While we get ready to start:

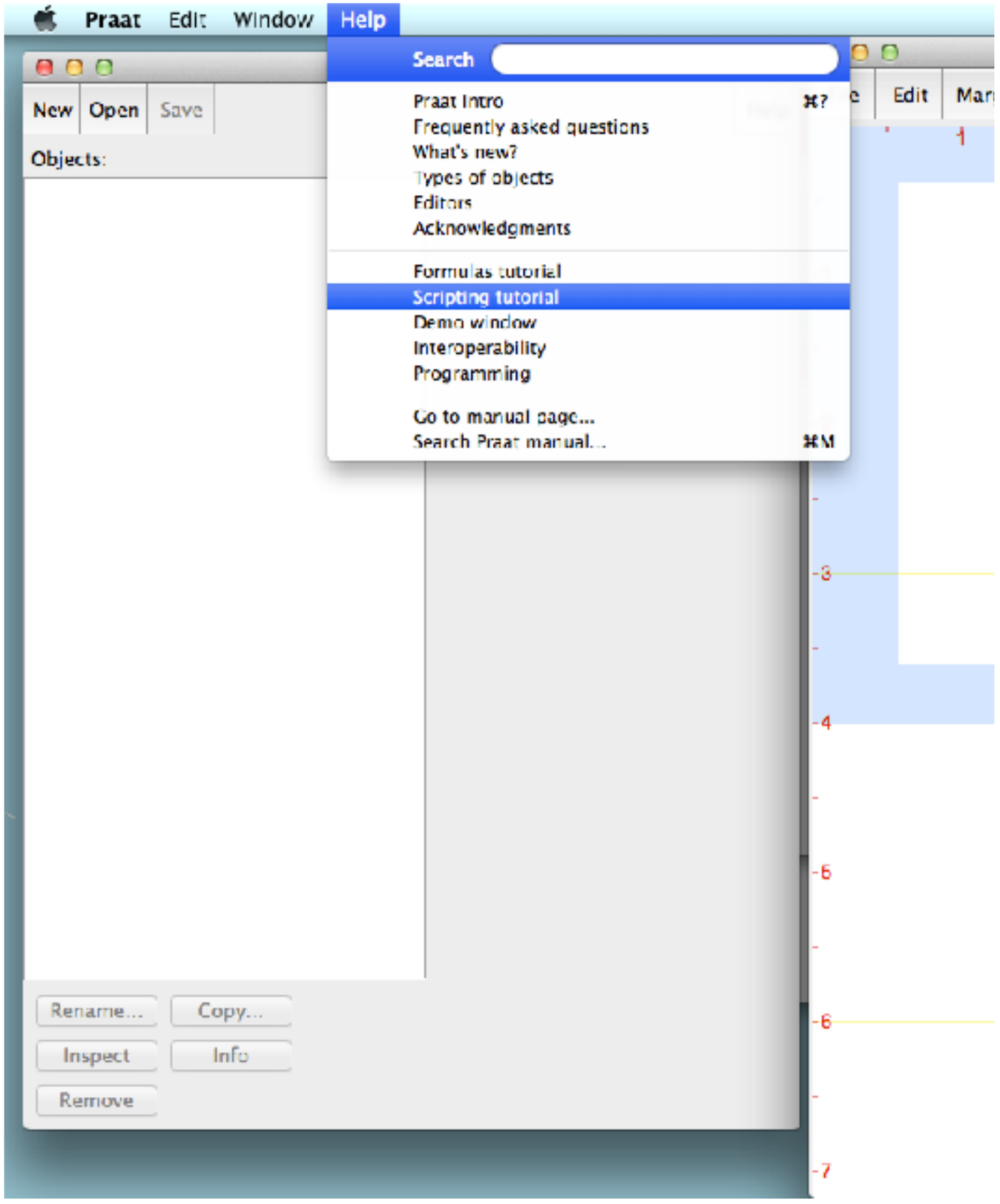
- Log in (if you are using a desktop) and open Praat.
- Download the scripts from the Canvas site (Files > Session 2 > Scripts)

Praat scripting workshop

Session 2

Outline for today

- Variables
- Printing stored values
- Debugging


Praat
Edit
Window
Help
Search
Praat Intro ⌘?
Frequently asked questions
What's new?
Types of objects
Editors
Acknowledgments
Formulas tutorial
Scripting tutorial
Demo window
Interoperability
Programming
Go to manual page...
Search Praat manual... ⌘M
New
Open
Save
Objects:
Rename...
Copy...
Inspect
Info
Remove
Edit
1
3
4
6
6
7

Quick quiz

- How do you add a comment to a Praat script?
- Why do we recommend to make extensive use of #?
- Name three Praat editors.
- What is the difference between writeInfoLine and appendInfoLine?

Variables

A more flexible greeting

```
# store the greeting in a string variable
# (more on these later)

greeting$ = "hello, world!"
writeInfoLine ( greeting$ )
```

A more flexible greeting

```
# or get it at run time in a form

  form Greet the nice user
       comment Please name the entity we should greet:
       text greeting
  endform

writeInfoLine ( "hello, ", greeting$, "!" )
appendInfoLine ( "hello, " + greeting$ + "!" )
appendInfoLine ( "hello, 'greeting$'!" )

# question: what do the plus signs do in the second line?
# question: what are the single quotes for in the third line?
```

Variables

• As we've just seen, a **variable** works like a named bucket that can store an arbitrary value and return it when you call the bucket's name.

