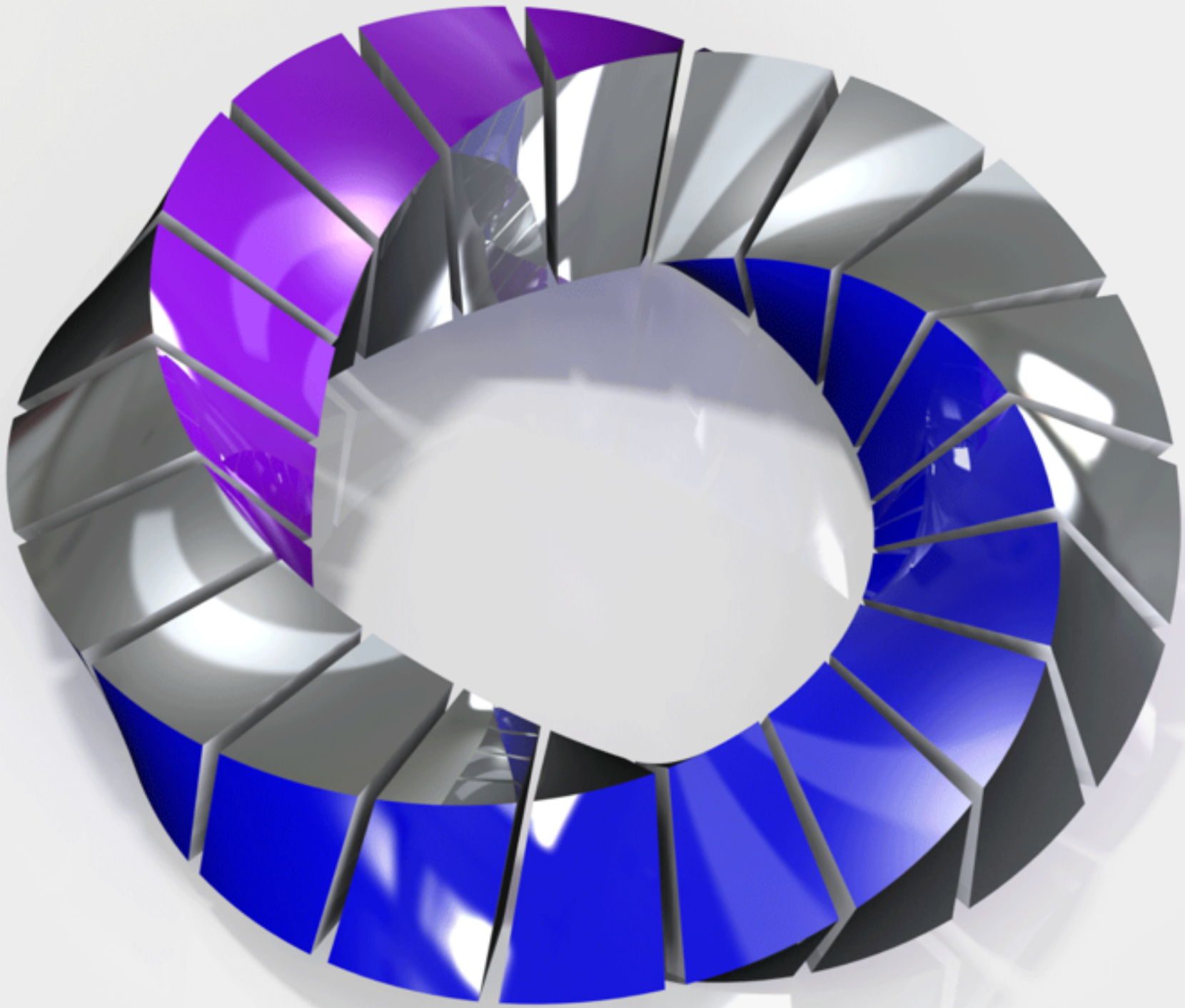
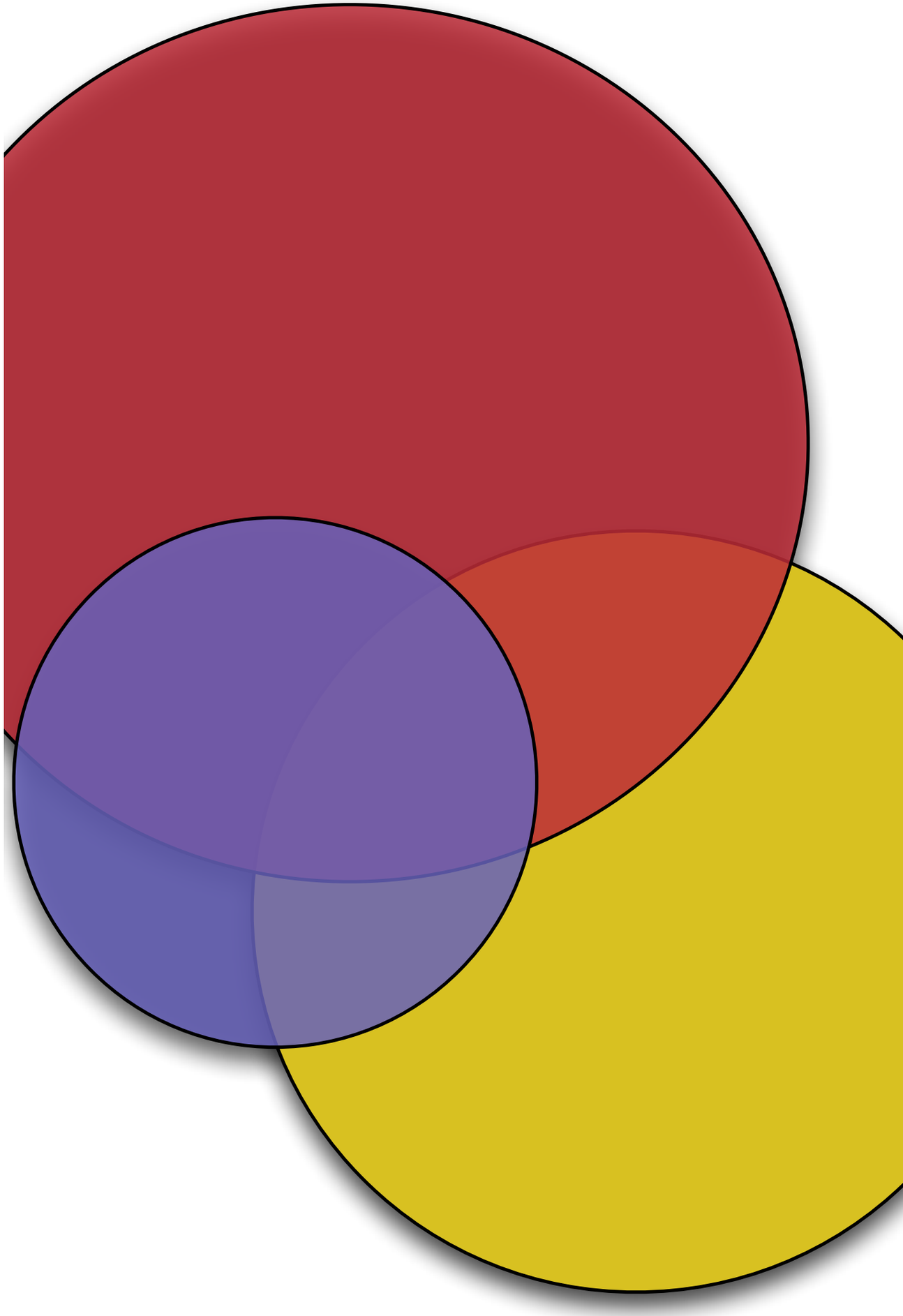


