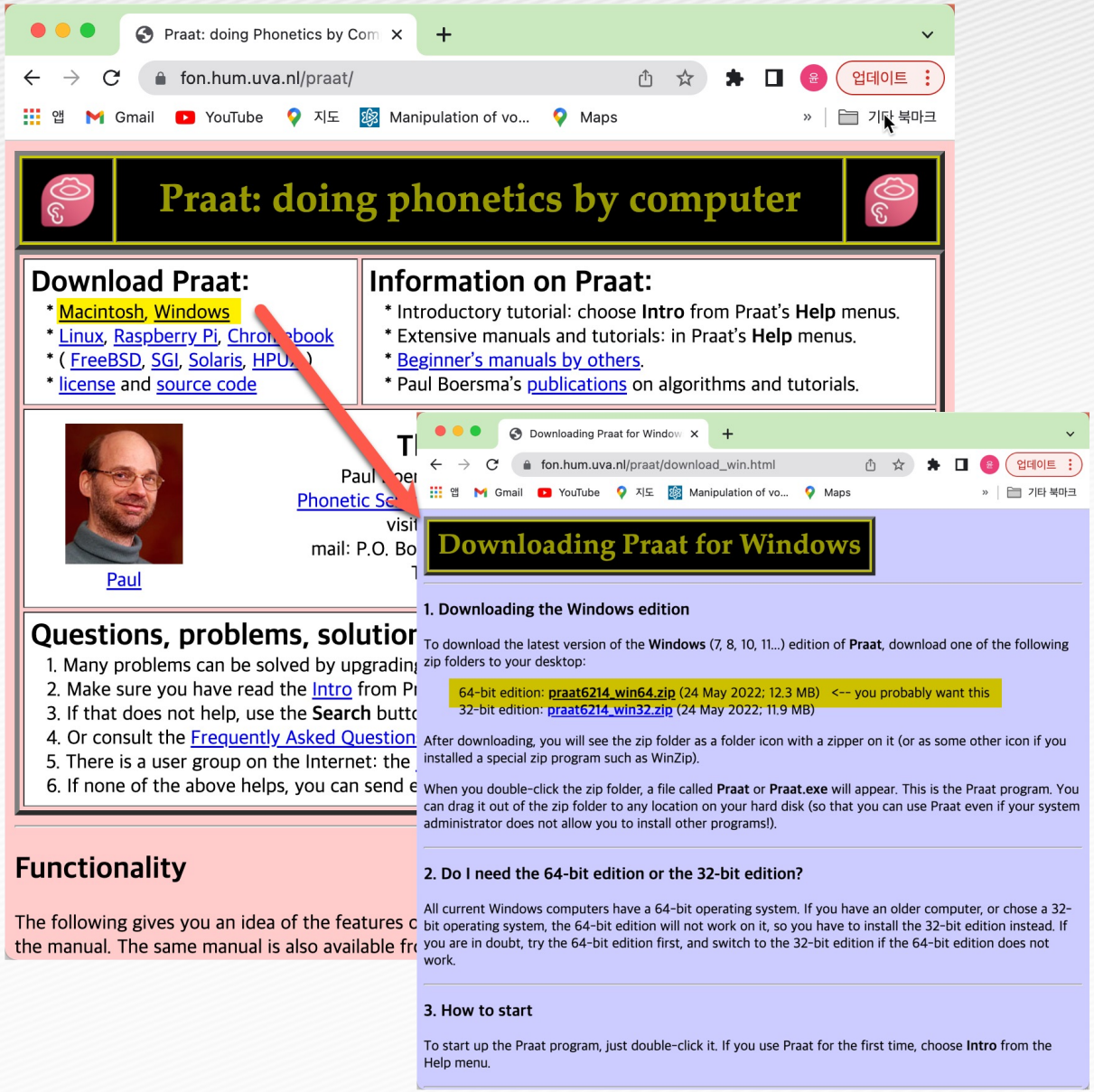
03 Multiple Files in a directory

04 Other acoustic features (e.g. F0)

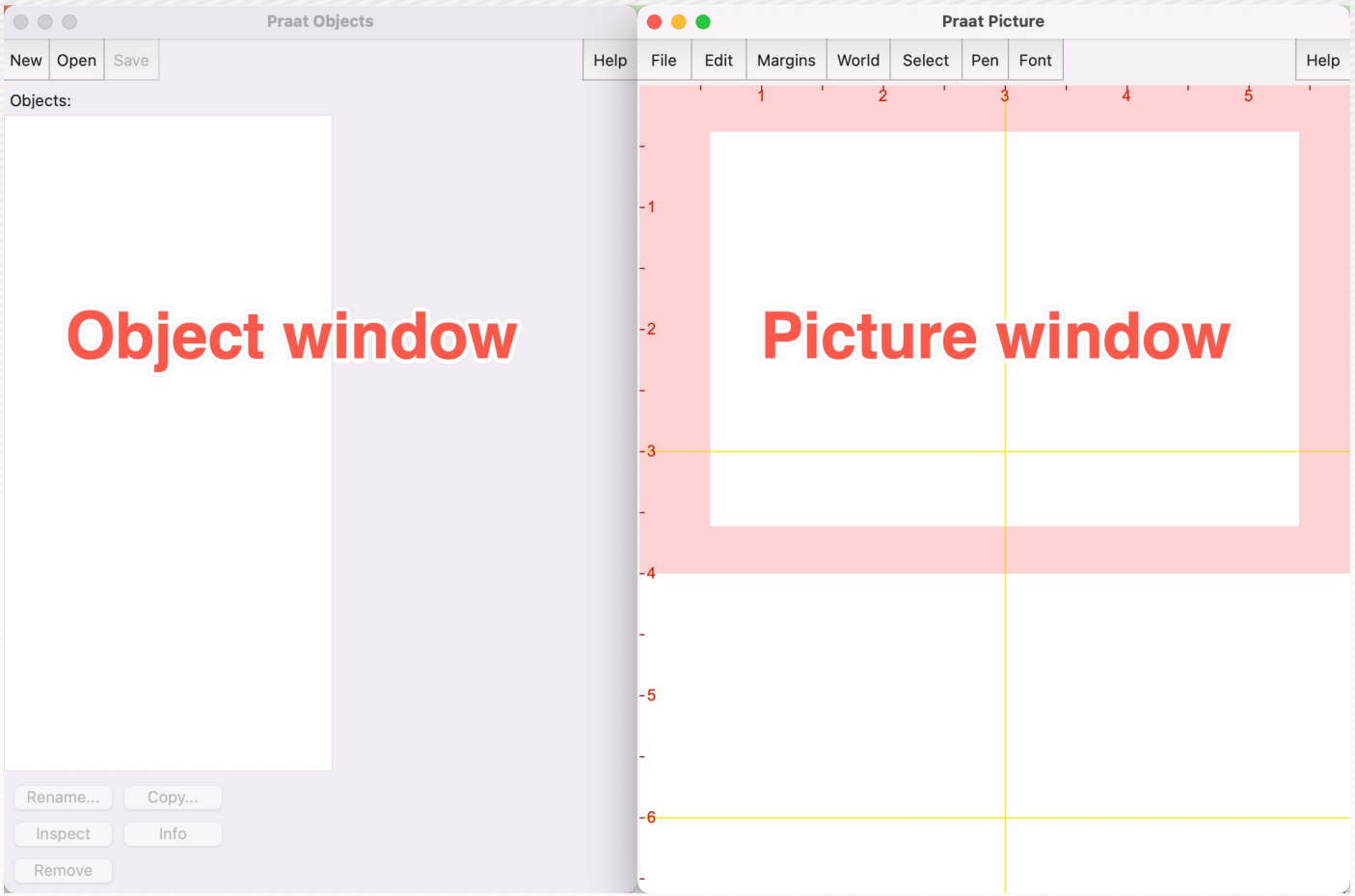
01

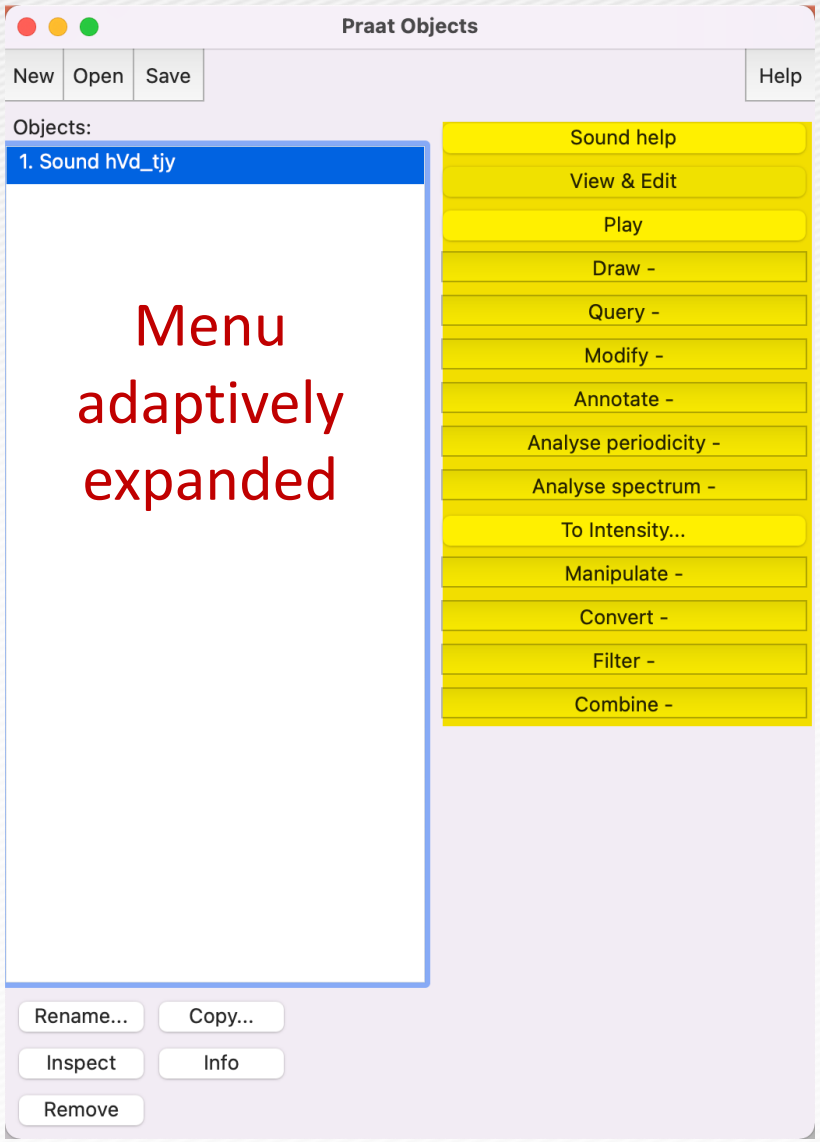
Praat

Installation



Structure


Praat Objects
New
Open
Save
Help
Objects:
Object window
Rename...
Copy...
Inspect
Info
Remove
Praat Picture
File
Edit
Margins
World
Select
Pen
Font
Help
Picture window


Praat Objects
New
Open
Save
Help
Objects:
1. Sound hVd_tjy
Menu adaptively expanded
Sound help
View & Edit
Play
Draw -
Query -
Modify -
Annotate -
Analyse periodicity -
Analyse spectrum -
To Intensity...
Manipulate -
Convert -
Filter -
Combine -
Rename...
Copy...
Inspect
Info
Remove

02

Scripting

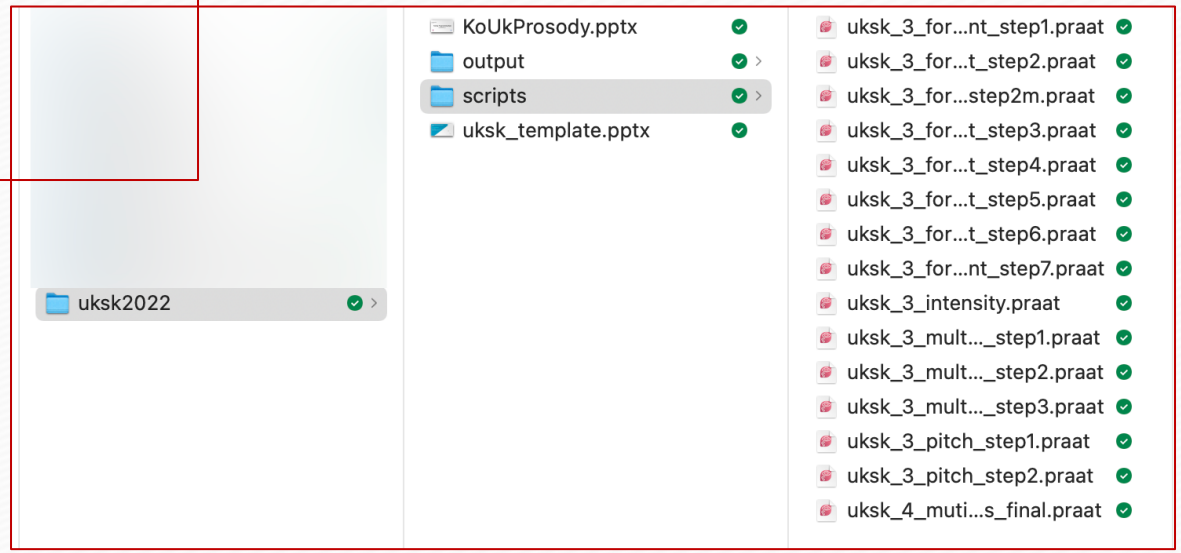
Scripting

Save time and effort by automating a sequence of operations

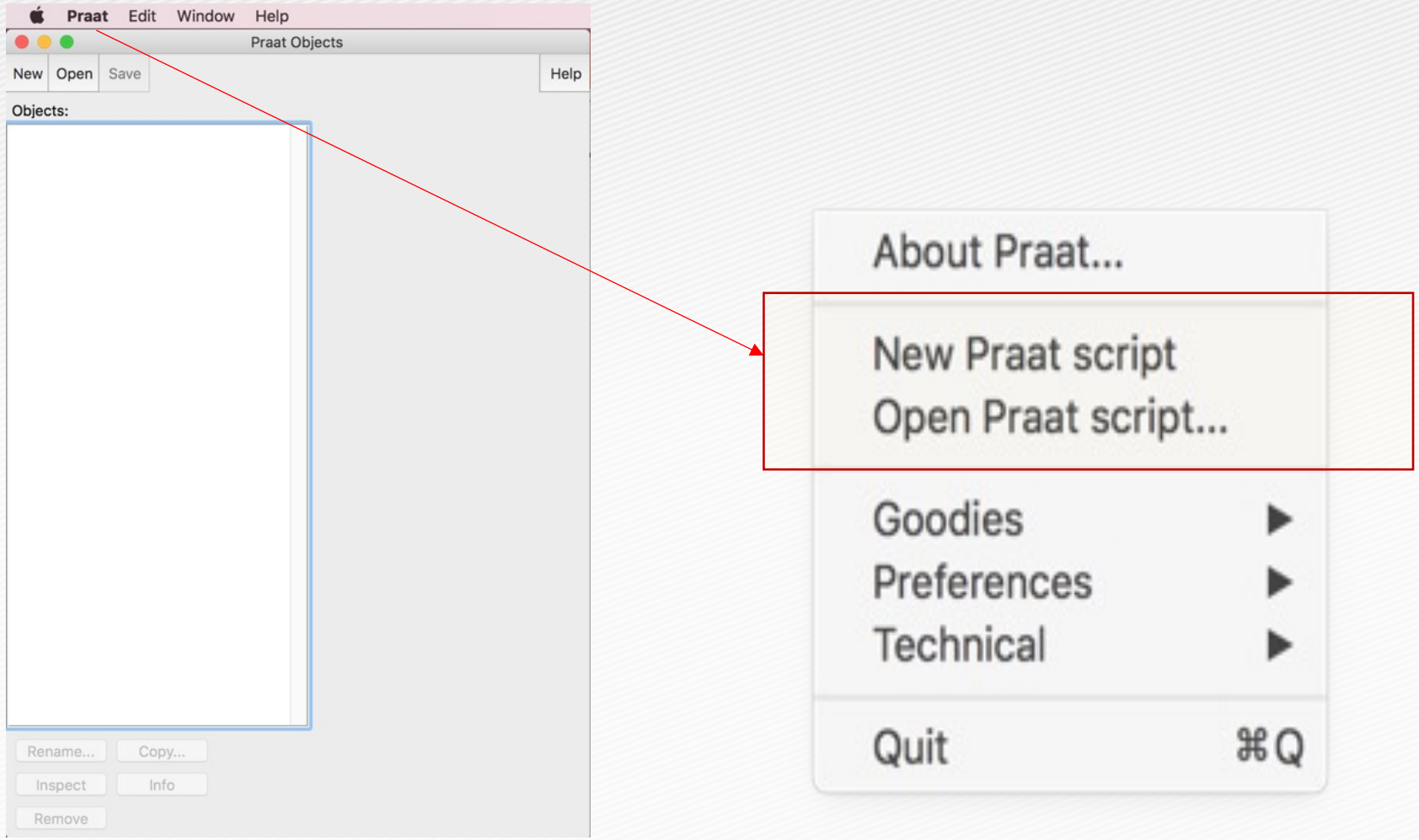
Directory structure

```
uksk2022/
      data/
      output/
      scripts/myScript01.praat
                  myScript02.praat
            ...
```

cf.
.praat
.psc

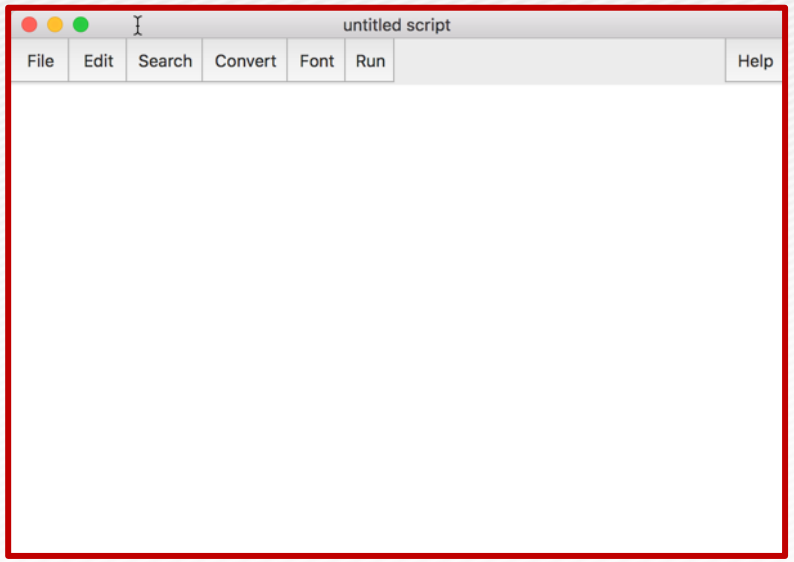


Basics: opening the scripting editor



New vs. existing scripts

Praat > New Praat Script



Praat > Open Praat Script...