• A variable is an address in your computer's memory. This address points at a **value.**

- The variable is on the left side of the equals operator (=) and the value of that variable is on the right side

```
# in Praat, a variable assignment looks like this:

greeting$ = "world!"
```

- We can read this as: the string variable 'greeting' points at the string constant "world!"

Variables (cont.)

- Variables have three attributes:

1. **name**: a unique identifier for the memory address
2. **type**: defines the affordances and behaviors of the variable
3. **value**: the information stored in the variable –can be a literal or constant value typed-into the script, read from the world, or calculated at runtime

Basic variable types

- Praat has two basic variable types:
    - **string variables**, like we have used so far, end in a $ character and can hold anything from the empty string to (at least) four copies of War and Peace.
    - **numeric variables** can be integers or real (decimal) numbers
- As with object types, variable types dictate what actions and behaviors are available for a variable.

Numeric variables

- Praat's numeric variables can hold values between -1,000,000,000,000,000 and +1,000,000,000,000,000 or real numbers between -10^{308} and 10^{308}

- Numeric constants may not contain commas (try it, see what happens)

- Decimal numbers must have at least a zero to the left of the decimal point.

- Numeric assignments look like these: numeric_printtotext1a

```
c = 35000 ; cm/s speed of sound in air length = 17.5 ;
cm vocal tract
f1_ə=(2*1-1)*c/ (4*length) f2_ə = (2 * 2 - 1) * c / ( 4
* length )

writeInfoLine( "According to the tube theory, F1 is ",
f1_ə, " and F2 is ", f2_ə )
```

Naming your variables

- In Praat, variables must begin with a lowercase ascii letter (a through z)
  - ~~1variable~~ but variable1
  - ~~Variable~~
  - ~~ävariable~~ but väriable

Converting variables

- You can convert a string to a number or a number to a string

```
a$ = string$ (a)
```

```
a = number (a$)
```

Printing values to a file

- For now, we are not taking measurements. We're making calculations based on theory

- But… you will take measurements of your data. And when you do, you will probably want to save the stored values in a file, like a .txt file.
  - You will probably use this file for data visualization and statistics

- So let's see how we can do that

Before we start

- Locate the path to your desktop
  - PC vs. Mac

- For example, in my computer, the path to my desktop is:
  - MAC: “/Users/MyFancyMacbook/Desktop/”
  - PC: “C:\\Users\MyFancyMacbook\Desktop\”

We won’t use it next, but hold on to this information
We will use it in a little bit

Printing values to a file

string_printtotext2a

```
string1$ = "Hello it's me"
string2$ = "I was wondering if after all these years you'd
like to meet"

writeInfoLine ( string1$ )
appendInfoLine ( string2$ )
```

Run the script
What happened?

Printing values to a file

string_printtotext2b

```
string1$ = "Hello it's me"
string2$ = "I was wondering if after all these years you'd
like to meet"

writeFileLine: "/Users/MyFancyMacbook/Desktop/
mygreeting.txt", "'string1$'"
appendFileLine: "/Users/MyFancyMacbook/Desktop/
mygreeting.txt", "'string2$'"
```

Run the script
What happened?

Printing values to a file

- More things we can print to a file…

```
c = 35000 ; cm/s speed of sound in air length = 17.5 ; cm vocal tract
f1_ə=(2*1-1)*c/ (4*length) f2_ə = (2 * 2 - 1) * c / ( 4 * length )
writeInfoLine( "According to the tube theory, F1 is ", f1_ə, " and F2 is ", f2_ə )
```

```
c = 35000 ; cm/s speed of sound in air
length = 17.5 ; cm vocal tract
f1_ə = (2 * 1 - 1) * c / (4 * length)
f2_ə = (2 * 2 - 1) * c / ( 4 * length)

writeFileLine: "/Users/MyFancyMacbook/Desktop/result.txt",
"According to the tube theory, F1 is ", f1_ə, " and F2 is
", f2_ə, "."
```

Run the script
What does the output look like?

```
textfile$ = "/Users/MyFancyMacbook/Desktop/
results.txt"

resultline$ = "F1'tab$'F2'newline$'"
fileappend "'textfile$'" 'resultline$'

c = 35000 ; cm/s speed of sound in air
length = 17.5 ; cm vocal tract
f1_ə = (2 * 1 - 1) * c / (4 * length)
f2_ə = (2 * 2 - 1) * c / ( 4 * length)

resultline$ = "'f1_ə''tab$''f2_ə'"
fileappend "'textfile$'" 'resultline$'
```

Run the script
What does the output look like?

Printing values to a file

- We can combine numeric and string variables in our output.

- Search on Google: “length() praat”

- What does “length()” do?

Printing values to a file

- We can combine numeric and string variables in our output.

```
string$ = “hello, it’s me”
length = length( string$ )
writeInfoLine ( "The string '",
string$, "' is ", length,
..." characters long." )
```

```
textfile$ = "/Users/MyFancyMacbook/Desktop/results_string.txt"

resultline$ = "string'tab$'length'newline$'"
fileappend "'textfile$'" 'resultline$'

string$ = "I was wondering if after all these years you'd like
to meet"

length = length( string$ )

resultline$ = "'string$''tab$''length'"
fileappend "'textfile$'" 'resultline$'
```

```
textfile$ = "/Users/MyFanyMacbook/Desktop/results_string.txt"

resultline$ = "string'tab$'length'newline$'"
fileappend "'textfile$'" 'resultline$'

string$ = "I was wondering if after all these years you'd like
to meet"

length = length( string$ )

resultline$ = "'string$''tab$''length'"
fileappend "'textfile$'" 'resultline$'
```