

Homework 9

posted on/before Mon Nov 14

due Mon Dec 5 in class

over 2 weeks to work on it (not including Thanksgiving)
will be worth more than any other assignment

Homework 10

posted on/before Mon Dec 5

due Wed Dec 14 at noon

in lieu of a final exam

(makes for one fewer homework assignment than I promised)

download from Brightspace

`Psychopy.zip` (Python files)

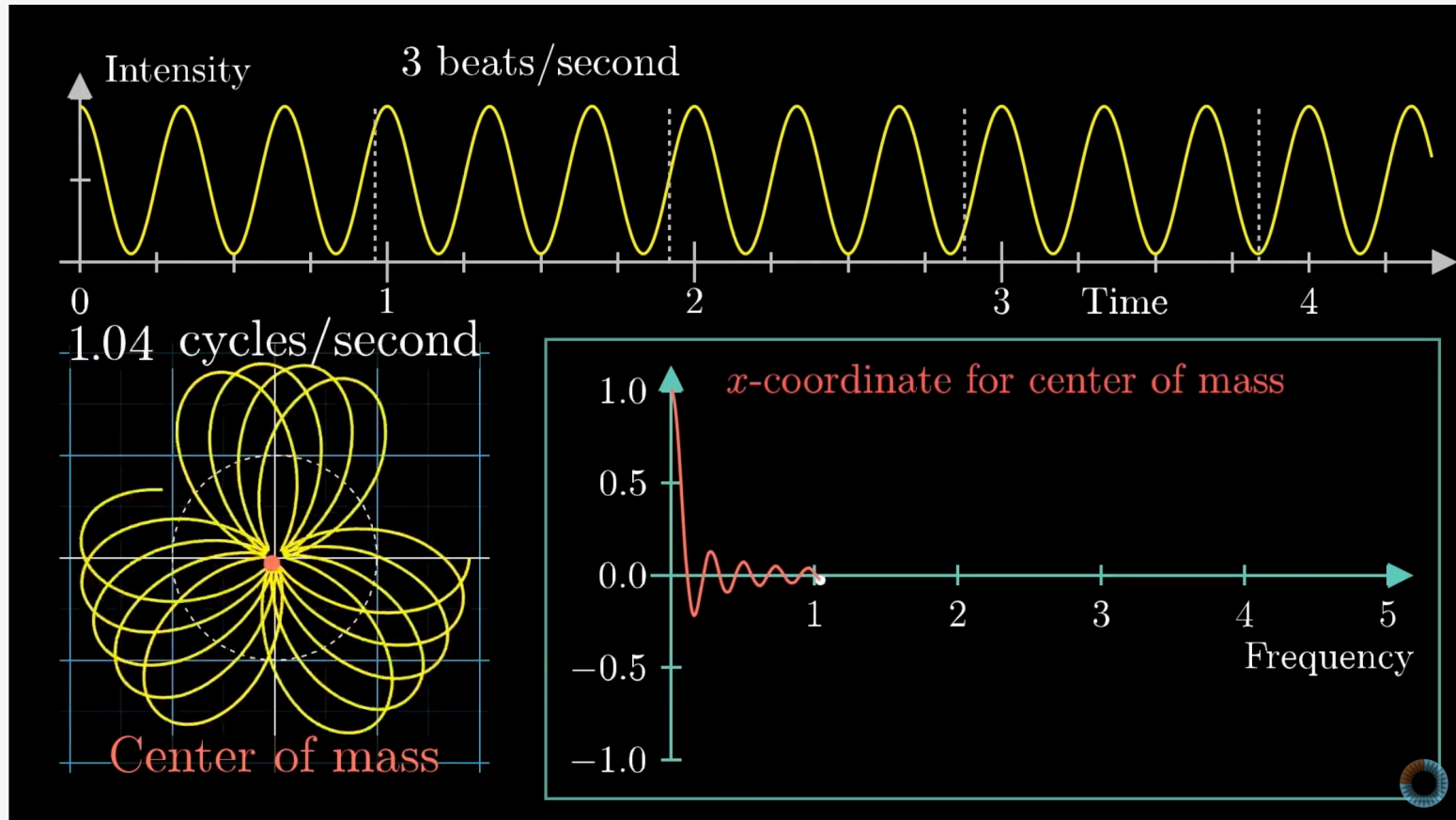
Fourier Analysis

https://en.wikipedia.org/wiki/Fourier_series

<https://www.youtube.com/watch?v=spUNpyF58BY>



Search



But what is the Fourier Transform? A visual introduction.