Praat Scripting

Day 2: Textgrids & Loops

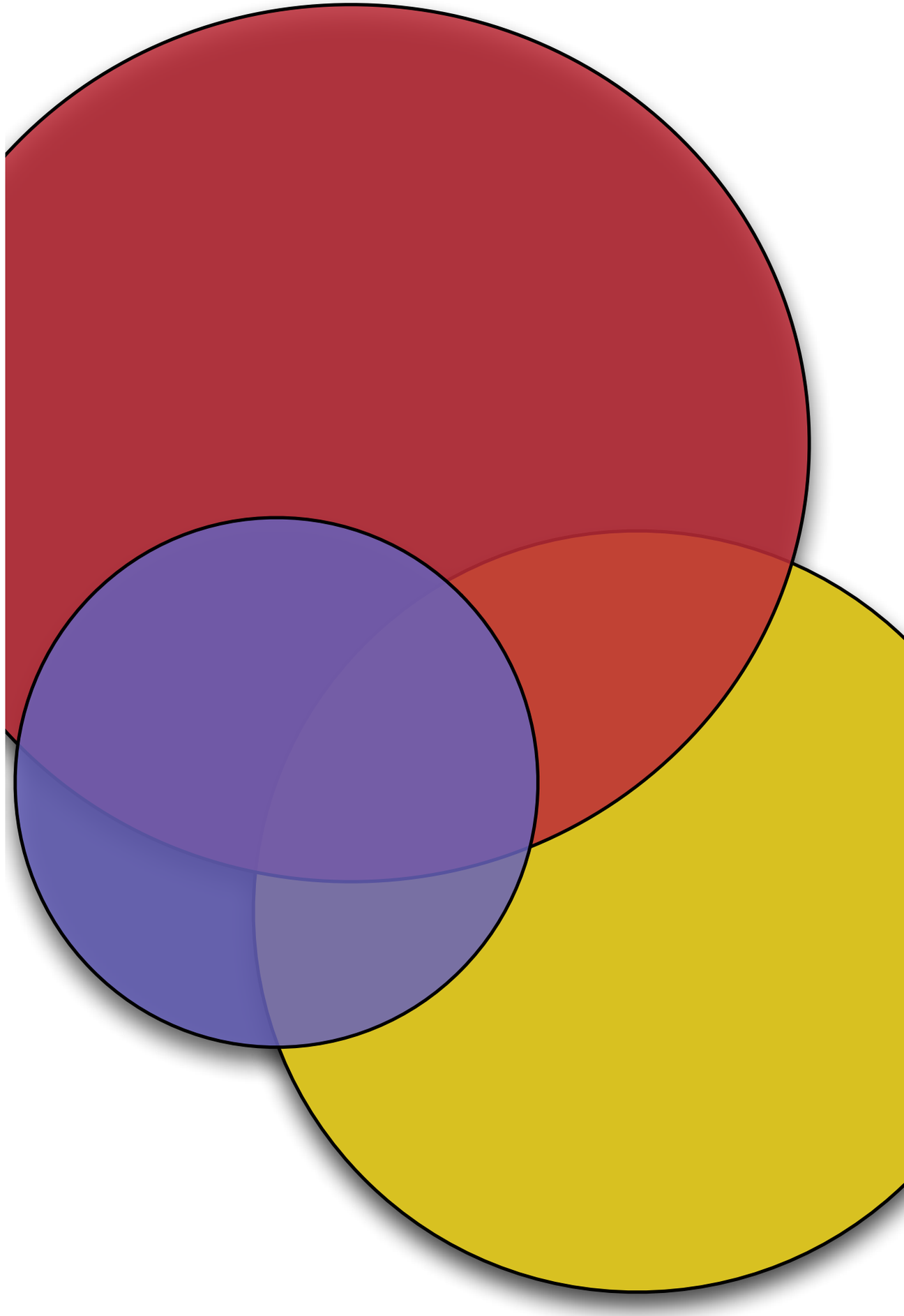
Yesterday we…

- tried to find the *way* of Praat (think like Boersma!)
- looked at many sources of errors and problems in Praat scripts