Script "/Users/tyoon/Dropbox/work/exphon/uksk2022/uksk_4_mutiple_files_fina...

File Edit Search Convert Font Run Help

```
form Measure phonetic features for segments in a textgrid
    sentence sound_dir ./sound
    sentence output result.txt
endform


# Add names for F1 and F2
writeInfoLine: "Basename",tab$,"phone",tab$,"word",tab$,"phone_duration
   ... "word_duration",tab$,"F0",tab$,"F0st",tab$,"F1",tab$,"F2",tab$,"

# to the output file
writeFileLine: output$, "Basename",tab$,"phone",tab$,"word",tab$,"phone
   ... "word_duration",tab$,"F0",tab$,"F0st",tab$,"F1",tab$,"F2",tab$,"


strings = Create Strings as file list: "list", sound_dir$ +"/"+ "*.wav"
numberOfFiles = Get number of strings

;writeInfoLine: "Number of wave files: ", numberOfFiles

for ifile to numberOfFiles
    selectObject: strings
    sound_file$ = Get string: ifile
```

Hello World!

old: command....
new: command:

- **W**rite in the Script window

```
writeInfoLine: “Hello World!”
```

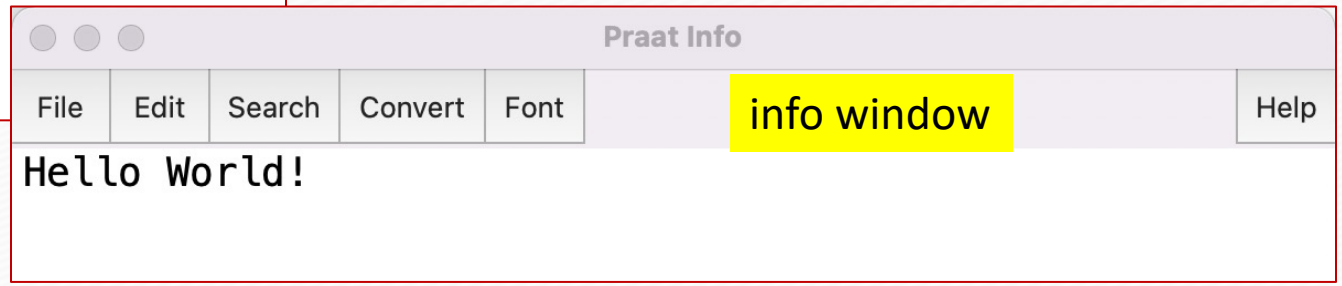
- **S**ave

```
File > Save
```

⇒ `uksk_1_hello_world.praat`

- **R**un

```
Run > Run
```



uksk_1_hello_world.praat

Info window

- clearinfo
- wirteInfo
- appendInfo
- writeInfoLine
- appendInfoLine

Example

```
##################################################
# This is my first Praat script.
# (c) Tae-Jin Yoon
# Created: July 1, 2022
##################################################

# clear messages in the info window
clearinfo

;writeInfoLine: “Hello World!”
appendInfoLine: ”Hello World!”

myNumber = 16
appendInfoLine: myNumber
```

1. 곧은 따옴표	2. 둥근 따옴표
1. '가나다'	1. ’가나다’
2. "가나다"	2. ”가나다”

ERROR

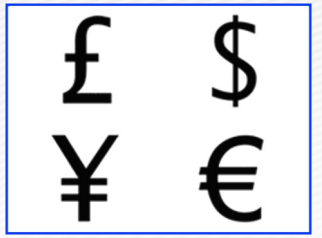
variable name

03

Variables

Numeric variables

- variable names begin with a lowercase letter
- No space or special characters (such as accents or pound signs) inside a variable name
- case sensitive (rollsRoyce ≠ rollsroyce).




```
age = 34
beers_drunk = 4
beersDrunk = 4
```

A folder that has parentheses and spaces

- Avoid minus(-), parentheses, colons, semicolons, and spaces in the directory or variable names

  01 bank (46;3)

- Use underline(_) or camel case instead
  - 01_bank_46_3.wav
  - CamelCase_3.wav

Numeric variables

```
clearinfo

# Calculating calories
breakfastCarolies = 553
lunchCarolies = 762
dinnerCalories = 976
totalCalories = breakfastCarolies + lunchCarolies
                ... + dinnerCalories
appendInfoLine: totalCarolies
```

continuation line

String variables

- String variable names end with $(dollar sign
- “double quotes”: the values of a string variable

```
clearinfo
name$ = "Dr. Kim"
;sentence$ = "Dr. Kim said "Hello World!""
;appendInfoLine: sentence$

sentence$ = "Dr. Kim said ""Hello World!"""
appendInfoLine: sentence$

;sentence$ = 'Dr. Kim said "Hello World!"'
appendInfoLine: sentence$
```

Which one is incorrect?

1. YearBirth
2. yEARrESIDING
3. studentid
4. symbol in IPA
5. f0_and_formants
6. f0AndFormants
7. 2ndFormant
8. name$

1. Incorrect 2. Correct 3. Correct 4. Incorrect 5. Correct 6. Correct 7. Incorrect 8. Correct.

Some common string operations

- Combining (concatenating) and subtracting strings

```
wavefile$ = ”mysound01.wav"

textgridfile$ = wavefile$ - ".wav" + ".TextGrid"

appendInfoLine: textgridfile$
```

Some common string operations

- **Non-printing (invisible) characters**

```
clearinfo

word1$ = "Formant1"
word2$ = "Formant2"

strConcat$ = word1$ + word2$
appendInfoLine: strConcat$

tabSepStr$ = word1$ + tab$ + word2$
appendInfoLine: tabSepStr$
```

output

```
Formant1Formant2
Formant1	Formant2
```

uksk_6_tab.praat

… for continuation

The three dots (...) tell Praat that the command continues on the next line

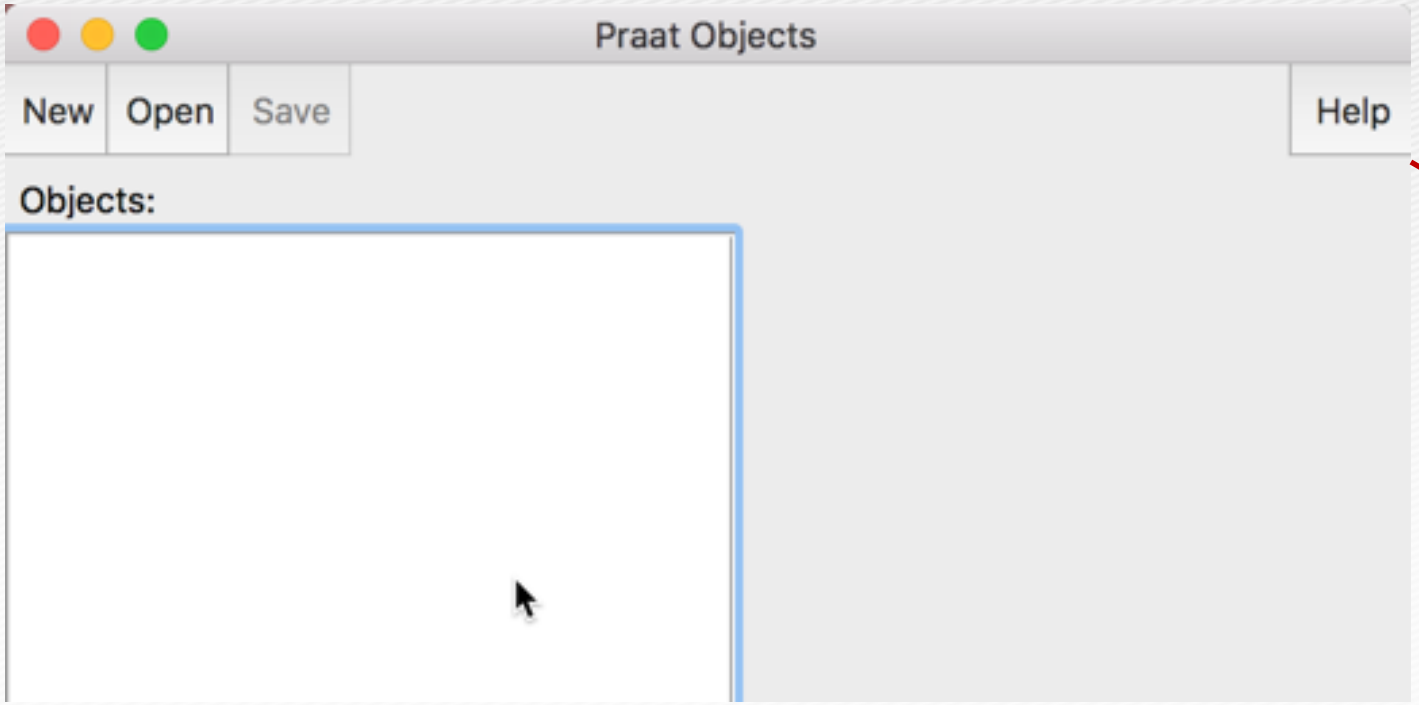
```
headerRow$ = "subject" + tab$
           ... + "date" + tab$
           ... + "vowel" + tab$
           ... + "f1" + tab$
           ... + "f2" + newline$

# later write out rows in the same format

appendInfoLine: headerRow$
```

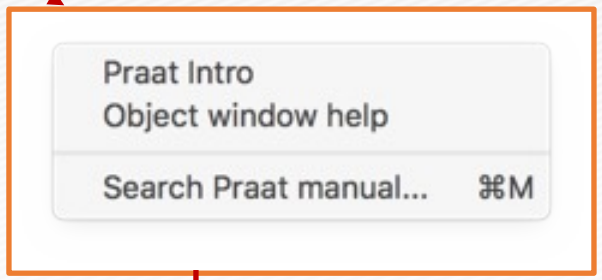
Which one is incorrect?

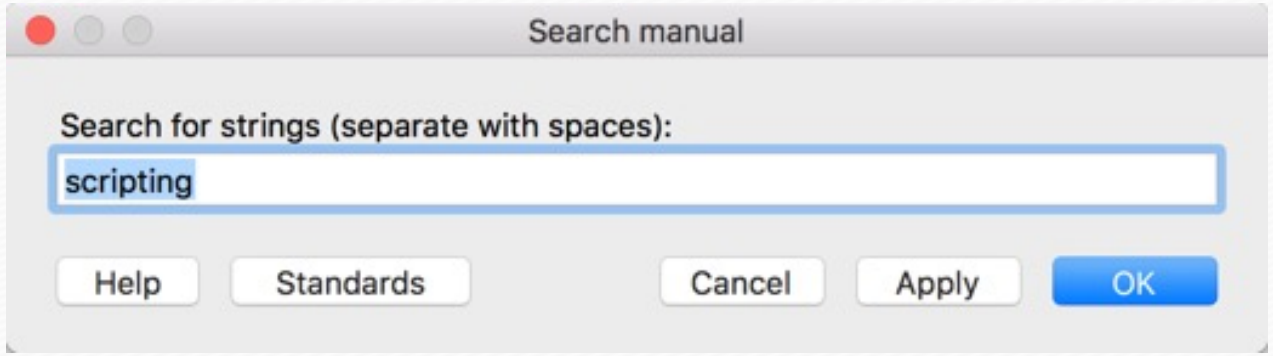
1. YearBirth
2. yEARrESIDING
3. studentid
4. symbol in IPA
5. f0_and_formants
6. f0AndFormants
7. 2ndFormant
8. name$

1. Incorrect 2. Correct 3. Correct 4. Incorrect 5. Correct 6. Correct 7. Incorrect 8. Correct.

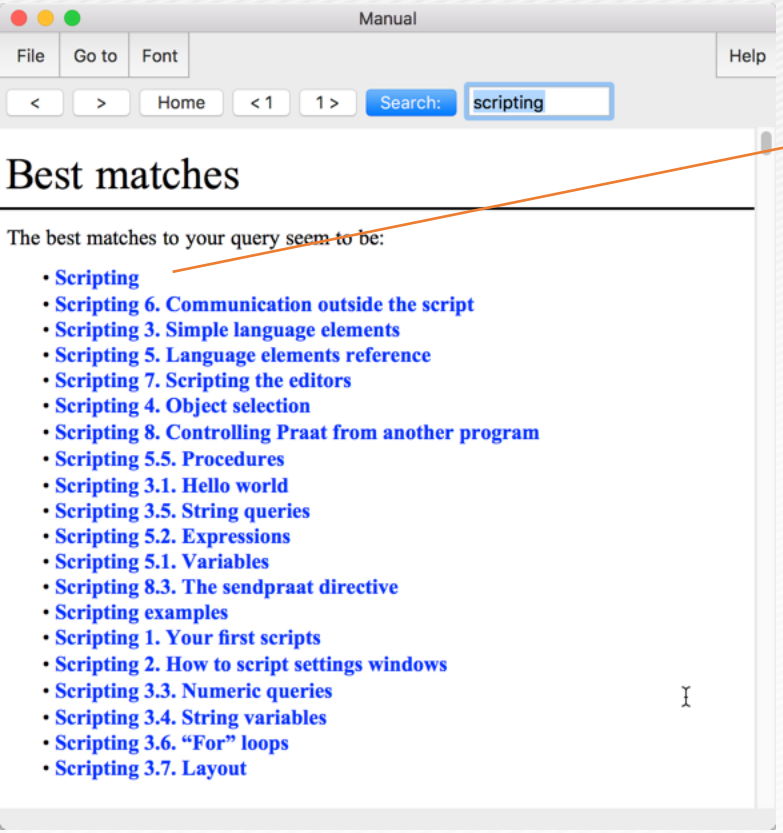
Help

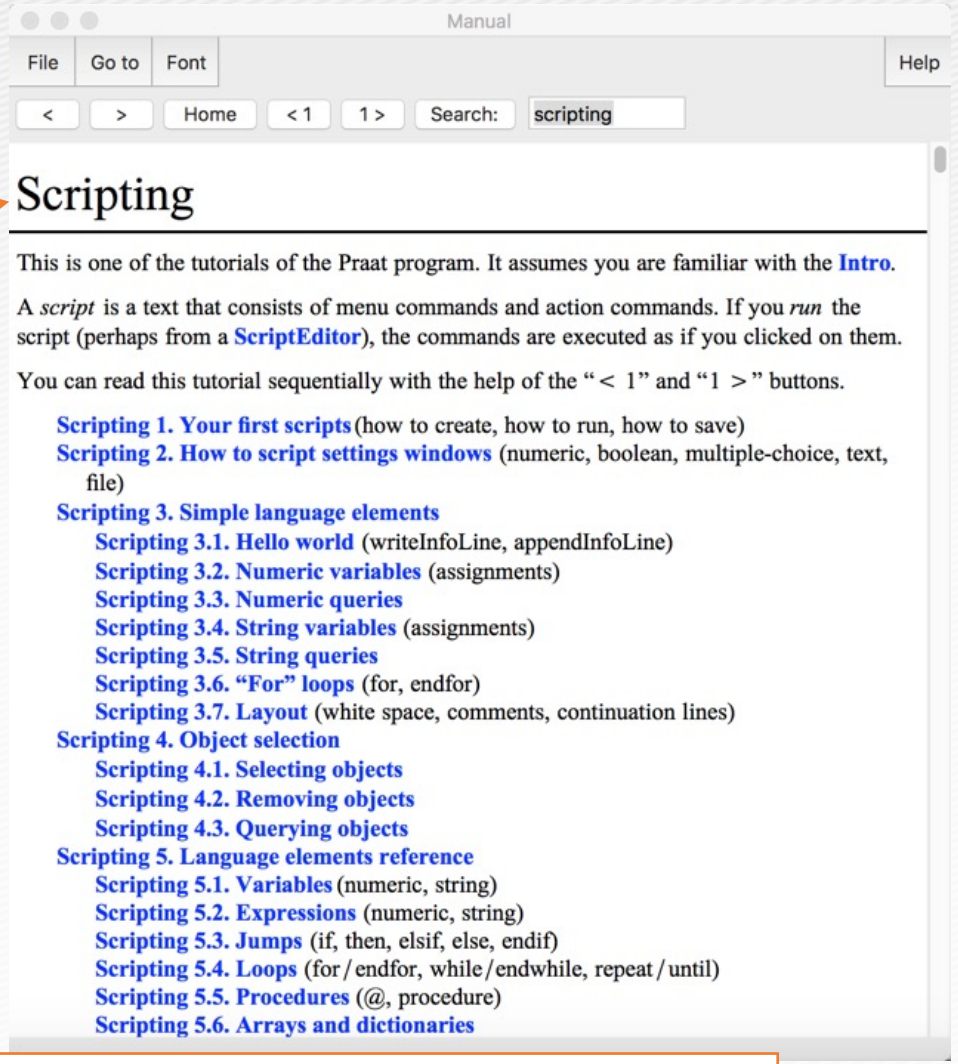

Praat Objects
New
Open
Save
Help
Objects:


Praat Intro
Object window help
Search Praat manual... ⌘M


Search manual
Search for strings (separate with spaces):
scripting
Help
Standards
Cancel
Apply
OK

Help for Praat scripting





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

04

Conditionals and Loops

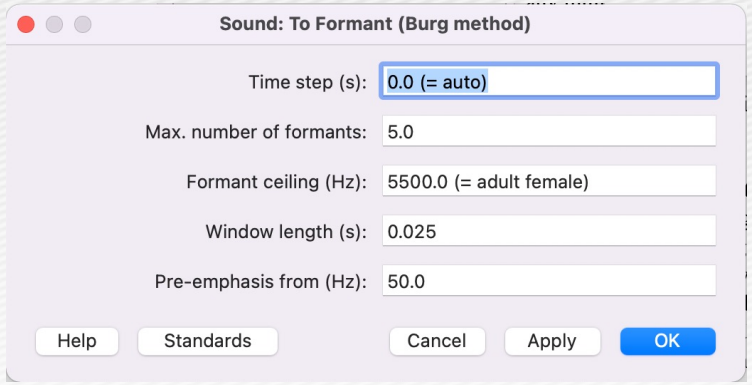
Boolean data type

- 0 or 1
- You use Boolen data type when you need logical value such as True or False

```
# This is a boolean variable with a value of false.
male = 0
```

conditionals

```
if boolExpression
     commands
elif boolExpression
     commands
else
     commands
endif
```



Formant ceiling (Hz)

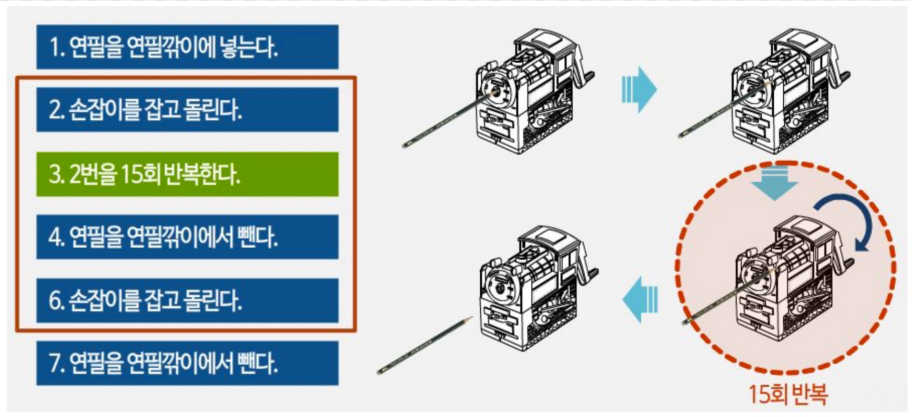
the maximum frequency of the formant search range, in hertz. It is crucial that you set this to a value suitable for your speaker. An average adult female speaker has a vocal tract length that requires an average ceiling of 5500 Hz (which is Praat's standard value), an average adult male speaker has a vocal tract length that requires an average ceiling of 5000 Hz, and a young child may have a vocal tract length that requires an average ceiling of 8000 Hz. These are just examples; there is large variation between speakers, and **Escudero, Boersma, Rauber & Bion (2009)** showed that the vocal tract length difference between a person's front vowels and their back vowels may be greater than the average vocal tract length difference between males and females. If you choose a too high ceiling, you may end up with too few formants in the low frequency region, e.g. analysing an [u] as having a single formant near 500 Hz whereas you want two formants at 300 and 600 Hz. You may like to experiment with this setting on steady vowels.

conditionals

```
clearinfo

male = 1
child = 0
formantCeiling = 5000

if male
      formantCeiling
elif child
      formantCeiling = 8000
else
      formantCeiling = 5500
endif

appendInfoLine: formantCeiling
```

Boolean expression

for Loops




```
clearinfo
appendInfoLine: "Insert a pencil in the sharpener."
repetition = 15
for i from 1 to repetition
     appendInfoLine: "Turn the handle ", i, " times."
endfor
appendInfoLine: "Remove the pencil from the sharpener."
```

Conditionals and Loops

```
clearinfo
appendInfoLine: "Insert a pencil in the sharpener."
repetition = 15
for i from 1 to repetition
     if i == 1
          appendInfoLine: "Turn the handle once."
     else
          appendInfoLine: "Turn the handle ", i, " times."
     endif
endfor
appendInfoLine: "Remove the pencil from the sharpener."
```

uksk_9_for_loop2.praat

05

Output Files

How to save the output to a file:

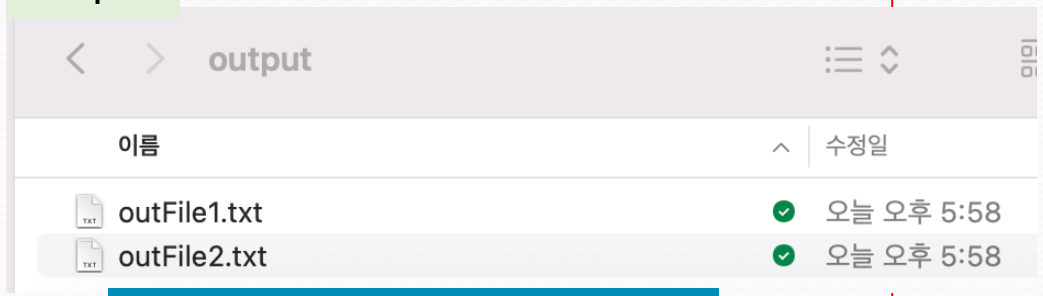
- writeFile
- WriteFileLine
- appendFile
- appendFileLine

Info window
- wirteInfo
- appendInfo
- writeInfoLine
- appendInfoLine

writeFile vs. appendFile

```
outDir$ = "../output/"

outFile1$ = outDir$ + "outFile1.txt"
outFile2$ = outDir$ + "outFile2.txt"

wMessage$ = "I'm writing, 'I love experimental phonetics'."
aMessage$ = "I'm appending, 'I love experimental phonetics'."

# let's write the message five times
for i from 1 to 5
      writeFile: outFile1$, wMessage$
endfor

# now we'll append 5 times
for j from 1 to 5
      appendFile: outFile2$, aMessage$
endfor
```

output



Open the output files to see the difference

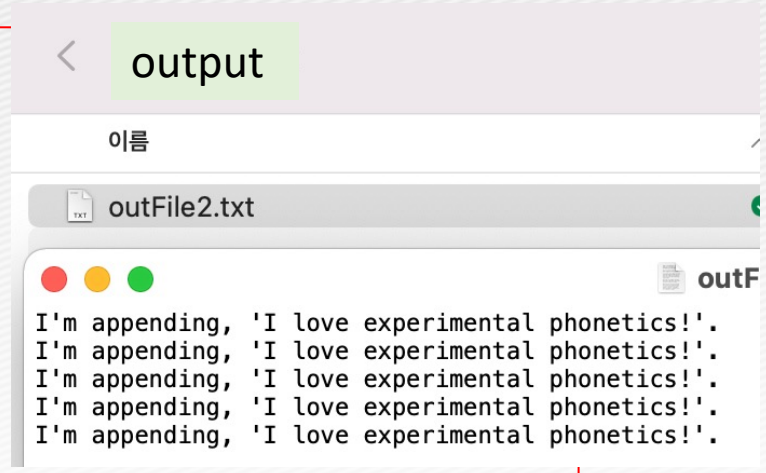
Checking if a file exists (1)

- Checking if the file exists and asking permission before overwriting it.
- Praat offers us the '`fileReadable`' command, which returns 1 if the file exists, and 0 if it doesn't.

```
outDir$ = "../output/"
outFile$ = outDir$ + "outFile1.txt"

if fileReadable: outFile$
     pauseScript: "File exists! Overwrite?"
endif
deleteFile: outFile$
```

Checking if a file exists (2)

```
outDir$ = "../output/"
askBeforeDelete = 1

outFile2$ = outDir$ + "outFile2.txt"

if askBeforeDelete and fileReadable: outFile2$
      pauseScript: "File exists. Overwrite?"
endif

deleteFile: outFile2$

aMessage$ = "I'm appending, 'I love experimental phonetics!'".
for i from 1 to 5
      appendFileLine: outFile2$, aMessage$
endfor
```



CSV (comma separated variable) file

```
outDir$ = "../output/"
# use commas for separator
sep$ = ","
outFile4$ = outDir$+"vowel.csv"
# Column names
header$ = “Name"+sep$+"Age"+sep$+"Vowel"+sep$+"F0"
        ...+sep$+"F1"+sep$+"F2"+newline$
writeFile: outFile4$, header$

# note that the numbers are actually strings!
yoon$ = "Yoon"+sep$+"20"+sep$+"i"+sep$+"120"+sep$
...+"240"+sep$+"2300"+newline$
kim$ = "Kim"+sep$+"22"+sep$+"i"+sep$
...+"114"+sep$+"230"+sep$+"2400"+newline$
lee$ = "Lee"+sep$+"19"+sep$+"e"+sep$
...+"119"+sep$+"340"+sep$+"2100"+newline$
appendFile: outFile4$, yoon$
appendFile: outFile4$, kim$
appendFile: outFile4$, lee$
```

output: vowel.csv

vowel

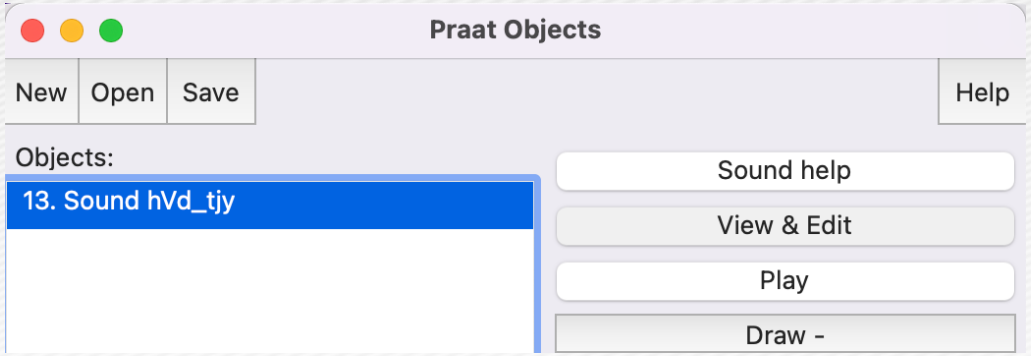
Name	Age	Vowel	F0	F1	F2
Yoon	20	i	120	240	2300
Kim	22	i	114	230	2400
Lee	19	e	119	340	2100

06

Object selection

Object selection

- Ways of selecting files in Praat:
  - by using the object's name
  - by using its number in the object window
  - By assigning object's ID to a variable