4,849,053 views • Jan 26, 2018

149K 1.2K SHARE SAVE ...



3Blue1Brown
3.17M subscribers

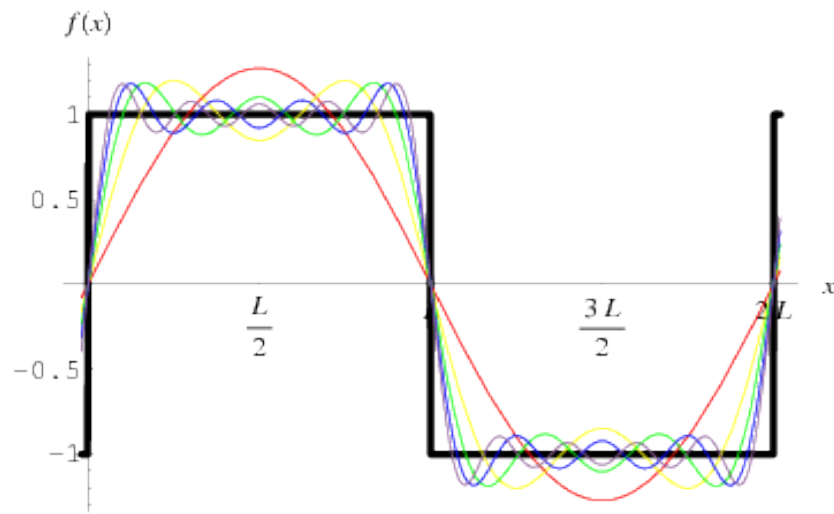
SUBSCRIBED



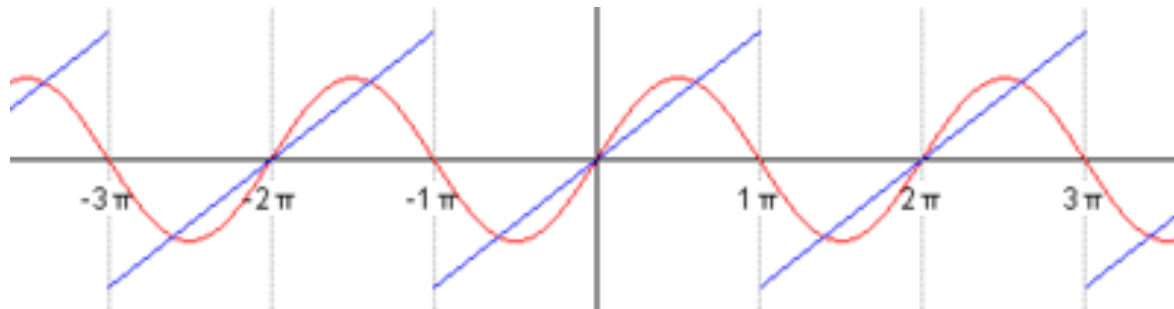
An animated introduction to the Fourier Transform.

Fourier Analysis

(continuous)
any function (1D signal, 2D image, etc.) can be expressed as
linear combination of sinusoids having some amplitude and phase