- talked about debugging as simple problem solving
- practiced using some of the core language (InfoWindow, functions, conditionals with if/elsif/else/endif, etc.)



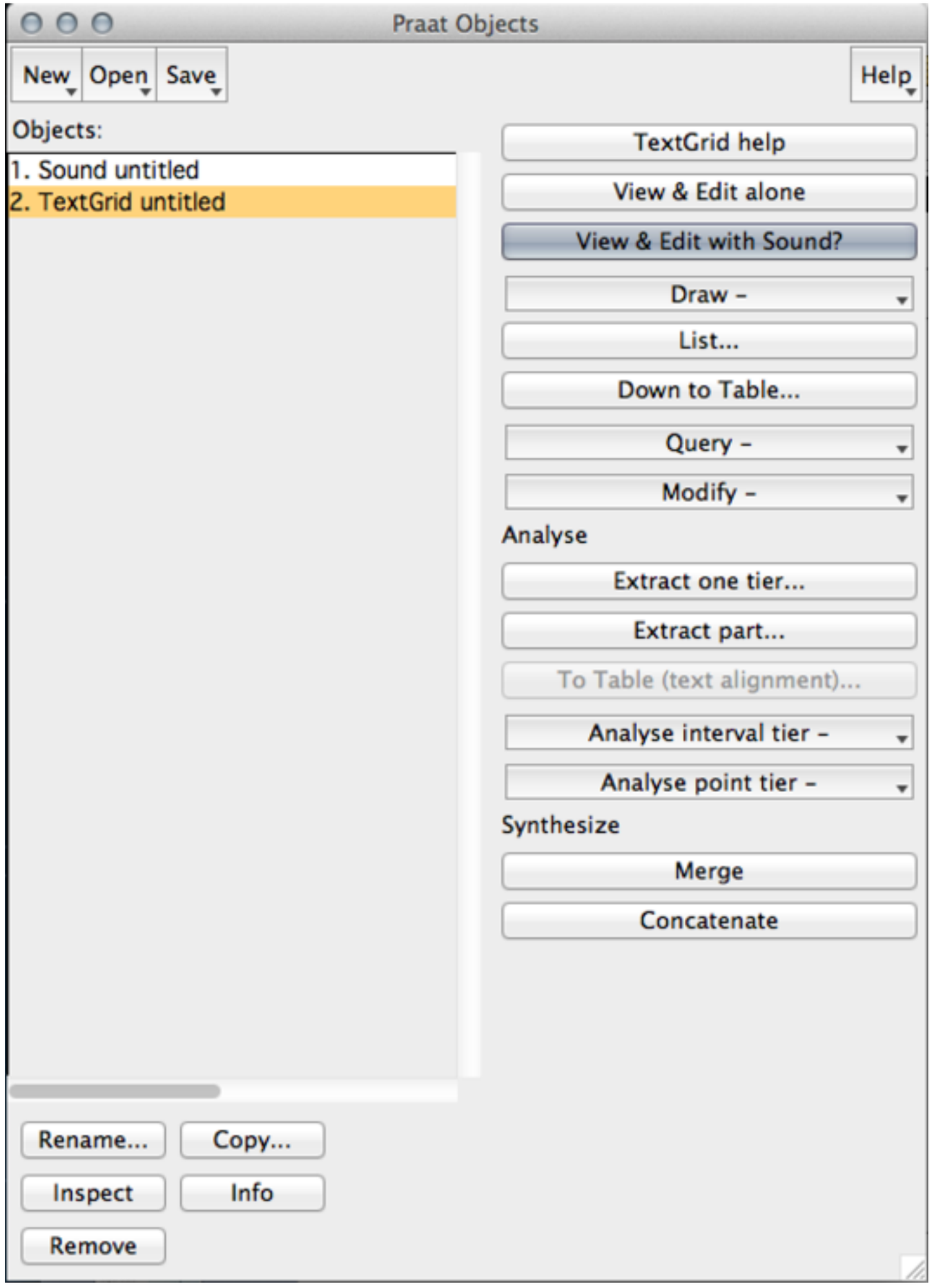
For Today!