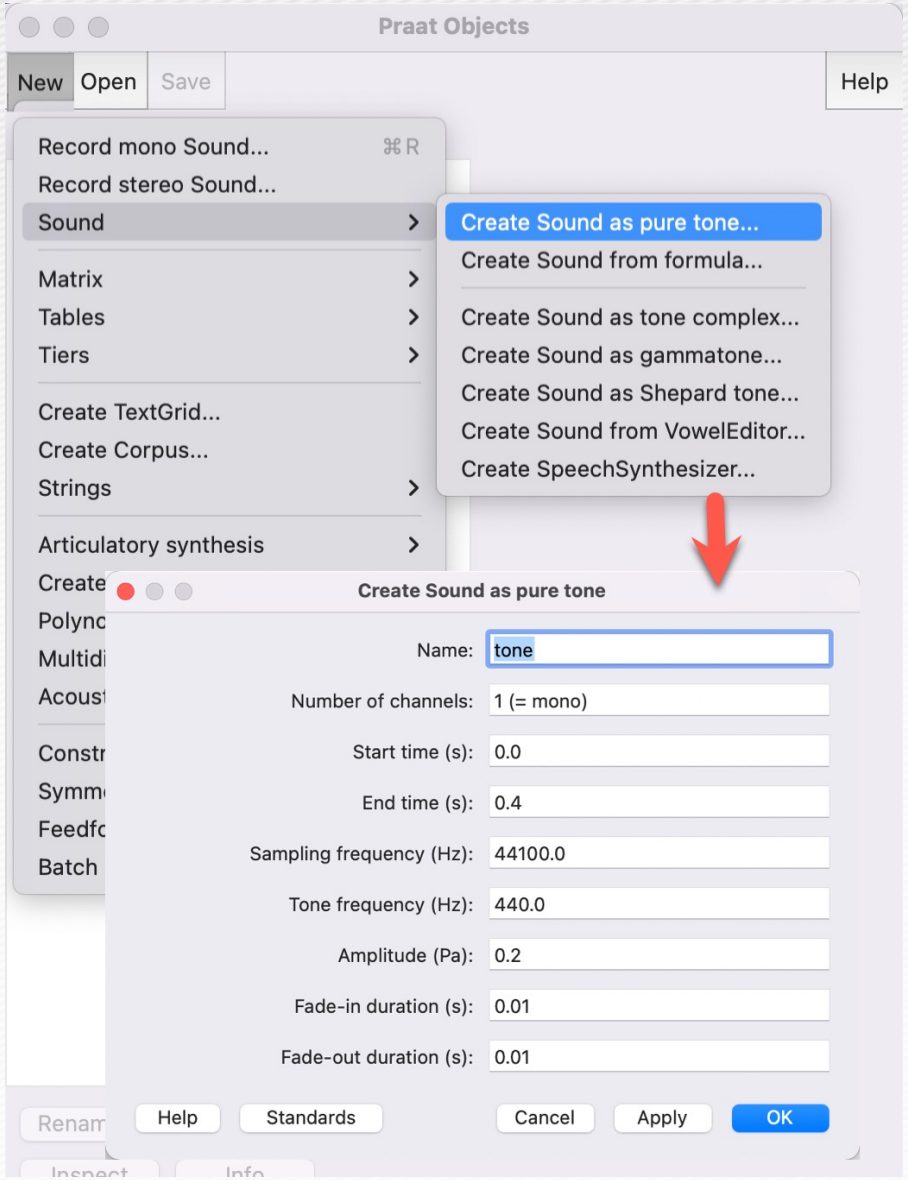


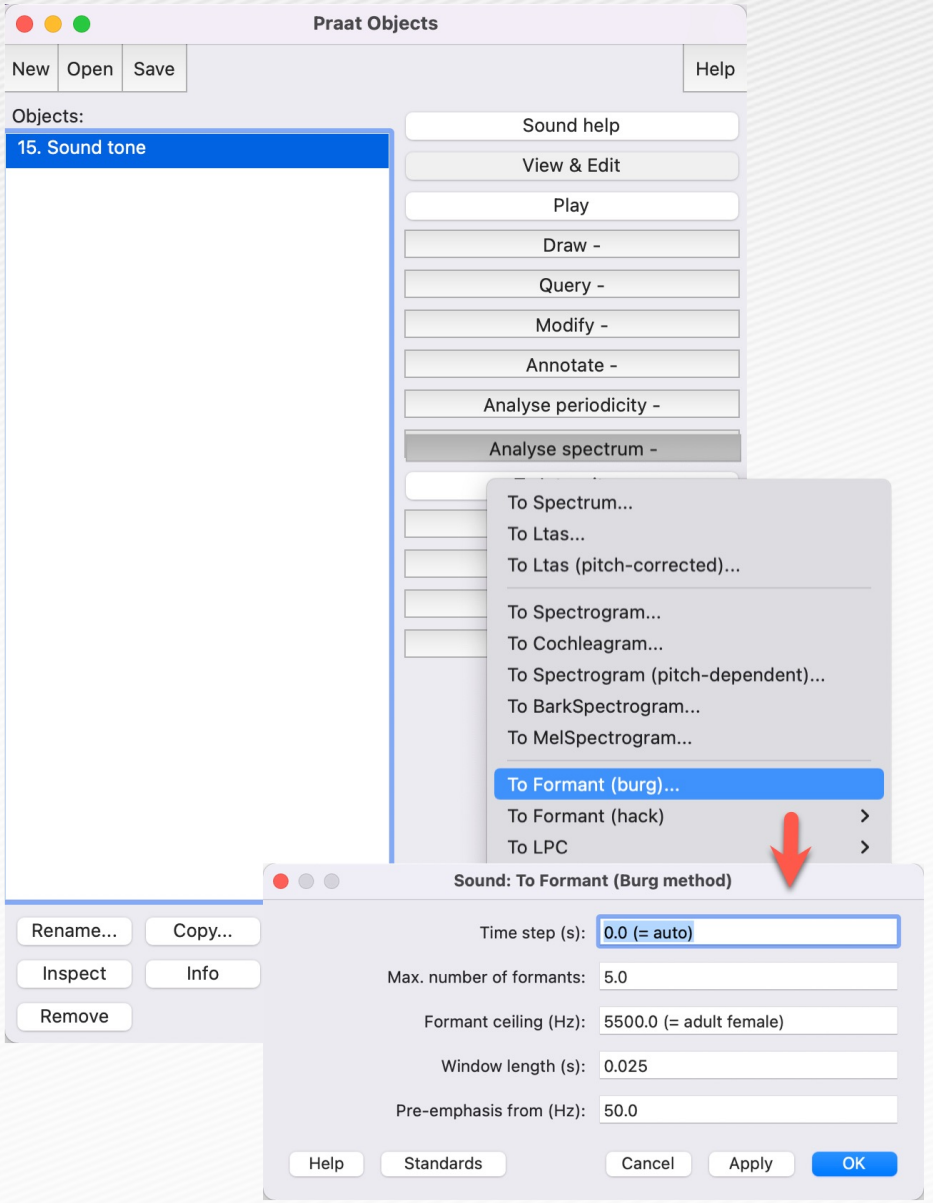

```
Read from file: "../data/hVd_tjy.wav"

#select by name
selectObject: "Sound hVd_tjy"

#select by object number
;selectObject: 13
```

unstable; not recommended

uksk_13_selectObject.praat

Extracting features from Praat commands





Extracting features from Praat commands

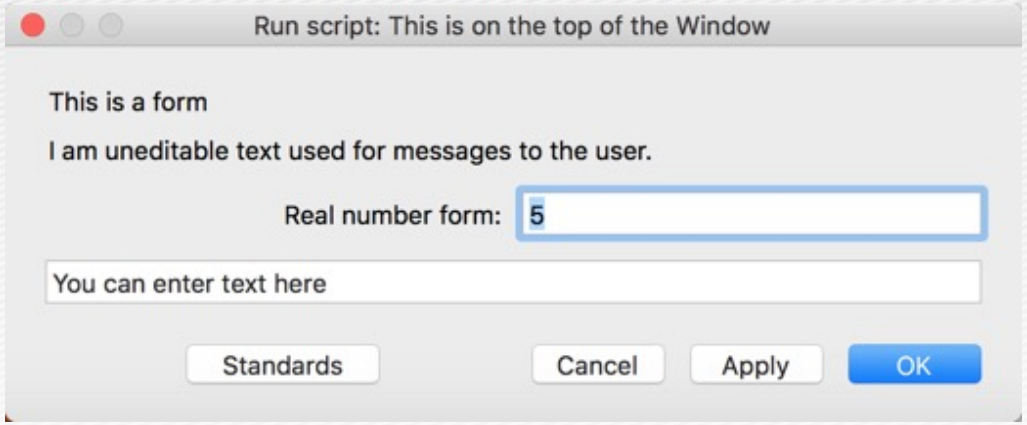
```
# create object from scratch
sndObj = Create Sound as pure tone:
   ..."tone", 1, 0, 0.4, 44100, 440, 0.2, 0.01, 0.01
formObj = To Formant (burg): 0, 5, 5500, 0.025, 50

selectObject: sndObj
plusObject: formObj

Remove
```

uksk_14_plusObject.praat

07

Directory Listing

pop-up windows



real
positive
integer
natural
boolean

comment
word
sentence
text
choice
button

```
form This is on the top of the Window
     comment This is a form
     comment I am uneditable text used for messages to the user.
     real Real_number_form 5
     text Enter_text_form You can enter text here
endform

clearinfo
appendInfoLine: "Form values: ",real_number_form,newline$,
                ... enter_text_form$
```

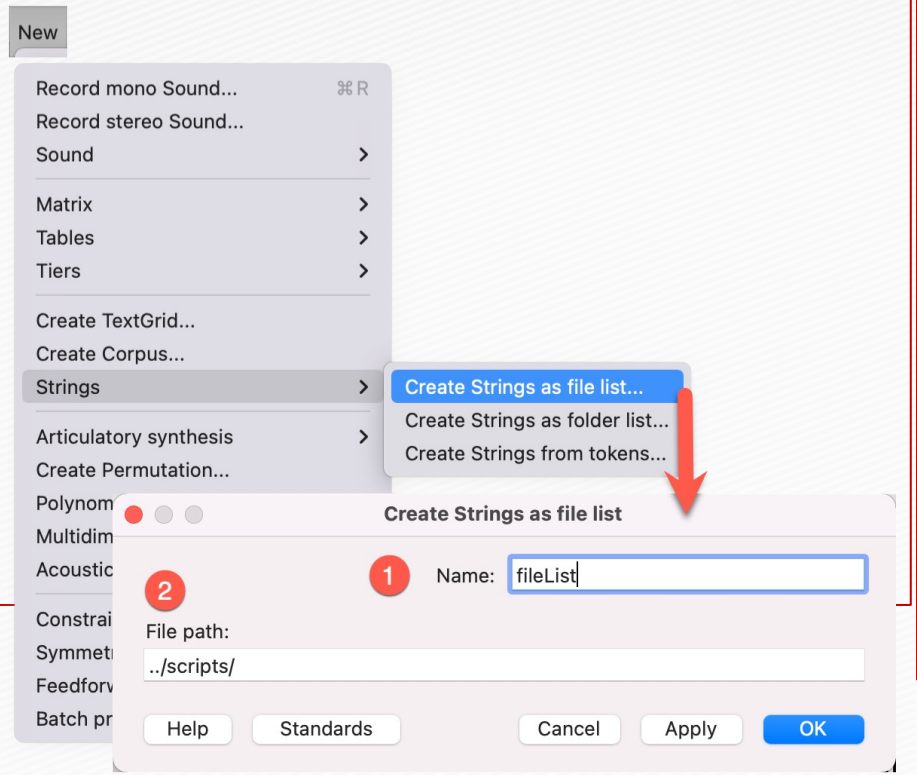
The pop-up window is for humans to read.

variables begin with lowercase

uksk_15_form.praat

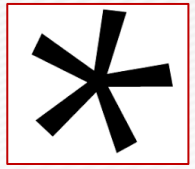
Looping through directories and files

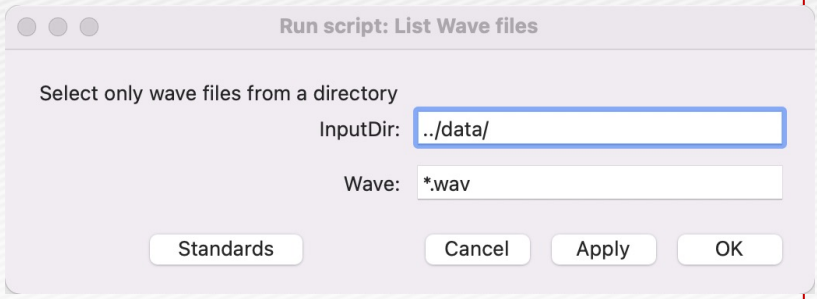
```
clearinfo
inputDir$ = "../scripts/"

fileList = Create Strings as file list: "fileList", inputDir$

selectObject: fileList

numFiles = Get number of strings
for fileNum from 1 to numFiles
     fileName$ = Get string: fileNum
     appendInfoLine: fileName$
endfor
```



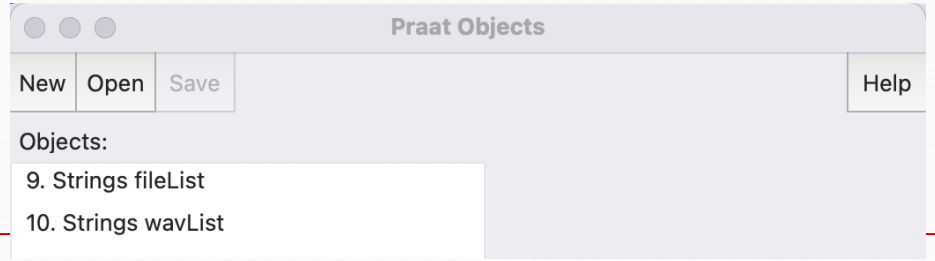
Picking up only wave files in a Directory

```
form List Wave files
comment Select only wave files from a directory
sentence InputDir ../data/
sentence Wave *.wav
endform

clearinfo
inputDirWavs$ = inputDir$ + wave$

wavList = Create Strings as file list: "wavList", inputDirWavs$
selectObject: wavList
numFiles = Get number of strings
for fileNum from 1 to numFiles
    fileName$ = Get string: fileNum
    appendInfoLine: fileName$
endfor
```





../data/*.wav



09

Extracting formants from vowels

Roadmap

01 Duration and Formant values from a single file

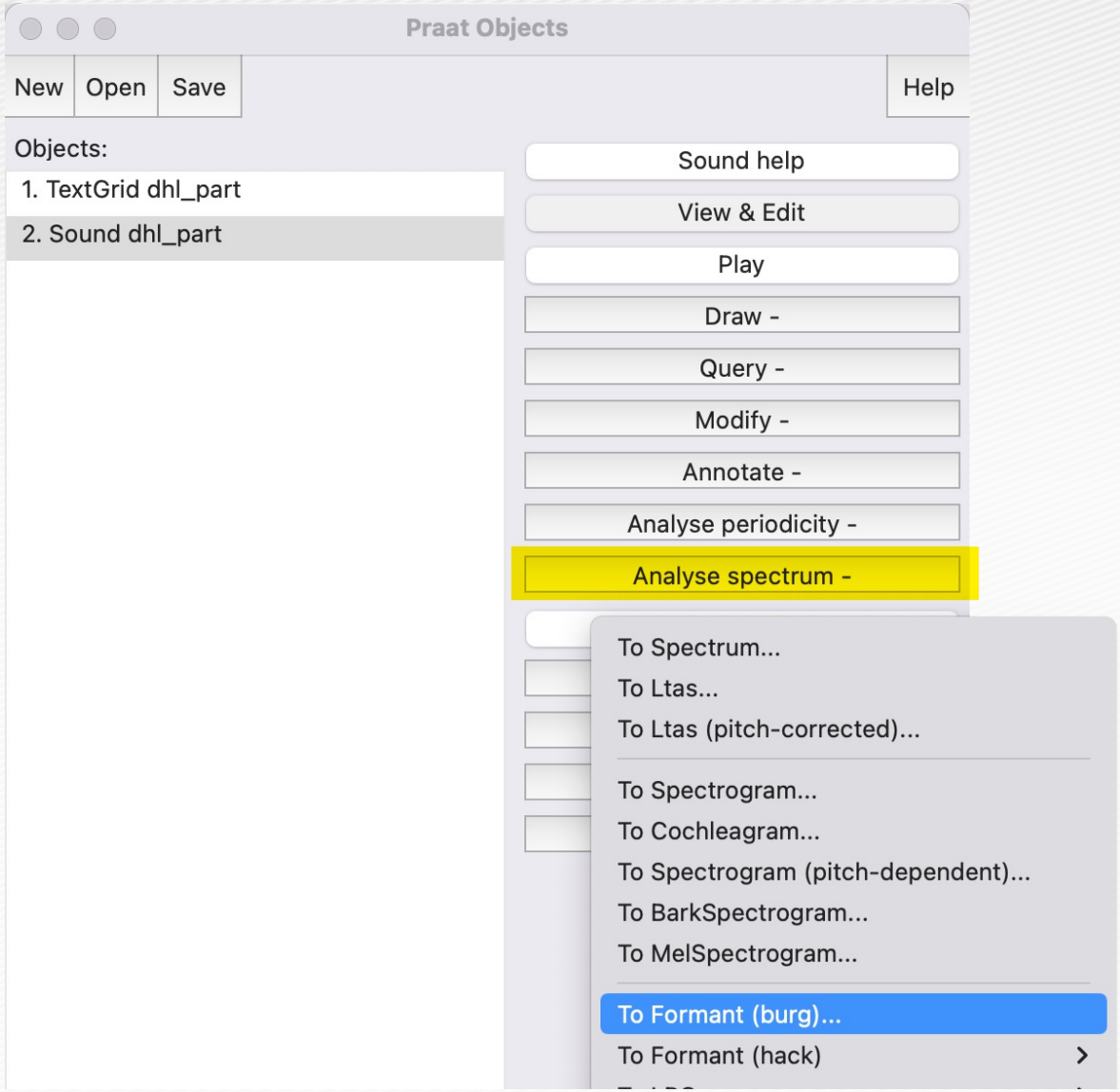
02 Pitch value from a single file

03 Formant values from files in a directory

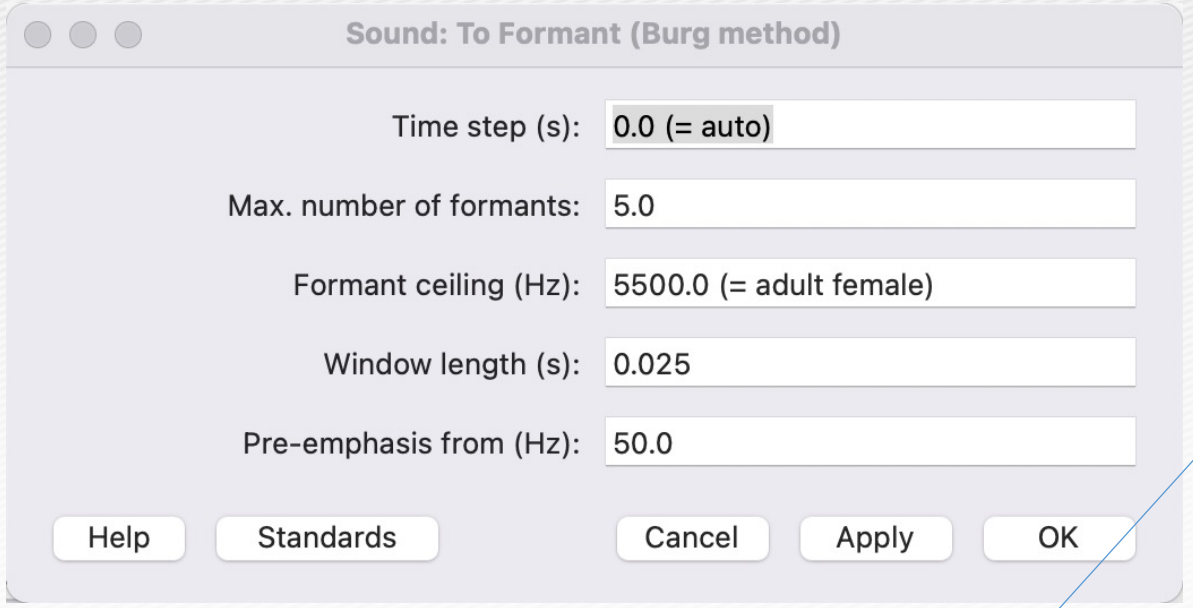
04 acoustic features from files in a directory