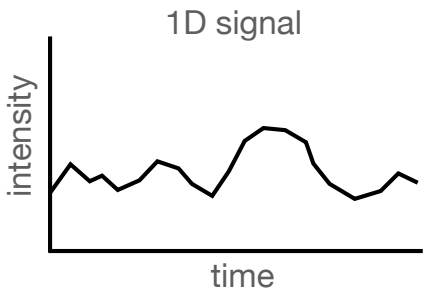


see Python example

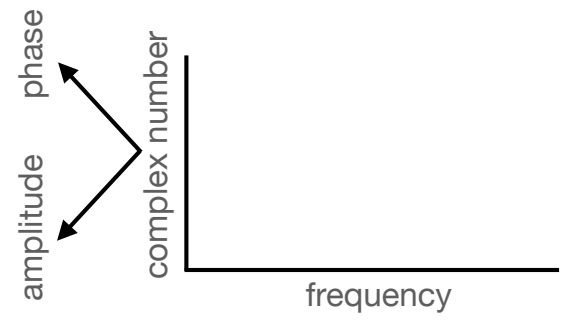


courtesy of Wikipedia

Fourier Analysis

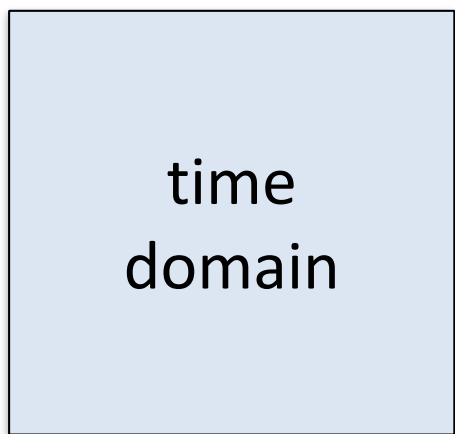


scalp voltage in EEG over time
intensity of sound over time

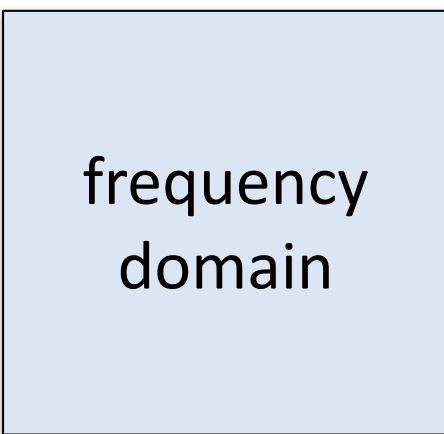


(time is "lost")

"Fourier Transform" (implemented as FFT)



$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i \omega t} dt$$



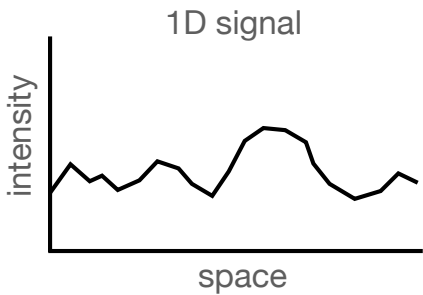
$$f(t) = \int_{-\infty}^{+\infty} F(\omega) e^{2\pi i \omega t} d\omega$$

"Inverse Fourier Transform" (implemented as iFFT)

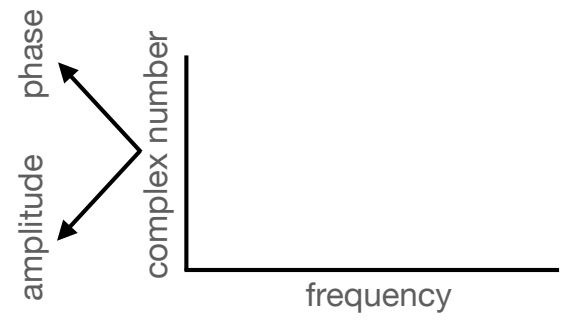
Euler's Formula

$$e^{it} = \cos(t) + i \sin(t)$$

Fourier Analysis



fMRI activation over a row of a scan
image intensity over a row of an image

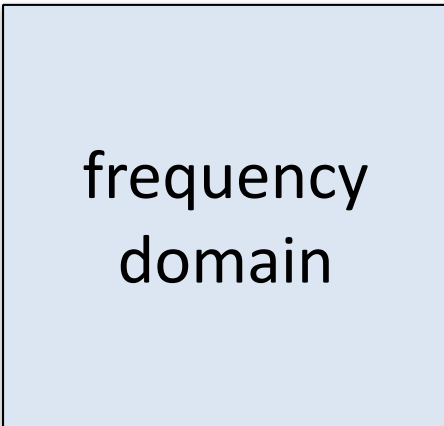


(space is "lost")