- TextGrids, under the hood
- Variable declaration & reassignment
- Iteration
- Praat's three looping constructs
- Using Praat loops to work with TextGrids



TextGrid refresher

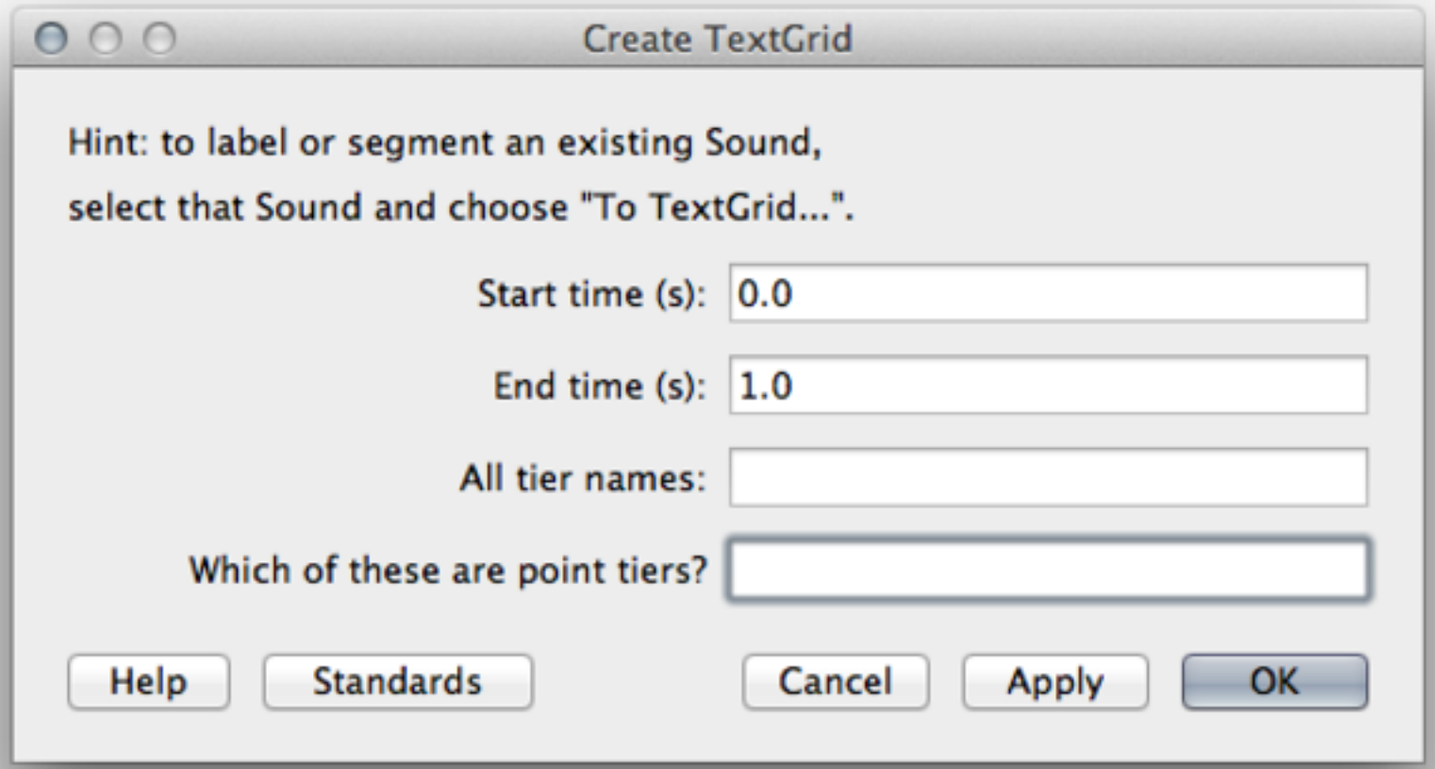
- A **Textgrid** is basically another kind of variable in Praat
- Instead of a simple string$ or number, though, a TextGrid is a multidimensional array
- It allows us to store and access structured data (usually about an associated sound file)
- Imagine your sound file is the delicious topping of a parfait...



TextGrids

- Each TextGrid has:
  - a start time
  - a stop time and
  - 1 or more tiers
- Tiers can be:
  - **IntervalTier**s with a series of **intervals** or
  - **point** tiers: labelled time points (aka **TextTier**s)

Create a TextGrid manually



- Normally a TextGrid is associated with a Sound object
- But it doesn't have to be
- New --> Create TextGrid...

So what?