9-1 Formant values - manually

files: data/dhl_part.wav
data/dhl_part.TextGrid

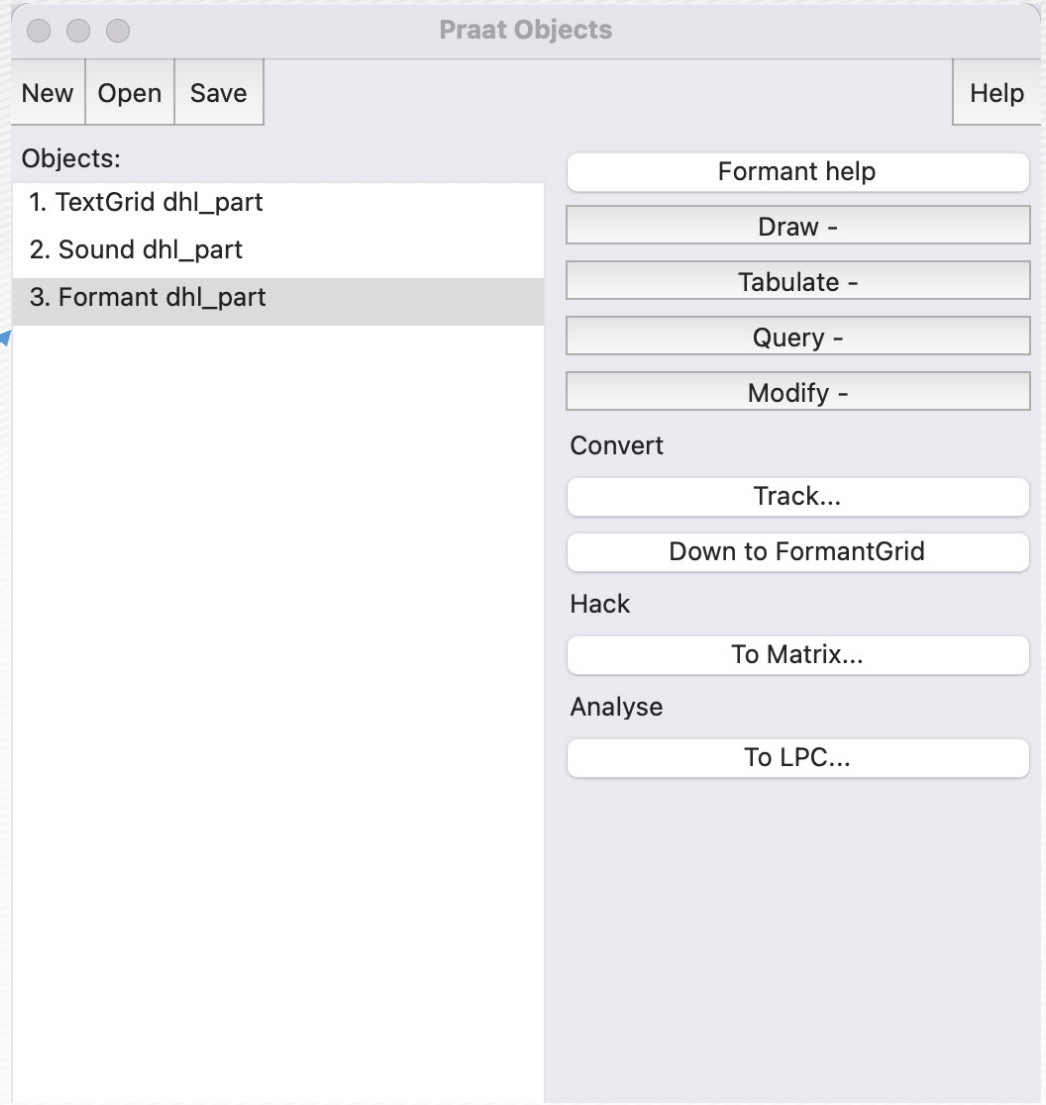


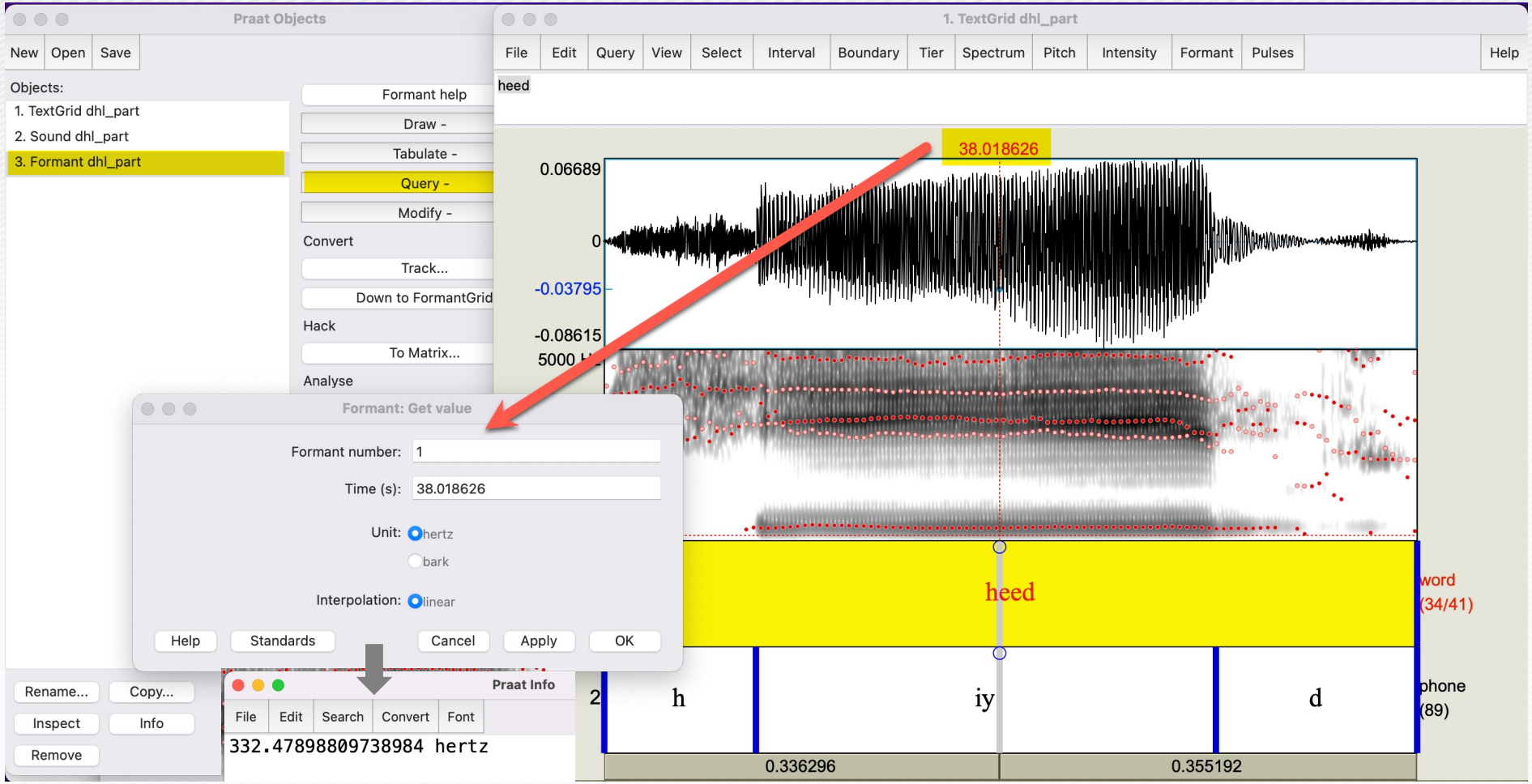
9-1 Formant values - manually



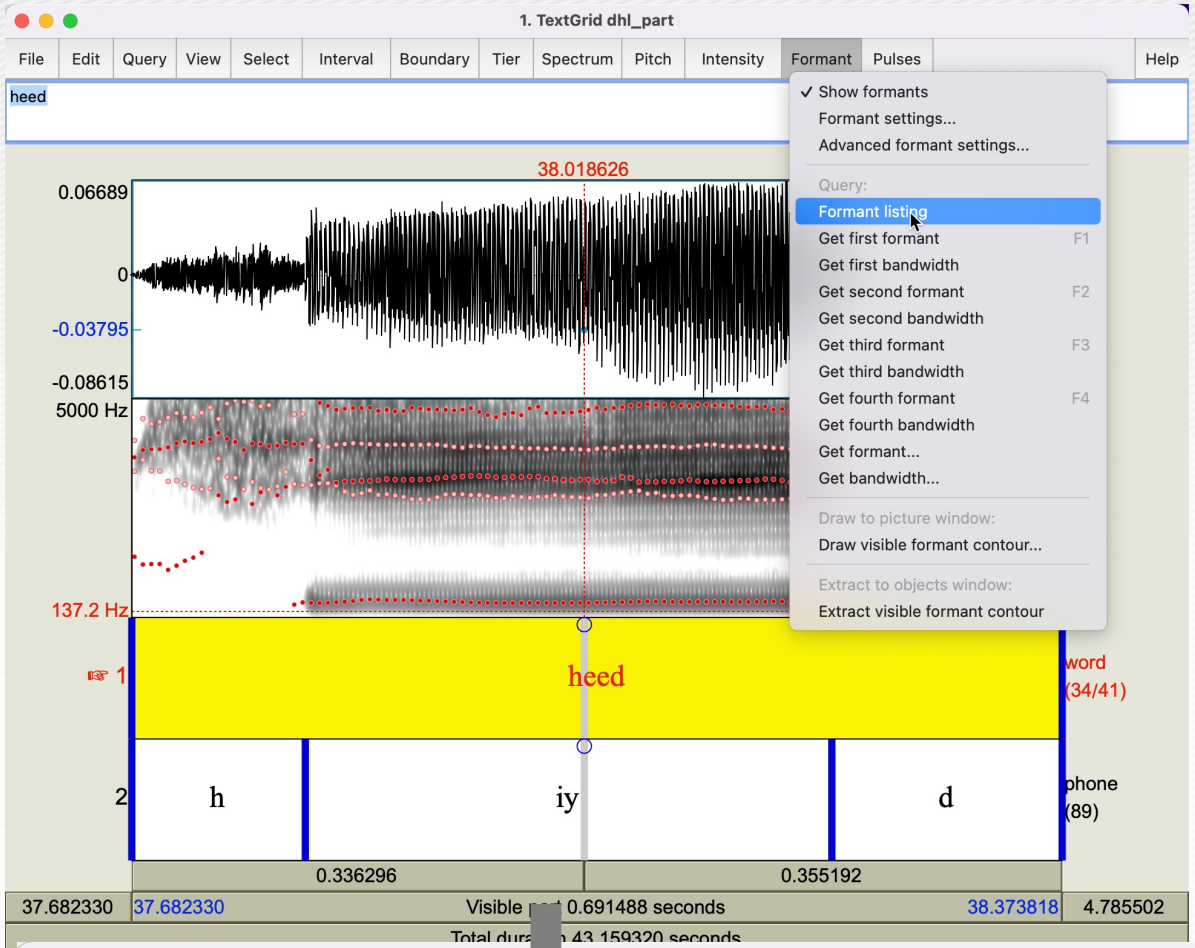
adult male speakers: 5000
adult female speakers: 5500
young child speakers: up to 8000

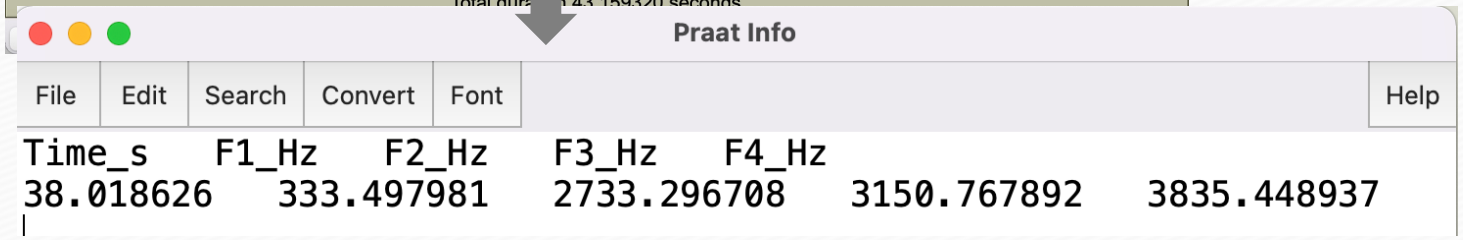
Formant Object is newly created.





1. Select both Sound Object and TextGrid Object together, and check the measurement point in time
2. Select Formant Object, and click Query > Get Value
3. Provide the Formant number and the measurement point (in time)





Put the cursor at the measurement point in the Edit window
Click Formant > Formant listing

Formant values - script

01 Opening sound and TextGrid files

02 Getting the mid point of vowels in the TextGrid

03 Formant Object from Sound files

04 Getting Formant values

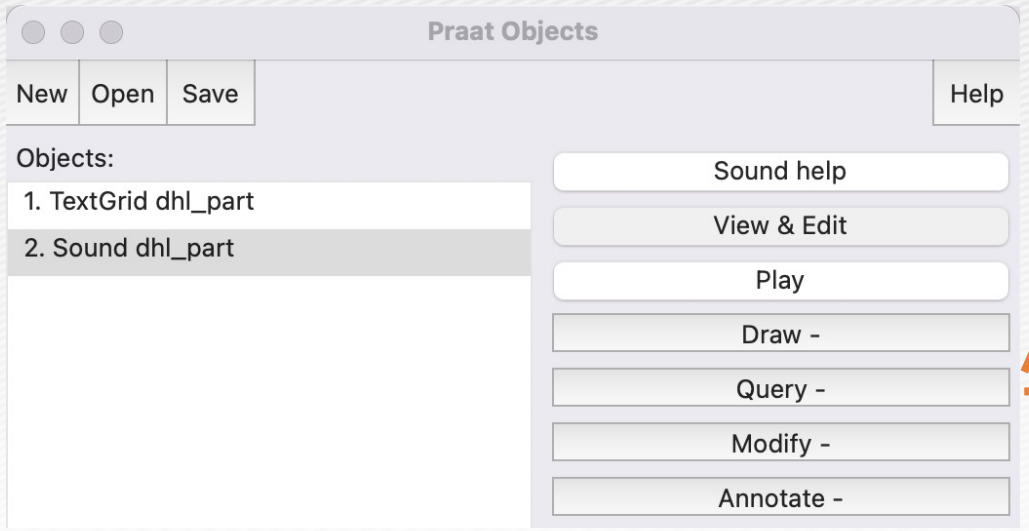
05 Saving the results on a log or txt file

step 1: reading files

```
## PART I

sound$ = Read from file: "../data/dhl_part.wav"
textgrid$ = Read from file: "../data/dhl_part.TextGrid"

# select Sound file

selectObject: "Sound dhl_part"

totalDuration = Get total duration
writeInfoLine: "Total duration: ", fixed$(totalDuration, 2), " seconds"

# Select TextGrid file

selectObject: "TextGrid dhl_part"

numberOfTiers = Get number of tiers
appendInfoLine: "Number of tiers is: ", numberOfTiers
```

up to two decimal places

Use Query!



TextGrid Object

Query -

- Query time domain
- Get number of tiers
- Get tier name...
- Is interval tier...
- Query interval tier
- Query point tier

Query interval tier:

- Get number of intervals...
- Get start time of interval...
- Get end time of interval...
- Get label of interval...
- Get interval at time...
- Get low interval at time...
- Get high interval at time...
- Get interval edge from time...
- Get interval boundary from time...
- Count intervals where...
- Get total duration of intervals where...