"Fourier Transform" (implemented as FFT)



$$\longrightarrow F(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i \omega x} dx \longrightarrow$$



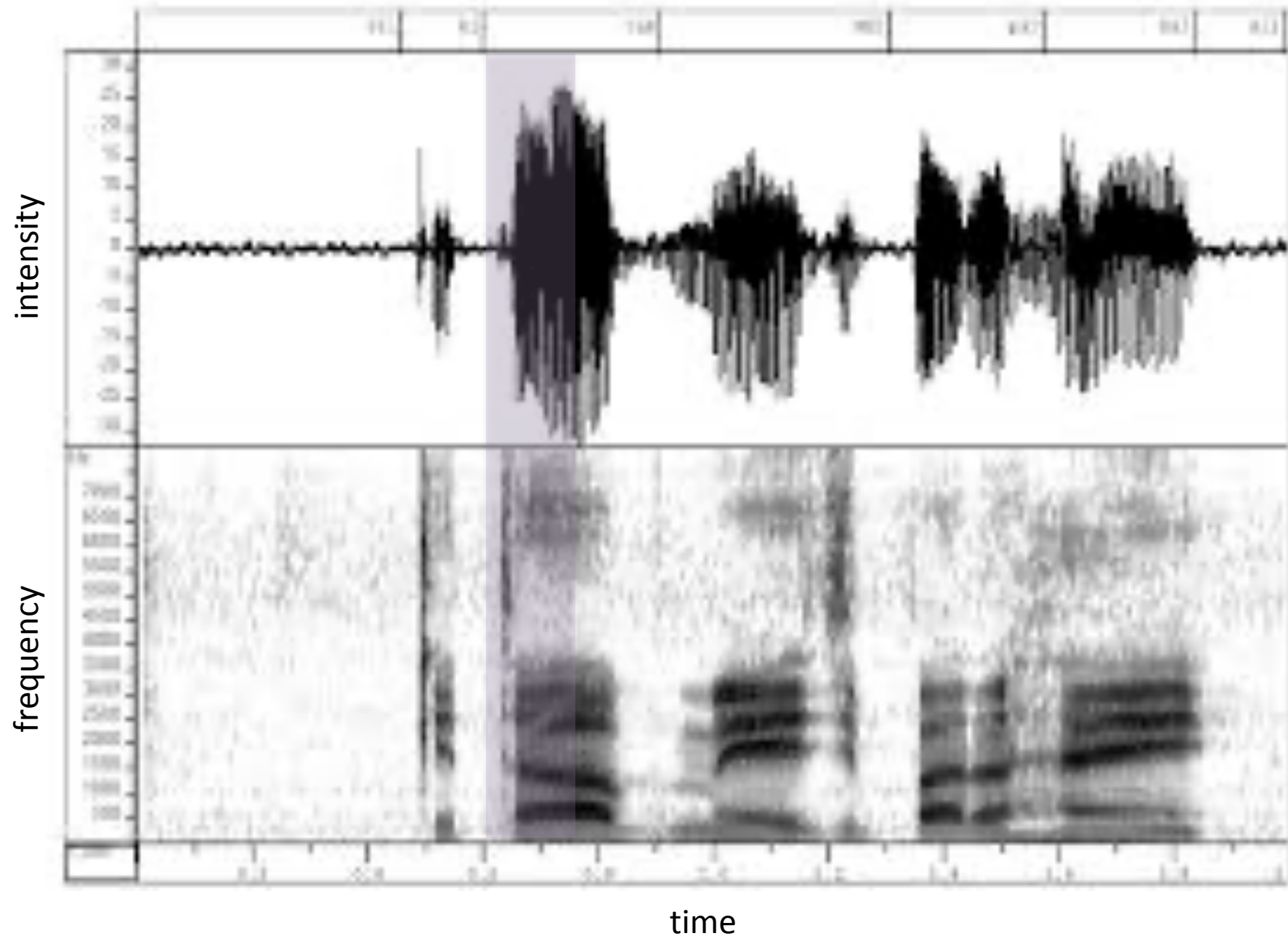
$$\longleftarrow f(x) = \int_{-\infty}^{+\infty} F(\omega) e^{2\pi i \omega x} d\omega \longleftarrow$$