- TextGrids store your segmentation judgments
- They make it easy to visualize transcriptions alongside their associated Sound objects
- TextGrids also make it easy to query specific intervals or points in your recordings
- And with scripting you can automate this!

Exercise

1. Open a new ScriptEditor window and clear your history

2. Read your sound file from yesterday into a Sound object (or record it again if you didn't save it)

3. Create a new TextGrid for this sound object (Annotate --> To TextGrid...)

4. Name the tiers: `words` and `onsets`

5. Create `onsets` as a point tier and click OK

6. Select both the Sound object and the TextGrid object

    1. Macintosh: command + click

    2. Windows or Linux: control + click

7. Paste command history into your ScriptEditor window

Create a TextGrid

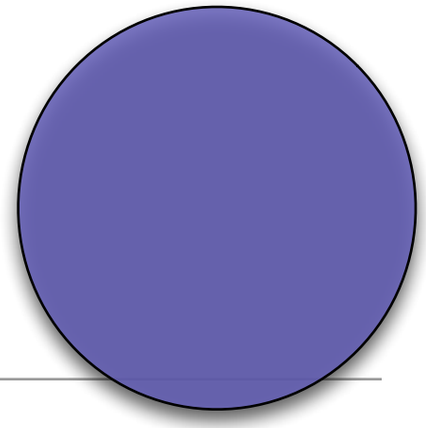
selectObject: "Sound untitled”
snd = selected: "Sound"
tg = To TextGrid: "words onsets", "onsets"

writeInfoLine: snd, tg
to View and Edit a TextGrid, you select two objects.
selectObject: snd
plusObject: tg

question: what hidden action must selectObject: be performing?

selectObject() ; try this command (parens needed still 6-25-14)

Quick Exercise: Create some intervals



- In the TextGridEditor...

- Select (with the mouse) the Sound part of the display where the interval should be

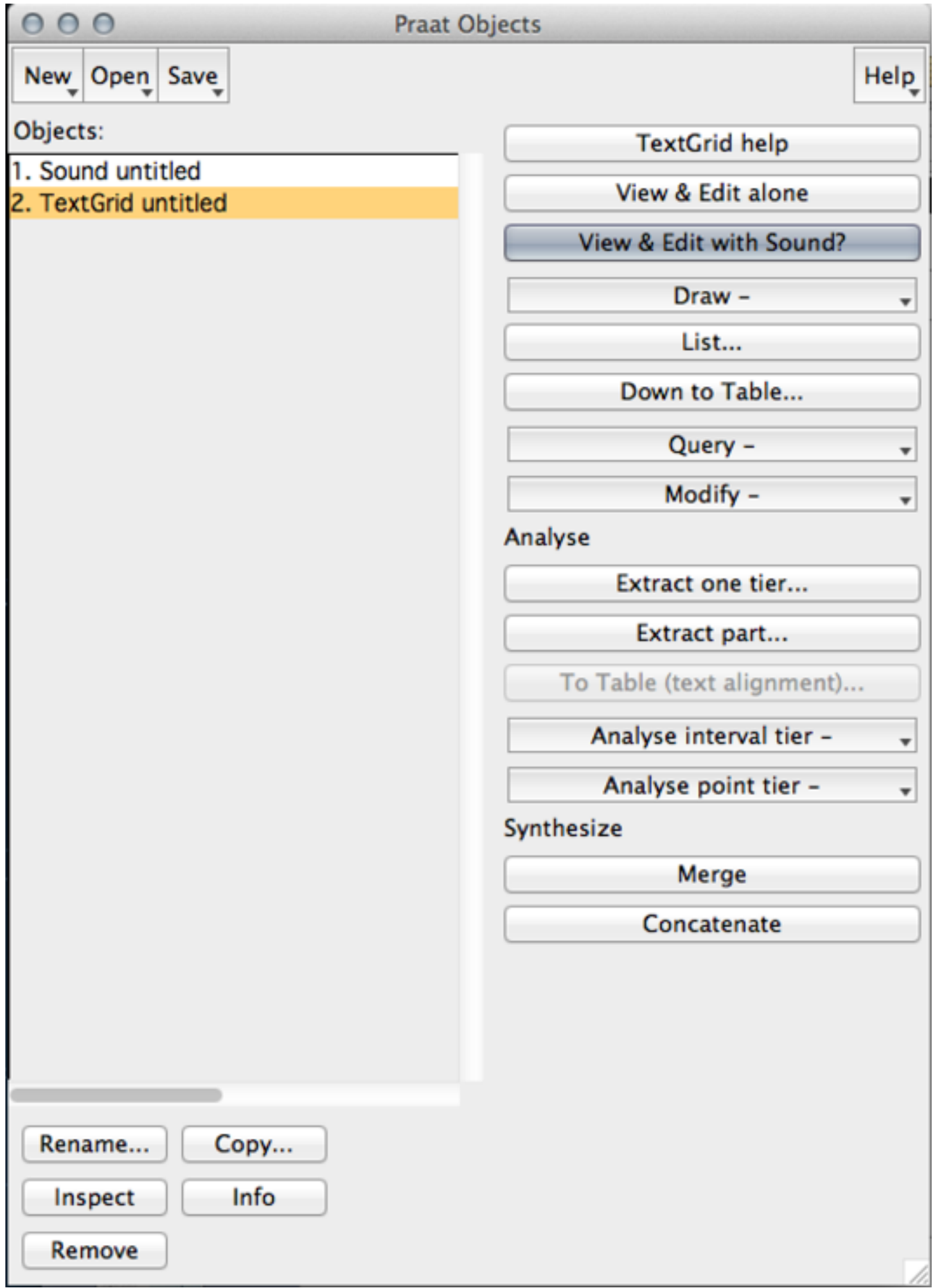
- From the `Interval` menu, select `Add interval on tier 1 (or just press [return], or just type ⌘1)`

