

Praat Scripting

02 Fundamentals

1. Commands and History Function

- Commands are predefined instructions in order to perform specific tasks
- Most commands start with an upper-case letter (some exceptions, e.g. , *writeInfoLine*, *clearinfo*, ...)
- All things that you do **manually** in Praat are saved internally and can be looked at using the history function¹
- In April 2013 the developers Paul Boersma and David Weenink started to integrate a new syntax in Praat and changed it again in January 2014

```
selectObject: "TextGrid vowels"      select TextGrid vowels
Get end point: 1, 1                  Get end point... 1 1
```

- All commands that need additional arguments are used with a colon (they will be displayed as ... in Praat's user interface!)
- String content is enclosed by double quotes, e.g. `Read from file: "/Users/Praat/vowels.wav"`
- Generally, the old syntax can still be used but the history function only displays the new syntax (unless you have installed an older version of Praat) but there are a few functions that you can't use anymore within the newer syntax (but there are new/different functions available)

2. Comments

- Comments are information which aren't considered actual input
- Comments can start with a # or ;
- Put comments in a new line and not directly after a command line

```
# This line is a comment
a = 1 + 2
; we just did math, and this is another comment
```

¹Open a new Praat script window. Print the history by using *Edit* → *Paste history* or the shortcut *Ctrl+H*

3. Variables

- A variable is a storage location which contains numeric or string information
- **Variables containing string information have to end in a \$ sign and the content has to be enclosed in double quotes**
- You can use variables within another variables and concatenate string variables with additional text using the + symbol

```
x = 99
x2 = x * x
```

```
title$ = "Dutch nasal place assimilation"
sentence$ = "Title of the talk: " + title$
```

- There are simple but important rules to follow when choosing names for variables
- They have to ...
 - ... start with a lower-case letter
 - ... contain only letters (upper or lower-case), digits, dots, and underscores
 - ... not contain spaces, dashes, punctuation marks (except dots), umlauts, or anything not in the previous case

▷ *Predefined variables*

newline\$	line break
tab\$	tabulator
defaultDirectory\$	directory that contains the script file

4. Output

- You can use two methods to print an output in the *Praat Info* window
 - `writeInfoLine: "Hello world"`
first clears the Info window and prints the content between the quotes
 - `appendInfoLine: "How do you do?"`
does not clear the Info window and prints the content between the quotes
- Printing the content of a variable:
 - `writeInfoLine: "The whole sentence is: ", sentence$, "."`
 - The variable needs to be separated from the string content by commas
- The command `clearinfo` also clears the Info window without printing an output

5. Operators and Functions

- In formulas you can use numerical, comparison, and logical *operators* and mathematical *functions*

addition	+	1+2	3
subtraction	-	3-2	1
multiplication	*	2*3	6
division	/	6/3	2
exponentiation	^	2^3	8
division, rounded	div	10 div 3	3
square root	sqrt ()	sqrt (9)	3
nearest integer	round ()	round (0.5)	1
next lowest integer	floor ()	floor (1.9)	1
next highest integer	ceiling ()	ceiling (0.1)	1
equal	=	5 + 6 = 10	0 (false)
unequal	<>	5 + 6 <> 10	1 (true)
less than	<	5 + 6 < 10	0
greater than	>	5 + 6 > 10	1
less than or equal	<=	5 + 6 <= 10	0
greater than or equal	>=	5 + 6 >= 10	1
not	not	not 5 + 6 = 10	1
and	and	x > 5 and x < 10	x between 6 and 9
or	or	label\$ = "a" or label\$ = "o"	a/o

- There are also functions which for scripting purposes do similar things as operators but tend to involve parentheses

left\$ (a\$, length)	left\$ ("example", 2)	ex
right\$ (a\$, length)	right\$ ("example", 2)	le
mid\$ (a\$, start, length)	mid\$ ("example", 2, 2)	xa
replace\$ (a\$, target\$, replacement\$, n)	replace\$ ("hello", "l", "m", 1)	hemlo
length (a\$)	length ("hello")	5
index (a\$, b\$)	index ("hello", "l")	3
rindex (a\$, b\$)	rindex ("hello", "l")	4
startsWith (a\$, b\$)	startsWith ("camel", "ca")	1 (true)
endsWith (a\$, b\$)	endsWith ("camel", "meal")	0 (false)
selected ()	selected ("TextGrid")	e.g. 1
selected\$ ()	selected\$ ("TextGrid")	TextGrid name
string\$ (number)	formats number into string	
number (a\$)	interprets a string as a number	
fixed\$ (number, precision)	fixed\$ (72.65687, 3)	72.657

6. Exercise 02: First short script

- (a) Read in the Sound `minimal_pairs.wav` and TextGrid file `minimal_pairs.Textrid`
- (b) How many intervals does your TextGrid have? Print the number in the Praat Info window
- (c) What is the duration of the first, second, and third interval? Print the information.

To help you get started, perform the things above manually and either write down the commands you have used in the correct order or use the history function.