Sound Object

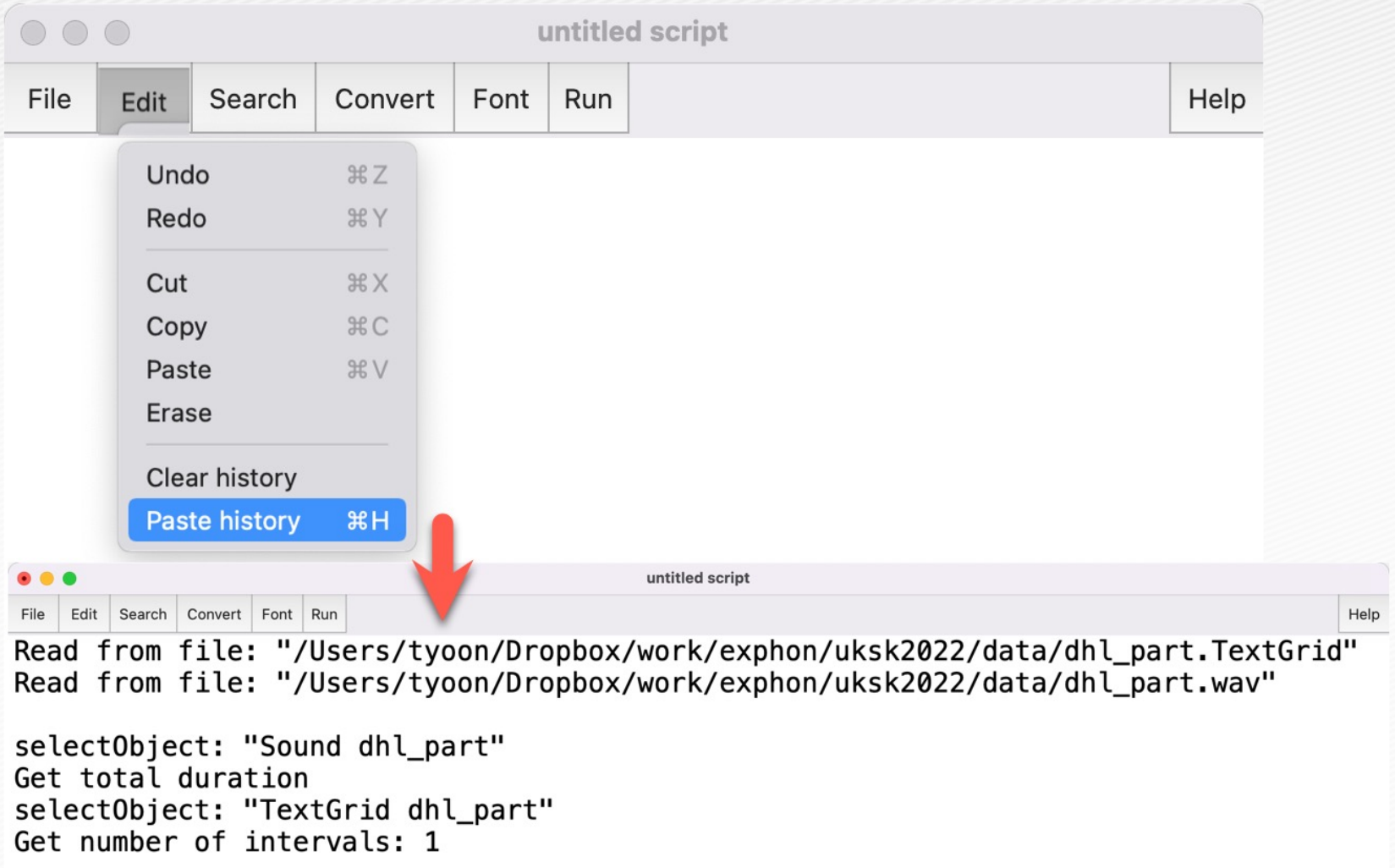
Query -

- 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)

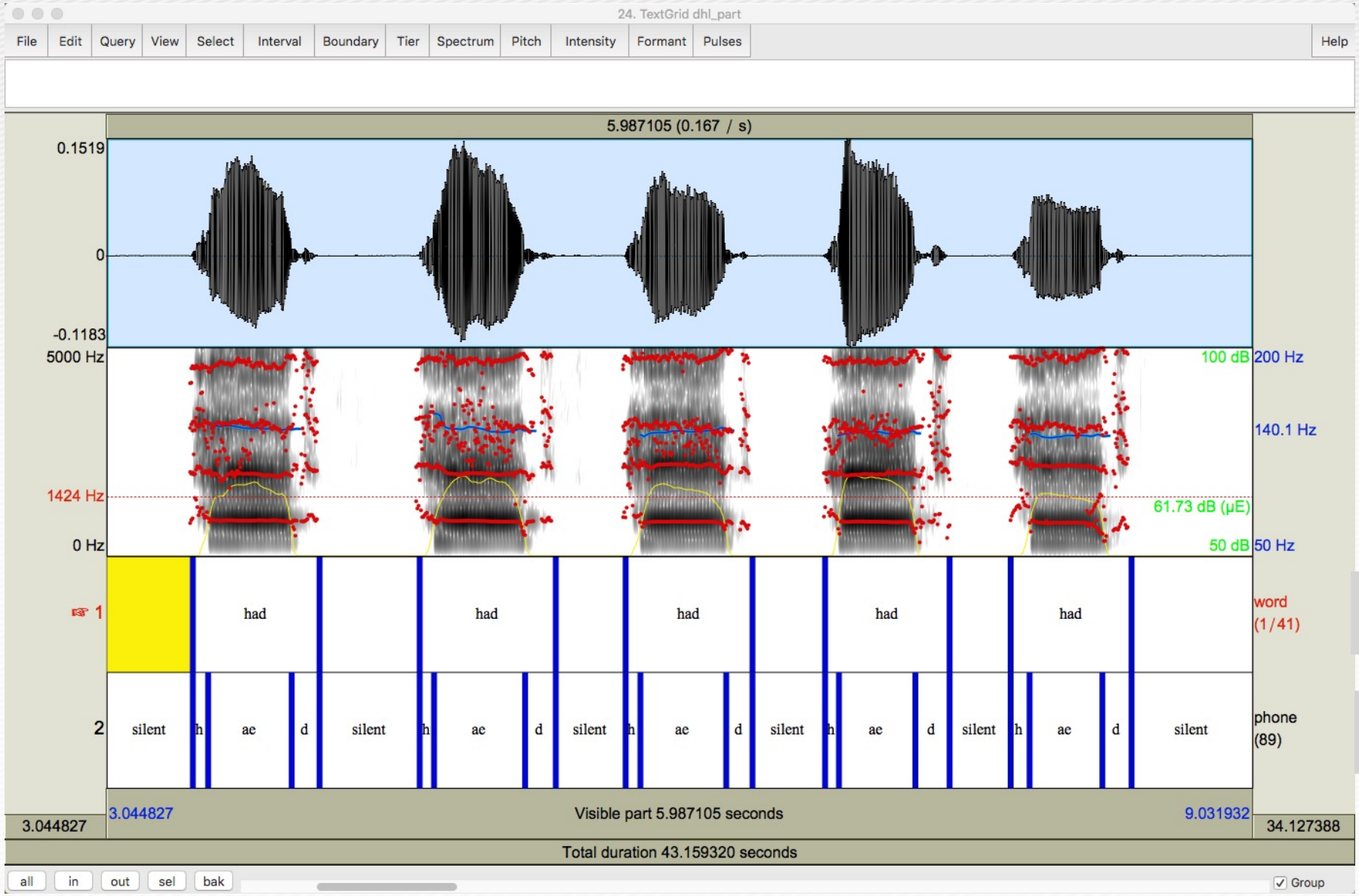
use `history` in the scripting window

Edit > Paste history in the Scripting Window

→ The history may show you how a function is used in the Praat script.



The structure of TextGrid



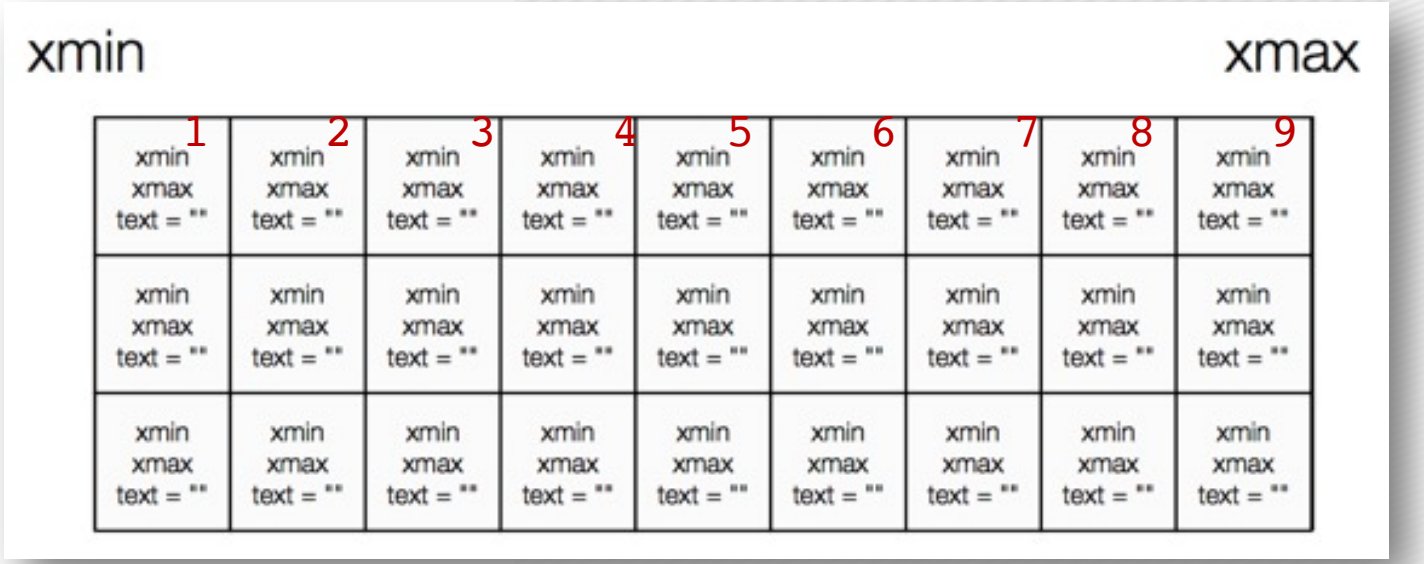
Tier 1 Word

Tier 2 Phone

The structure of TextGrid

dhl_part.TextGrid — 편집됨

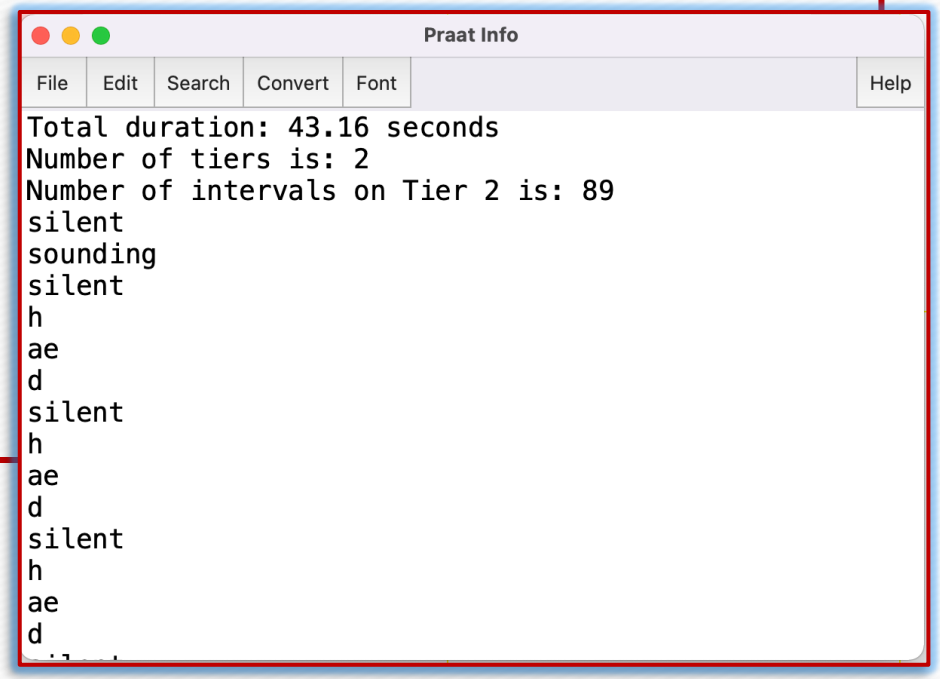
```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 43.15931972789116
tiers? <exists>
size = 2
item []:
    item [1]:
        class = "IntervalTier"
        name = "word"
        xmin = 0
        xmax = 43.15931972789116
        intervals: size = 41
        intervals [1]:
            xmin = 0
            xmax = 3.493036021831923
            text = ""
        intervals [2]:
            xmin = 3.493036021831923
            xmax = 4.155220950123126
            text = "had"
        intervals [3]:
            xmin = 4.155220950123126
            xmax = 4.678391977485527
            text = ""
        intervals [4]:
            xmin = 4.678391977485527
```




```
for index from 1 to num_intervals
    # DO SOMETHING
endfor
```

Step 2: Looping the TextGrid interval

```
## PART II
# print out phone tier information line by line

num_intervals = Get number of intervals: 2

appendInfoLine: "Number of intervals on Tier 2 is: ", num_intervals

# Loop the interval tiers
for i from 1 to num_intervals

    phone$ = Get label of interval: 2, i
    appendInfoLine: phone$

endfor
```



uksk_18_formant_step2.praat

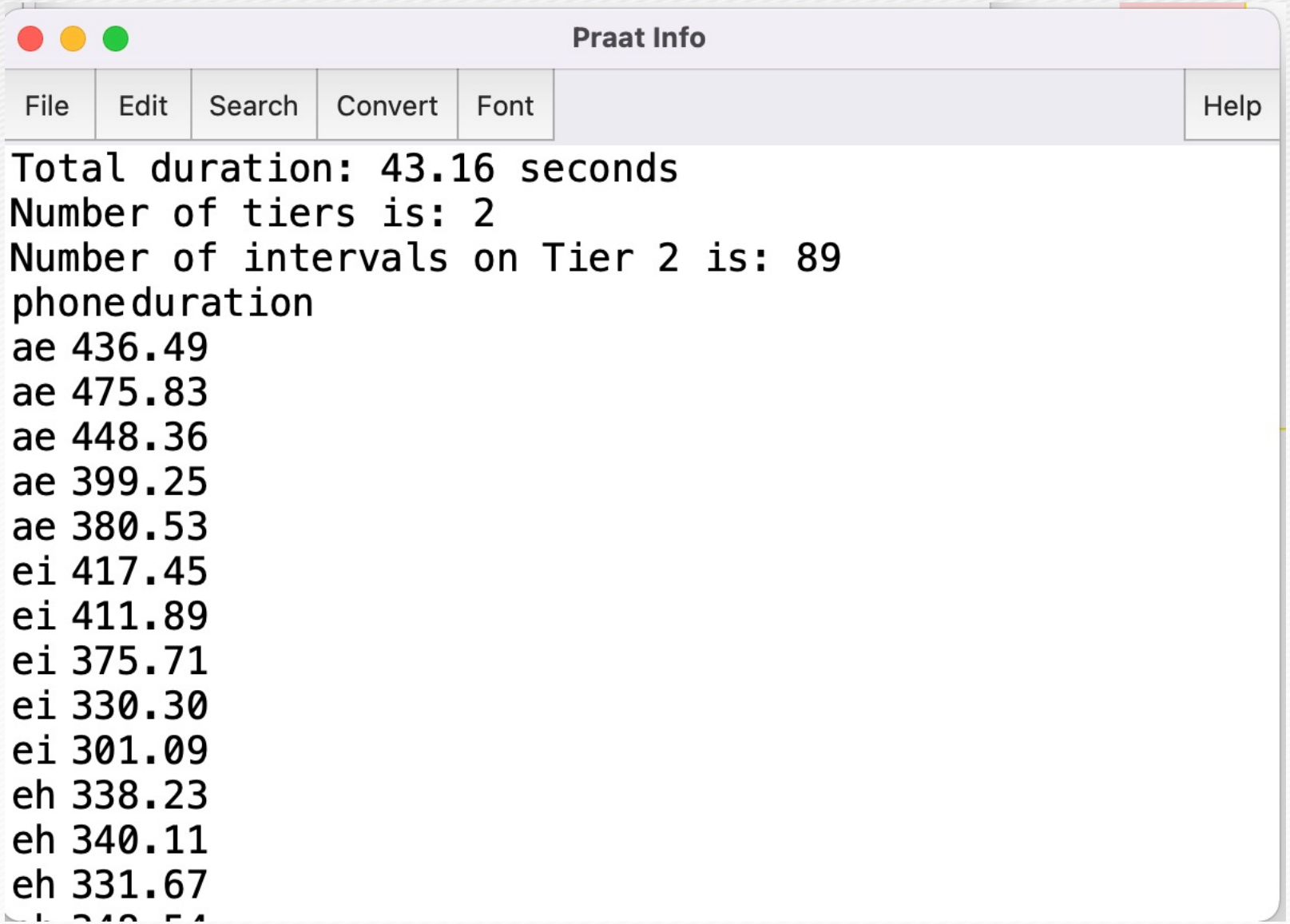
Step 2(modified): Vowel duration only

```
## PART II (modified)
# print out phone tier information line by line

num_intervals = Get number of intervals: 2
appendInfoLine: "Number of intervals on Tier 2 is: ", num_intervals

appendInfoLine: "phone",tab$,"duration"

# Loop the interval tiers
for i from 1 to num_intervals
    phone$ = Get label of interval: 2, i

    if phone$ == "ae" or phone$ == "ei"
       ... or phone$ == "eh" or phone$ == "iy"
        startTime = Get start point: 2, i
        endTime = Get end point: 2, i
        duration = (endTime - startTime)*1000
        appendInfoLine: phone$,tab$,fixed$(duration,2)
    endif
endfor
```