- Start typing to add a text label (the interval you just created is already implicitly selected)

- Save as a text file

TextGrid objects

- Let's explore the TextGrid object
- What does that List… button do?
- Down to Table…
- Modify
- What if I asked you to extend your Spectrogram painting script from yesterday to include a TextGrid?



Envisioning TextGrid Structure

xmin xmax

xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""
item[ ]								

Envisioning TextGrid Structure

xmin xmax

xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""
xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""
xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""	xmin xmax text = ""

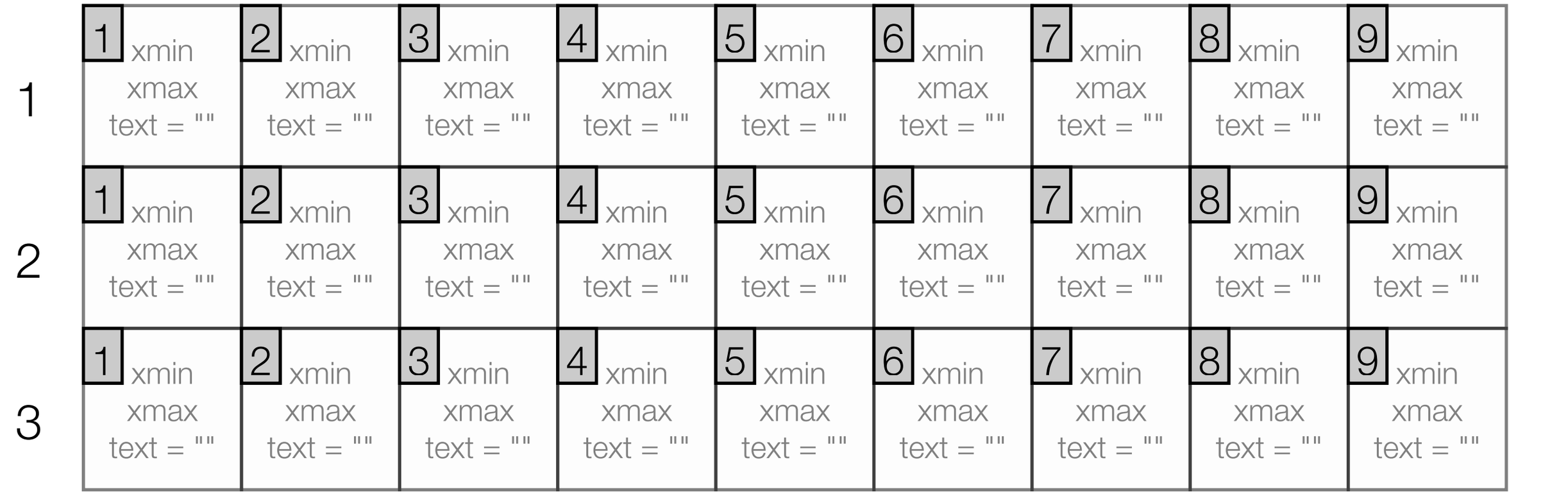
```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 0.4
tiers? <exists>
size = 2
item []:
    item [1]:
        class = "IntervalTier"
        name = "words"
        xmin = 0
        xmax = 0.4
        intervals: size = 1
        intervals [1]:
            xmin = 0
            xmax = 0.4
            text = ""
item [2]:
    class = "TextTier"
    name = "onsets"
    xmin = 0
    xmax = 0.4
    points: size = 0
```

TextGrid contents

- A TextGrid is just a text file!
- **xmin** is the start of the time domain and **xmax** is the end
- **size** is the number of tiers
- Intervals in an IntervalTier also have **xmin**, **xmax** but also **text**

Envisioning TextGrid Structure



Extended Exercise 2: data from a TextGrid

1. Download and expand day2Code.zip, open “tobi TextGrids” folder
2. Load the tobi.wav and tobi.TextGrid files into the Objects window
3. Write a script to:
    1. Print the duration and number of tiers to the Info window
    2. Then for each tier, do the following:
        1. Print the tier number
        2. If this tier is an interval tier:
            1. Print the number of intervals
        3. Else (it is a point tier)
            1. Print the number of points

Ea 2 Solution (part 1 of 4)

```
# hw2-key.praat : read TextGrid file and report
# some attributes

tGrid = Read from file: "tobi/tobi.TextGrid"
totalDuration = Get total duration
writeInfoLine: "Total duration: ", fixed$(totalDuration, 2), "
seconds"

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

# print out tier information one tier at a time
```