"Inverse Fourier Transform" (implemented as iFFT)

Euler's Formula

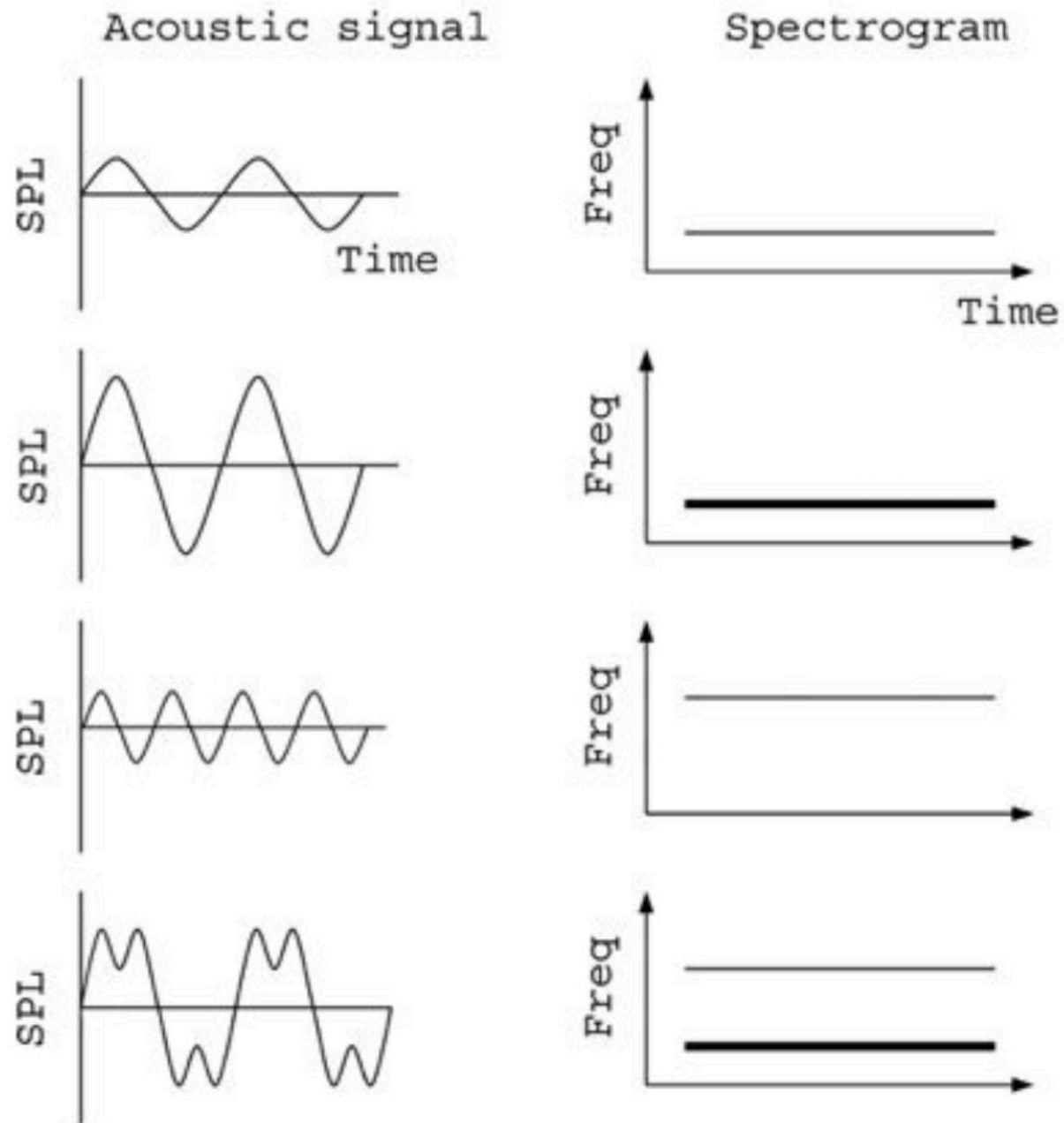
$$e^{ix} = \cos(x) + i \sin(x)$$