sec to milisec

uksk_18_formant_step2m.praat

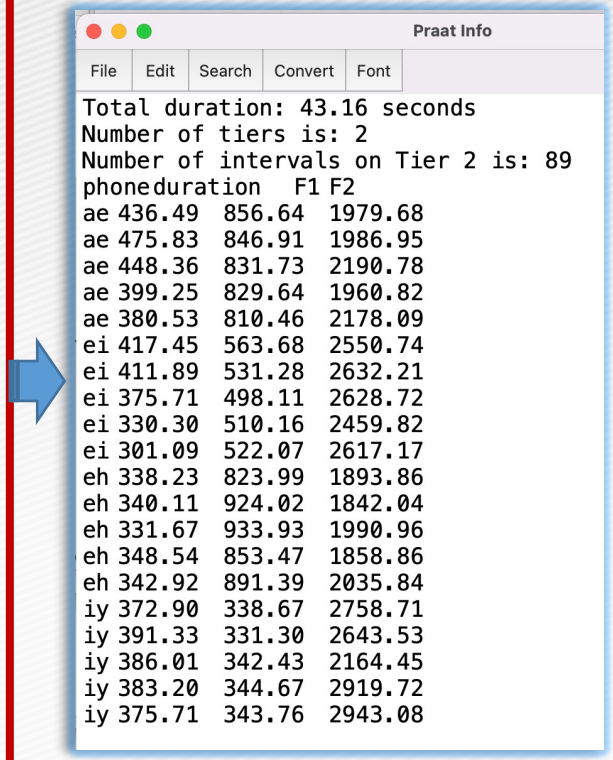
Step 2(modified): Vowel duration only - output



step 3: formants

```
## PART I
sound$ = Read from file: "../data/dhl_part.wav"
textgrid$ = Read from file: "../data/dhl_part.TextGrid"

# select Sound file
selectObject: "Sound dhl_part"

totalDuration = Get total duration
writeInfoLine: "Total duration: ", fixed$(totalDuration, 2), " seconds"

# Make formant Object
########################
selectObject: "Sound dhl_part"
To Formant (burg): 0, 5, 5500, 0.025, 50

# Select TextGrid file
selectObject: "TextGrid dhl_part"

numberOfTiers = Get number of tiers
appendInfoLine: "Number of tiers is: ", numberOfTiers
```

```
## PART II (modified)
# print out phone tier information line by line
num_intervals = Get number of intervals: 2
appendInfoLine: "Number of intervals on Tier 2 is: ", num_intervals

# Add names for F1 and F2
appendInfoLine: "phone",tab$,"duration",tab$,"F1",tab$,"F2"

# Loop the interval tiers
for i from 1 to num_intervals
    phone$ = Get label of interval: 2, i
    if phone$ == "ae" or phone$ == "ei"
       ... or phone$ == "eh" or phone$ == "iy"

        startTime = Get start point: 2, i
        endTime = Get end point: 2, i
        midTime = startTime + (endTime - startTime)/2
        duration = (endTime - startTime)*1000
        # Note: appendInfoLine --> appendInfo + tab$ at the end
        appendInfo: phone$,tab$,fixed$(duration,2),tab$
        ##########################
        # Measure F1 and F2
        ##########################
        selectObject: "Formant dhl_part"
        f1 = Get value at time: 1, midTime, "Hertz", "Linear"
        f2 = Get value at time: 2, midTime, "Hertz", "Linear"
        appendInfoLine: fixed$(f1,2),tab$,fixed$(f2,2)
        selectObject: "TextGrid dhl_part"
    endif
endfor
```



uksk_18_formant_step3.praat

step 4: save results to a file

```
## PART II (modified 2)
# print out phone tier information line by line
num_intervals = Get number of intervals: 2
appendInfoLine: "Number of intervals on Tier 2 is: ", num_intervals

## to the tab-limited txt file
output$ = "../output/result.txt"

# Add names for F1 and F2
writeInfoLine: "phone",tab$,"duration",tab$,"F1",tab$,"F2"

# to the output file
writeFileLine: output$, "phone",tab$,"duration",tab$,"F1",tab$,"F2"

# Loop the interval tiers

     #TO BE ADDED
endfor
```

- writeFile
- WriteFileLine
- appendFile
- appendFileLine

step 4: save results to a file

```
# Loop the interval tiers
for i from 1 to num_intervals
    phone$ = Get label of interval: 2, i
    if phone$ == "ae" or phone$ == "ei"
       ... or phone$ == "eh" or phone$ == "iy"
        startTime = Get start point: 2, i
        endTime = Get end point: 2, i
        midTime = startTime + (endTime - startTime)/2
        duration = (endTime - startTime)*1000
        # Note: appendInfoLine --> appendInfo + tab$ at the end
        appendInfo: phone$,tab$,fixed$(duration,2),tab$

        # output
        appendFile: output$, phone$,tab$,fixed$(duration,2),tab$

        ############################
        # PART III: Measure F1 and F2
        ############################
        #TO BE ADDED

    endif
endfor
```

step 4: save results to a file

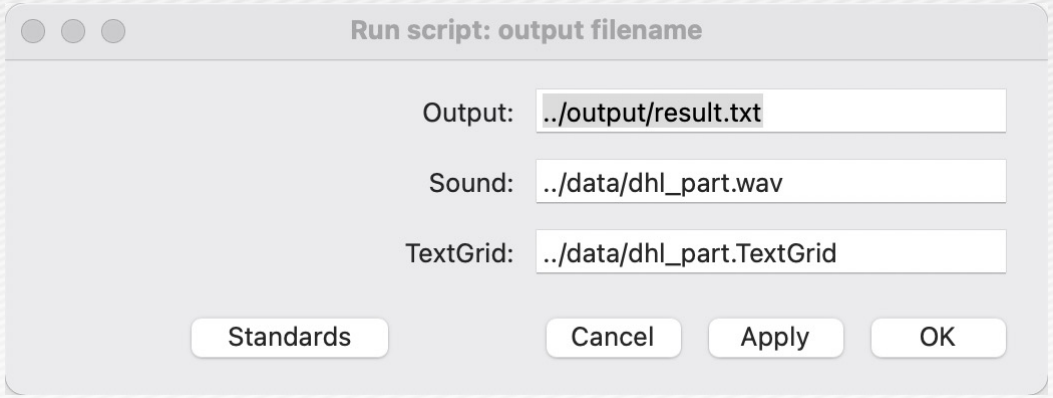
```
############################
# PART III: Measure F1 and F2
############################
selectObject: "Formant dhl_part"
f1 = Get value at time: 1, midTime, "Hertz", "Linear"
f2 = Get value at time: 2, midTime, "Hertz", "Linear"
appendInfoLine: fixed$(f1,2),tab$,fixed$(f2,2)

# save output to a file
appendFileLine: output$, fixed$(f1,2),tab$,fixed$(f2,2)


selectObject: "TextGrid dhl_part"
```

result.txt

phone	duration	F1	F2
ae	436.49	856.64	1979.68
ae	475.83	846.91	1986.95
ae	448.36	831.73	2190.78
ae	399.25	829.64	1960.82
ae	380.53	810.46	2178.09
ei	417.45	563.68	2550.74
ei	411.89	531.28	2632.21
ei	375.71	498.11	2628.72
ei	330.30	510.16	2459.82
ei	301.09	522.07	2617.17
eh	338.23	823.99	1893.86
eh	340.11	924.02	1842.04
eh	331.67	933.93	1990.96
eh	348.54	853.47	1858.86
eh	342.92	891.39	2035.84
iy	372.90	338.67	2758.71
iy	391.33	331.30	2643.53
iy	386.01	342.43	2164.45
iy	383.20	344.67	2919.72
iy	375.71	343.76	2943.08

Step 5: use form to save output files

```
form output filename
        sentence Output ../output/result.txt
        sentence Sound  ../data/dhl_part.wav
        sentence TextGrid ../data/dhl_part.TextGrid
endform

sound$ = Read from file: sound$
textgrid$ = Read from file: textGrid$
# select Sound file
selectObject: "Sound dhl_part"
totalDuration = Get total duration
writeInfoLine: "Total duration: ",
fixed$(totalDuration, 2), " seconds"
# Make formant Object
selectObject: "Sound dhl_part"
To Formant (burg): 0, 5, 5500, 0.025, 50
```



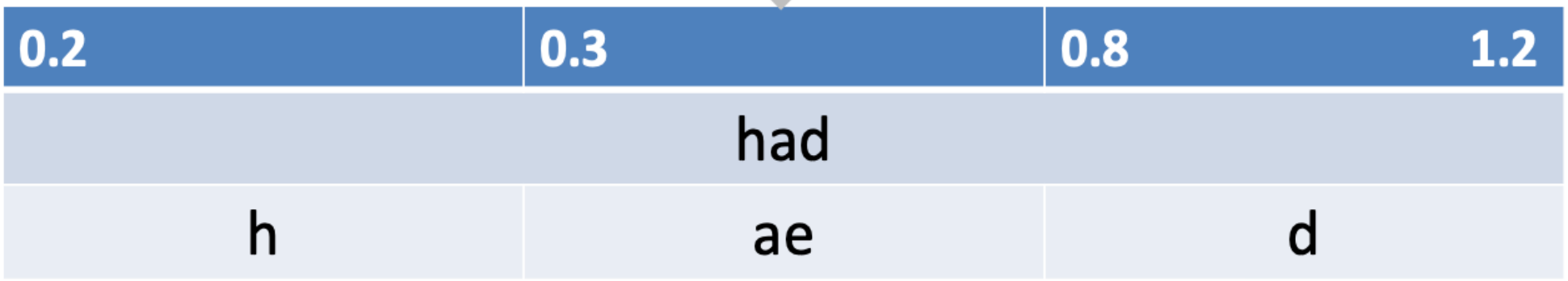

```
## PART II (modified 2)
…
## to the tab-limited txt file
output$ = "../output/result.txt"
```

ksss_3_formant_step5.praat

uksk_18_formant_step5.praat

Step 6: Adding word info in Tier 1

(2) what label would be on tier 1 when the time is at 0.55



(1) The middle point in time of the vowel [ae] is 0.55

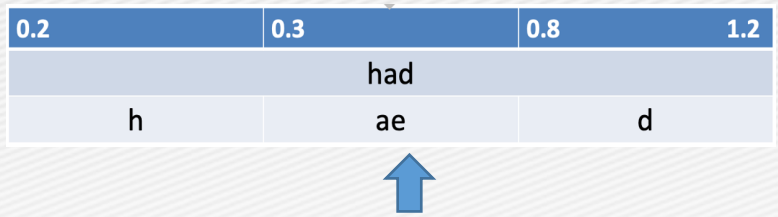
0.3 + (0.8-0.3)/2 = 0.55

The wanted output

```
ae      had     duration      F0
...
```

Step 6: Tier 1의 단어 정보 추가하기


```
##################
# IDENTIFY WORD
##################
word_index = Get interval at time: 1, midTime
word$ = Get label of interval: 1, word_index

word_start = Get starting point: 1, word_index
word_end = Get end point: 1, word_index

word_duration = (word_end - word_start)*1000

# Note: appendInfoLine --> appendInfo + tab$ at the end
appendInfo: phone$,tab$,word$,tab$,fixed$(duration,2),tab$
appendInfo: fixed$(word_duration,2),tab$

# output
appendFile: output$, phone$,tab$,word$,tab$
appendFile: output$, fixed$(duration,2),tab$
appendFile: output$, fixed$(word_duration,2),tab$
```



Step 6: Tier 1의 단어 정보 추가하기 - output

Output: result.txt

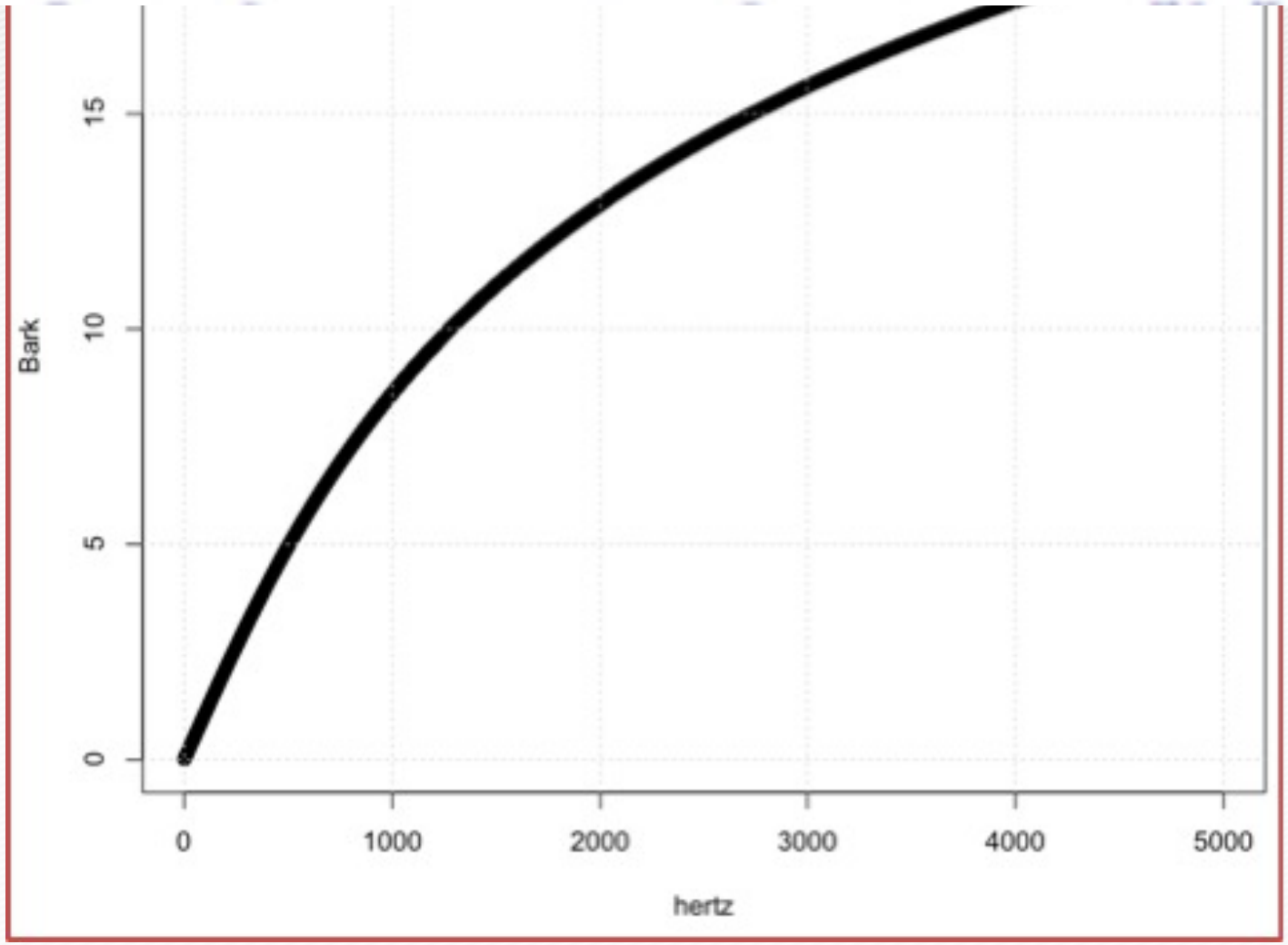
Praat Info

File Edit Search Convert Font Help

phone	word	duration	word_duration	F1	F2
ae	had	436.49	662.18	856.64	1979.68
ae	had	475.83	711.55	846.91	1986.95
ae	had	448.36	662.90	831.73	2190.78
ae	had	399.25	650.56	829.64	1960.82
ae	had	380.53	631.78	810.46	2178.09
ei	hayed	417.45	748.65	563.68	2550.74
ei	hayed	411.89	660.10	531.28	2632.21
ei	hayed	375.71	617.83	498.11	2628.72
ei	hayed	330.30	552.43	510.16	2459.82
ei	hayed	301.09	516.78	522.07	2617.17
eh	head	338.23	599.63	823.99	1893.86
eh	head	340.11	595.65	924.02	1842.04
eh	head	331.67	655.92	933.93	1990.96
eh	head	348.54	543.18	853.47	1858.86
eh	head	342.92	565.67	891.39	2035.84
iy	heed	372.90	628.09	338.67	2758.71
iy	heed	391.33	691.49	331.30	2643.53
iy	heed	386.01	711.53	342.43	2164.45
iy	heed	383.20	639.44	344.67	2919.72
iy	heed	375.71	618.37	343.76	2943.08