Ea 2 Solution (part 2 of 4)

```
name$ = Get tier name: 1
appendInfoLine: tab$, "Tier 1: """, name$, """"

if do( "Is interval tier...", 1 )
     intervals = Get number of intervals: 1
     appendInfoLine: tab$, tab$, "there are ", intervals,
     ..." intervals."
else
     points = Get number of points: 1
     appendInfoLine: tab$, tab$, "there are ", points,
     ..." points."
endif
```

Ea 2 Solution (part 3 of 4)

```
name$ = Get tier name: 2
appendInfoLine: tab$, "Tier 2: """, name$, """"

if do( "Is interval tier...", 2 )
     intervals = Get number of intervals: 2
     appendInfoLine: tab$, tab$, "there are ", intervals,
     ..." intervals."
else
     points = Get number of points: 2
     appendInfoLine: tab$, tab$, "there are ", points,
     ..." points."
endif
```

Ea 2 Solution (part 4 of 4)

```
name$ = Get tier name: 3
appendInfoLine: tab$, "Tier 3: """, name$, """"

if do( "Is interval tier...", 3 )
     intervals = Get number of intervals: 3
     appendInfoLine: tab$, tab$, "there are ", intervals,
     ..." intervals." )
else
     points = Get number of points: 3
     appendInfoLine: tab$, tab$, "there are ", points,
     ..." points."
endif

removeObject: tGrid
```

[do this with a loop]

Praat Scripting

Declarations

Unknown Variable

- So far we've just assigned variables whenever we need them, but

- Praat will give you an "Unknown variable" error if you try to use a variable before you declare it.

- Try something like one of the following:

```
y = m * x + b

text$ = "Please say what this word is " + bit$
```

- What can we say about the errors?

Declaring Variables

- The way around this problem is to always **declare** each variable prior to using it. We could have fixed our line formula with something like:

```
deltaX = 2

deltaY = 4

b = 1

m = deltaY / deltaX

x = 8

y = m * x + b

writeInfoLine: "y equals ", y, " when x is ", x
```

Unknown Variable Example

- You'll also get this Unknown Variable error if you try to use a variable in a test prior to assigning it any value, e.g.

```
if length( variable$ )
    writeInfoLine: "It exists and it is """, variable$, """"
else
    writeInfoLine: "Variable has not been set yet"
endif
```

Exercise

- Please fix error.praat (you can download it with day2Code.zip)

```
# error.praat - generates an error

if length( variable$ )
    writeInfoLine: "It exists and it is """, variable$, """"
else
    writeInfoLine: "Variable has not been set yet"
endif
```

Praat Scripting

Introduction to loops

Lather, rinse, repeat

- You, no doubt, are familiar with the concept of a **loop** from life.

- You're also smart enough to know not to keep lathering, rinsing, and repeating until you run out of shampoo (or hair [or both!]).

- What we need is some way to tell Praat to **iterate** a specified number of times or until some test condition is satisfied

Iteration

- Iteration is simply the ability for a programming language to specify that certain blocks of code be executed over and over

- We will see two types of loops:

- **list iteration:** In Praat, **for** loops allow us to say "repeat this block for each number from n to k"

- **conditional iteration:** In Praat, **while** and **until** loops allow us to specify a test condition and either repeat our block of code until that test condition is false or until it is true.

List Iteration

- e.g. for each house on the street from 980 to 1422
    - knock on door,
    - yell, "trick or treat",
    - receive candy,
    - say, "thank you!"
- Notice that there are three parts:
    1. Variable representing the thing you are operating on (here the 'house'),
    2. A starting number, and
    3. An ending number

List Iteration

- The variable will be a numeric variable.
- The starting number can be any Praat expression that returns a number
- The ending number can also be any Praat expression that returns a number

100 coin tosses with ‘for’


```
# coins.praat

heads = 0
tails = 0

for toss from 1 to 100
     if randomInteger(1, 2) = 1
          heads = heads + 1
     else
          tails = tails + 1
     endif
endfor

writeInfoLine: "Praat simulated ", heads, " heads & ",
...tails, " tails."
```

Loading Files

```
# loader.praat - load all of the (lowercase) .wav files
# from dir$ into the Objects list

dir$ = "stimuli/"
strings = Create Strings as file list: "list",
...dir$ + "/*.wav"

numberOfFiles = Get number of strings

for ifile to numberOfFiles
    selectObject: strings
    fileName$ = Get string: ifile
    Read from file: dir$ + "/" + fileName$
endfor

removeObject: strings
```

Exercise: Plotting Slope & Intercept (1/2)

```
# slope.praat - use slope + intercept to draw points and line

# set up Picture window and draw x & y axes
Erase all
Select outer viewport: 0, 6, 0, 6
Axes: 0, 30, 0, 30
Draw line: 0, -30, 0, 30
Draw line: -30, 0, 30, 0

# clear the info window
writeInfoLine()

# continues on next slide...
```