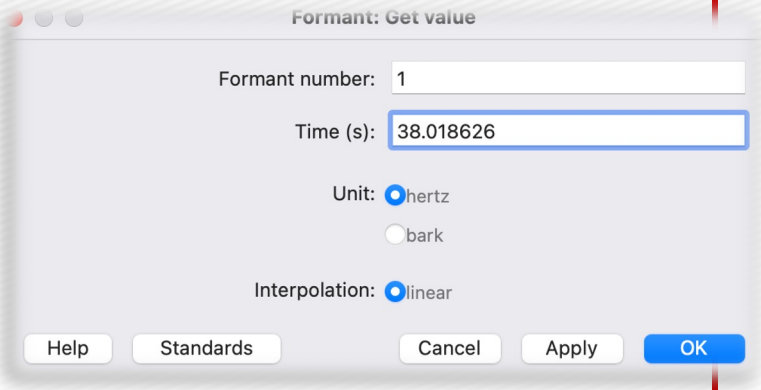
Step 7: Converting from Hz to Bark

```
Bark = 7*log(hertz/650+sqrt(1+(hertz/650)^2))
```



Step 7: Converting from Hz to Bark

```
        selectObject: "Formant dhl_part"
        f1 = Get value at time: 1, midTime, "Hertz", "Linear"
        f2 = Get value at time: 2, midTime, "Hertz", "Linear"

        ## CHANGE TO BARK
        ##########################################
        #  IN R
        #      hertz = seq(1, 5000, 1)
        #      Bark = 7*log(hertz/650+sqrt(1+(hertz/650)^2))
        #      plot(hertz, Bark, type="b")
        #      grid()
        ##########################################

        # http://www.fon.hum.uva.nl/praat/manual/Formulas_4__Mathematical_functions.html
        f1_Bark = hertzToBark(f1)
        f2_Bark = hertzToBark(f2)
        appendInfo: fixed$(f1,2),tab$,fixed$(f2,2),tab$
        appendInfoLine: fixed$(f1_Bark,2),tab$,fixed$(f2_Bark,2)

        #output
        appendFile: output$, fixed$(f1,2),tab$,fixed$(f2,2),tab$
        appendFileLine: output$, fixed$(f1_Bark,2),tab$,fixed$(f2_Bark,2)
        selectObject: "TextGrid dhl_part"
    endif
endfor
```



uksk_18_formant_step7.praat

Handling multiple files in a directory

```
form Measure phonetic features for segments in a textgrid
    comment Don't forget to end with / in sound_dir
    sentence sound_dir ../data/
    sentence output ../output/result.txt
endform
clearinfo
strings = Create Strings as file list: "list", sound_dir$ + "*.wav"
numberOfFiles = Get number of strings
writeInfoLine: "Number of wave files: ", numberOfFiles
for ifile to numberOfFiles
    selectObject: strings
    sound_file$ = Get string: ifile
    appendInfoLine: "Name of wave files: ", ifile,tab$,sound_file$
    basename$ = sound_file$ - ".wav"
    appendInfoLine: "Basename: ", basename$

    # Read wave and textgrid files
    Read from file: sound_dir$+"/"+sound_file$
    Read from file: sound_dir$+"/"+basename$+".TextGrid"
    # select TextGrid and cound the number of intervals in the tier
2
    selectObject: "TextGrid 'basename$'"
    num_intervals = Get number of intervals: 2
    appendInfoLine: "Number of intervals in Tier 2 is: ",
num_intervals
endfor
```

01

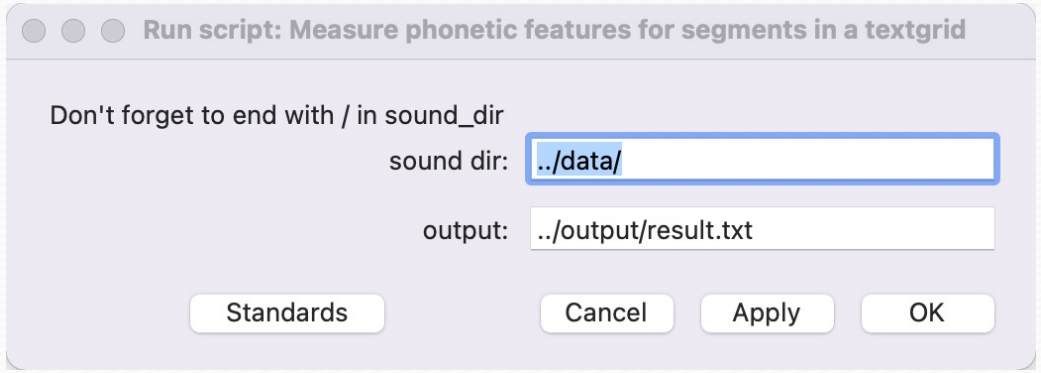
02

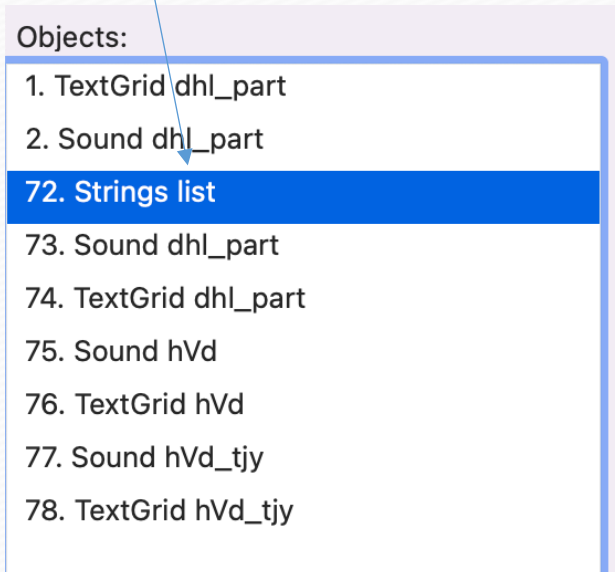
uksk_19_multiple_step1.praat

Handling multiple files in a directory

01

```
form Measure phonetic features for segments in a textgrid
    comment Don't forget to end with / in sound_dir
    sentence sound_dir ../data/
    sentence output ../output/result.txt
endform
clearinfo
strings = Create Strings as file list: "list", sound_dir$ + "*.wav"
numberOfFiles = Get number of strings
writeInfoLine: "Number of wave files: ", numberOfFiles
```





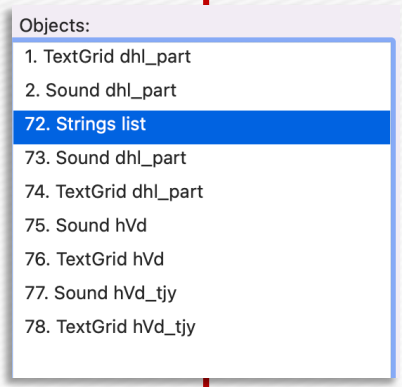
../data/*.wav

uksk_19_multiple_step1.praat

Handling multiple files in a directory

02

```
for ifile to numberOfFiles
    selectObject: strings
    sound_file$ = Get string: ifile
    appendInfoLine: "Name of wave files: ", ifile,tab$,sound_file$
    basename$ = sound_file$ - ".wav"
    appendInfoLine: "Basename: ", basename$

    # Read wave and textgrid files
    Read from file: sound_dir$+sound_file$
    Read from file: sound_dir$+basename$+".TextGrid"

    # select TextGrid and cound the number of intervals in the tier 2
    selectObject: "TextGrid 'basename$'"
    num_intervals = Get number of intervals: 2
    appendInfoLine: "Number of intervals in Tier 2 is: ", num_intervals
endfor
```



uksk_19_multiple_step1.praat

Handling multiple files in a directory

```
form Measure phonetic features for segments in a textgrid
   sentence sound_dir ../data/
   sentence output ../output/result.txt
endform

# Add names for F1 and F2
writeInfoLine: "phone",tab$,"word",tab$,"duration",tab$,
  ... "word_duration",tab$,"F1",tab$,"F2",tab$,"F1_Bark",tab$,"F2_Bark"

# to the output file
writeFileLine: output$, "phone",tab$,"word",tab$,"duration",tab$,
  ... "word_duration",tab$,"F1",tab$,"F2",tab$,"F1_Bark",tab$,"F2_Bark"

strings = Create Strings as file list: "list", sound_dir$ +"/"+ "*.wav"
numberOfFiles = Get number of strings
;writeInfoLine: "Number of wave files: ", numberOfFiles

for ifile to numberOfFiles
   selectObject: strings
   sound_file$ = Get string: ifile
   ;appendInfoLine: "Name of wave files: ", ifile,tab$,sound_file$
   basename$ = sound_file$ - ".wav"
   ;appendInfoLine: "Basename: ", basename$

   # Read wave and textgrid files
   Read from file: sound_dir$+"/"+sound_file$
   Read from file: sound_dir$+"/"+basename$+".TextGrid"

   # Formant Object
   selectObject: "Sound 'basename$'"
   To Formant (burg): 0, 5, 5500, 0.025, 50

   # select TextGrid and cound the number of intervals in the tier 2
   selectObject: "TextGrid 'basename$'"
   num_intervals = Get number of intervals: 2
   ;appendInfoLine: "Number of intervals in Tier 2 is: ", num_intervals

endfor
```

```
# Loop the interval tiers
for i from 1 to num_intervals
   phone$ = Get label of interval: 2, i
   if phone$ == "ae" or phone$ == "ei"
      ... or phone$ == "eh" or phone$ == "iy"
      startTime = Get start point: 2, i
      endTime = Get end point: 2, i
      midTime = startTime + (endTime - startTime)/2
      duration = (endTime - startTime)*1000

      #################
      # IDENTIFY WORD
      #################
      word_index = Get interval at time: 1, midTime
      word$ = Get label of interval: 1, word_index
      word_start = Get starting point: 1, word_index
      word_end = Get end point: 1, word_index
      word_duration = (word_end - word_start)*1000

      appendInfo: phone$,tab$,word$,tab$,fixed$(duration,2),tab$
      appendInfo: fixed$(word_duration,2),tab$

      # output
      appendFile: output$, phone$,tab$,word$,tab$
      appendFile: output$, fixed$(duration,2),tab$
      appendFile: output$, fixed$(word_duration,2),tab$

      ################################
      # Measure F1 and F2
      ################################
      selectObject: "Formant 'basename$'"
      f1 = Get value at time: 1, midTime, "Hertz", "Linear"
      f2 = Get value at time: 2, midTime, "Hertz", "Linear"

      f1_Bark = hertzToBark(f1)
      f2_Bark = hertzToBark(f2)

      appendInfo: fixed$(f1,2),tab$,fixed$(f2,2),tab$
      appendInfoLine: fixed$(f1_Bark,2),tab$,fixed$(f2_Bark,2)

      #output
      appendFile: output$, fixed$(f1,2),tab$,fixed$(f2,2),tab$
      appendFileLine: output$, fixed$(f1_Bark,2),tab$,fixed$(f2_Bark,2)

      selectObject: "TextGrid 'basename$'"

   endif
endfor
```

Praat Info

File Edit Search Convert Font Help

phone	word	duration	word_duration	F1	F2	F1_Bark	F2_Bark
ae	had	436.49	662.18	856.64	1979.68	7.63	12.83
ae	had	475.83	711.55	846.91	1986.95	7.56	12.85
ae	had	448.36	662.90	831.73	2190.78	7.46	13.51
ae	had	399.25	650.56	829.64	1960.82	7.45	12.77
ae	had	380.53	631.78	810.46	2178.09	7.32	13.47
ei	hayed	417.45	748.65	563.68	2550.74	5.49	14.53
ei	hayed	411.89	660.10	531.28	2632.21	5.22	14.75
ei	hayed	375.71	617.83	498.11	2628.72	4.94	14.74
ei	hayed	330.30	552.43	510.16	2459.82	5.05	14.29
ei	hayed	301.09	516.78	522.07	2617.17	5.15	14.71
eh	head	338.23	599.63	823.99	1893.86	7.41	12.54
eh	head	340.11	595.65	924.02	1842.04	8.05	12.35
eh	head	331.67	655.92	933.93	1990.96	8.11	12.87
eh	head	348.54	543.18	853.47	1858.86	7.60	12.41
eh	head	342.92	565.67	891.39	2035.84	7.85	13.02
iy	heed	372.90	628.09	338.67	2758.71	3.50	15.07
iy	heed	391.33	691.49	331.30	2643.53	3.43	14.78
iy	heed	386.01	711.53	342.43	2164.45	3.54	13.43
iy	heed	383.20	639.44	344.67	2919.72	3.56	15.45
iy	heed	375.71	618.37	343.76	2943.08	3.55	15.51

uksk_19_multiple_step2.praat

10

Adding F0 & Intensity…

```
form Measure phonetic features for segments in a textgrid
comment Don't forget / at the end of sound_dir
   sentence sound_dir ../data/
   sentence output ../output/result.txt
endform

clearinfo
# Add names for F1 and F2
writeInfoLine: "Basename",tab$,"phone",tab$,"word",tab$,"duration",tab$,
   ...
"word_duration",tab$,"F0",tab$,"F0st",tab$,"F1",tab$,"F2",tab$,"F1_Bark",ta
b$,"F2_Bark",tab$,"Intensity"

# to the output file
writeFileLine: output$,
"Basename",tab$,"phone",tab$,"word",tab$,"duration",tab$,
   ...
"word_duration",tab$,"F0",tab$,"F0st",tab$,"F1",tab$,"F2",tab$,"F1_Bark",ta
b$,"F2_Bark",tab$,"Intensity"

strings = Create Strings as file list: "list", sound_dir$ +"/"+ "*.wav"
numberOfFiles = Get number of strings

;writeInfoLine: "Number of wave files: ", numberOfFiles

for ifile to numberOfFiles

endfor
```

```
selectObject: strings
 sound_file$ = Get string: ifile
 ;appendInfoLine: "Name of wave files: ", ifile,tab$,sound_file$
 basename$ = sound_file$ - ".wav"
 ;appendInfoLine: "Basename: ", basename$


 # Read wave and textgrid files
 Read from file: sound_dir$+"/"+sound_file$
 Read from file: sound_dir$+"/"+basename$+".TextGrid"

 # Formant Object
 selectObject: "Sound 'basename$'"
 To Formant (burg): 0, 5, 5500, 0.025, 50

 # Pitch Object
 selectObject: "Sound 'basename$'"
 To Pitch: 0, 75, 600

 # Intensity Object  #
 selectObject: "Sound 'basename$'"
 To Intensity: 100, 0.0, 1

 # select TextGrid and cound the number of intervals in the tier 2
 selectObject: "TextGrid 'basename$'"
 num_intervals = Get number of intervals: 2
 ;appendInfoLine: "Number of intervals in Tier 2 is: ", num_intervals
```

```
# Loop the interval tiers
for i from 1 to num_intervals

   phone$ = Get label of interval: 2, i

   if phone$ == "ae" or phone$ == "ei"
      ... or phone$ == "eh" or phone$ == "iy"

      startTime = Get start point: 2, i
      endTime = Get end point: 2, i
      midTime = startTime + (endTime - startTime)/2
      duration = (endTime - startTime)*1000

      ##################
      # IDENTIFY WORD
      ################
      word_index = Get interval at time: 1, midTime
      word$ = Get label of interval: 1, word_index
      word_start = Get starting point: 1, word_index
      word_end = Get end point: 1, word_index

      word_duration = (word_end - word_start)*1000

      appendInfo: basename$,tab$,phone$,tab$,word$,tab$,fixed$(duration,2),tab$
      appendInfo: fixed$(word_duration,2),tab$

      # output
      appendFile: output$, basename$,tab$,phone$,tab$,word$,tab$
      appendFile: output$, fixed$(duration,2),tab$
      appendFile: output$, fixed$(word_duration,2),tab$

   endif
endfor
```

```
#####################
# Measure F0
#####################
selectObject: "Pitch 'basename$'"
f0 = Get value at time: midTime, "Hertz", "Linear"
# f0 Hz to semitone
f0st = hertzToSemitones(f0)
appendInfo: fixed$(f0,2),tab$,fixed$(f0st,2),tab$
#output
appendFile: output$, fixed$(f0,2),tab$,fixed$(f0st,2),tab$

##############################
# Measure F1 and F2
##############################
selectObject: "Formant 'basename$'"
f1 = Get value at time: 1, midTime, "Hertz", "Linear"
f2 = Get value at time: 2, midTime, "Hertz", "Linear"

f1_Bark = hertzToBark(f1)
f2_Bark = hertzToBark(f2)

appendInfo: fixed$(f1,2),tab$,fixed$(f2,2),tab$
appendInfo: fixed$(f1_Bark,2),tab$,fixed$(f2_Bark,2)

#output
appendFile: output$, fixed$(f1,2),tab$,fixed$(f2,2),tab$
appendFile: output$, fixed$(f1_Bark,2),tab$,fixed$(f2_Bark,2),tab$

#####################
# Measure Intensity
#####################
selectObject: "Intensity 'basename$'"
dB = Get mean: startTime, endTime, "dB"
appendInfoLine: fixed$(dB,2)
#output
appendFileLine: output$, fixed$(dB,2)
selectObject: "TextGrid 'basename$'"
```

uksk_19_mutiple_final.praat

Any questions?