Plotting Slope & Intercept (2/2)

```
# slope and intercept formula for a line
deltaX = 2
deltaY = 4
b = 1
x = 0
m = deltaY / deltaX
y = m * x + b

appendInfoLine: "y equals ", y, " when x is ", x

# challenge 1: can you make this plot 10 points?
Draw circle...", x, y, 0.5

# challenge 2: can you draw a line through the points?
# Draw line: startingX, startingY, finalX, finalY

# challenge 3: what if you wanted the points and line to
# appear to continue to infinity?
```

Extended Assignment 3:

- Part 1: modify the tobi reporting script you wrote earlier to use a loop
- Part 2: (optional, outside of class) write a script to:

    1. Read through a directory of files (in day2Code/ea3Files)
    2. Load each wav file into the Objects window
    3. Check if each wav file has a corresponding TextGrid file
        1. If it does:
            1. open the TextGrid file as a TextGrid object
        2. if it does not:
            1. create the TextGrid object (see the existing TextGrid for tier names and types)
            2. Save it to a text file
    4. E-mail me the script for comments (we will do this together in class tomorrow)

Praat Scripting

Conditional Looping

Conditional Iteration

- e.g. while the sign is a red hand, continue not to walk

- The test for a loop can be any of the tests that we have used for conditionals so far

- In other words... any Praat expression or variable that will be 0 when false and something else when true

Conditional Iteration

- Praat lets you put your test at the end or the beginning of your loop...
    - **repeat/until *test*** always executes at least once, ends when the test is true
    - **while *test*/endwhile** may execute zero times!
- Which one you use when is often a question of taste (and readability)

Repeat Until Boxcars

```
# simulate a pair of six-sided dice until we roll two sixes
throws = 0

repeat
      pips = randomInteger(1, 6) + randomInteger(1, 6)
      throws = throws + 1
until pips = 12  ; total is 12 when both dice are sixes

writeInfoLine: "It took ", throws, " trials to reach ", pips,
..." with two dice."
```

Exercise: while not boxcars

```
# can you modify this code (boxcars.praat) to use
# while/endwhile instead of repeat/until?

throws = 0

repeat
     pips = randomInteger(1, 6) + randomInteger(1, 6)
     throws = throws + 1
until pips = 12  ; total is 12 when both dice are sixes

writeInfoLine: "It took ", throws, " trials to reach ", pips,
" with two dice."
```

While not Boxcars (solution)

```
throws = 0
pips = 0

while pips <> 12 ; total is 12 when both dice are sixes
      pips = randomInteger(1, 6) + randomInteger(1, 6)
      throws = throws + 1
endwhile
writeInfoLine: "It took ", throws, " trials to reach ", pips,
..." with two dice."
```

Return to TextGrids

Exercise: enloopification

- Please convert your solution to ea2 (the tobi reporting script) to use a single loop (feel free to iterate a list or to use a conditional loop) instead of those three repeated blocks. [reminder]

- You can find ea2-key.praat in day2Code.zip

Enloopification (a solution)

```
for tier from 1 to numberOfTiers
     name$ = Get tier name: tier
     appendInfoLine: tab$, "Tier ", tier, ": """,
     ...name$, """"

     if do ( "Is interval tier...", tier )
          intervals = Get number of intervals: tier
          appendInfoLine: tab$, tab$, "there are ", intervals,
          ..." intervals."
     else
          points = Get number of points: tier
          appendInfoLine: tab$, tab$, "there are ", points,
          ..." points."
     endif
endfor

appendFile: "foo.txt", info$()
```

Save info window contents to text file!

Praat Scripting

Nested Loops

Nested loops

- Imagine you are decorating cookies. If you have multiple trays of cookies, your solution will be a nested loop!
- For each cookie tray:
  - move tray to table
  - while there are undecorated cookies on the tray
    - take a cookie
    - spread white frosting
    - let cool slightly
    - add ears, eye, and nose
    - place on drying rack
- move empty tray to sink

Exercise: Data from a TextGrid

- Please open the three files in day2Code/tobi/
- Please modify the loopified version of ea2 to add the following:
    - for each interval tier
        - if the tier is called "words"
            - for each interval on the tier
                - if the interval has a label set
                    - write that label to the info window
                    - get the start time and end for the labeled interval
                    - write the start and stop times to the Info window

Exercise: Data from a TextGrid (optional)

- Optionally, modify the loopified version of ea2 to add the following:
  - for each point tier
    - for each point
      - get the time and label of the point
      - write point, time, and label to the Info window

Extended Exercise: tokenize the Sound object

- Once you have the first exercise completed
- Try to use the label along with start and stop times from the "words" tier to save the sound associated with that interval to separate wav files.
- Hint: select a Sound object and see what the command: Convert --> Extract part... can do.
- Hint 2: Save as WAV file: "tobi/" + label$ + ".wav"

Optional Exercise: Extending a TextGrid

- After recording my sentences and carefully marking up my TextGrid, a student points out that an important example sentence is missing from tobi.wav

- So I record `s4.wav`, but I would really like it to be added not only to the existing tobi.wav but also to the TextGrid

- Extending the sound is easy (Objects window, select both sounds, choose `Combine --> Concatenate`)

- But now my TextGrid is too short (the time domain is wrong)

- Can you fix this? (there are several perfectly valid ways)