speech spectrograms



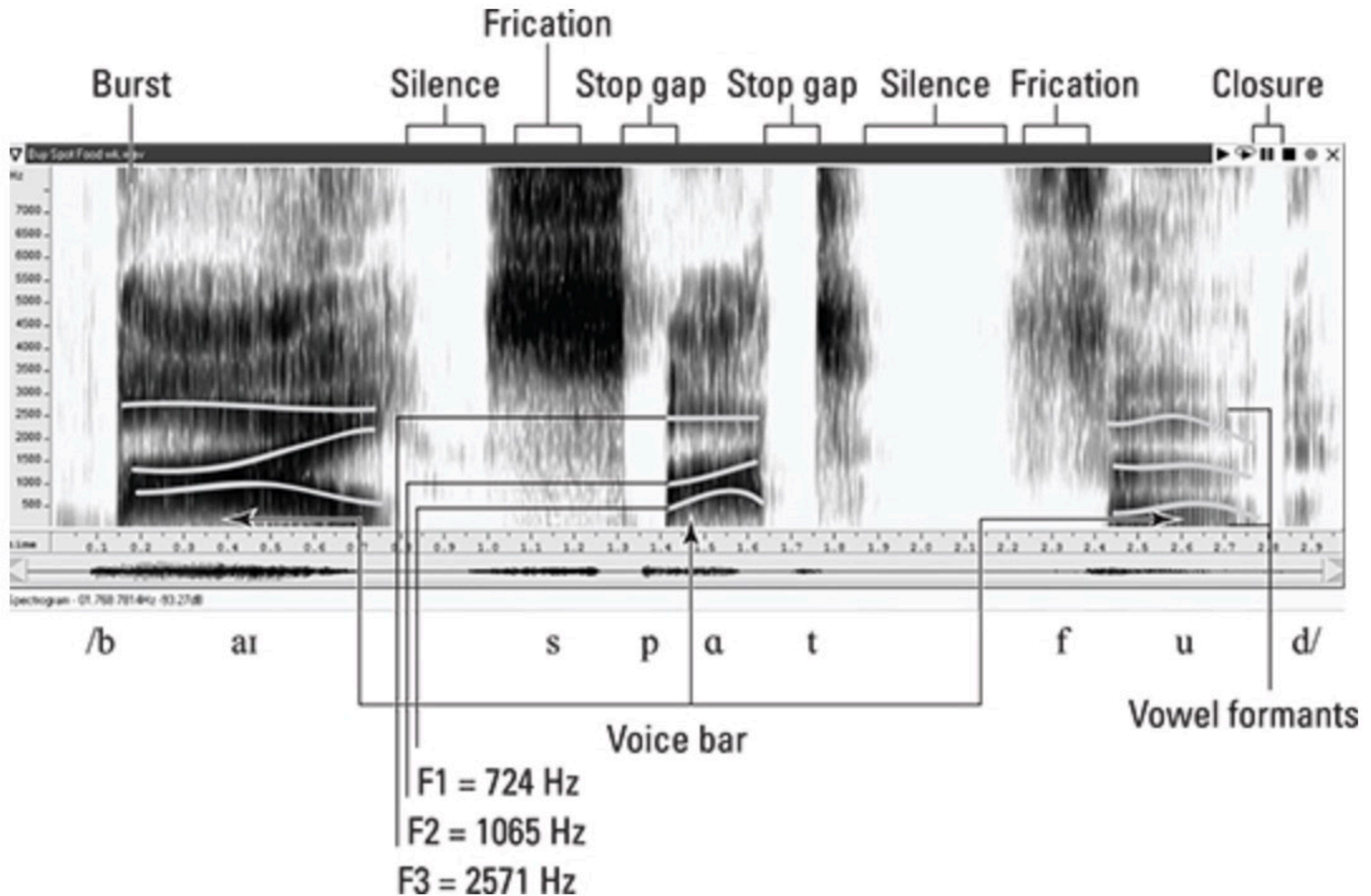
Fourier analysis over sliding windows of time

speech spectrograms



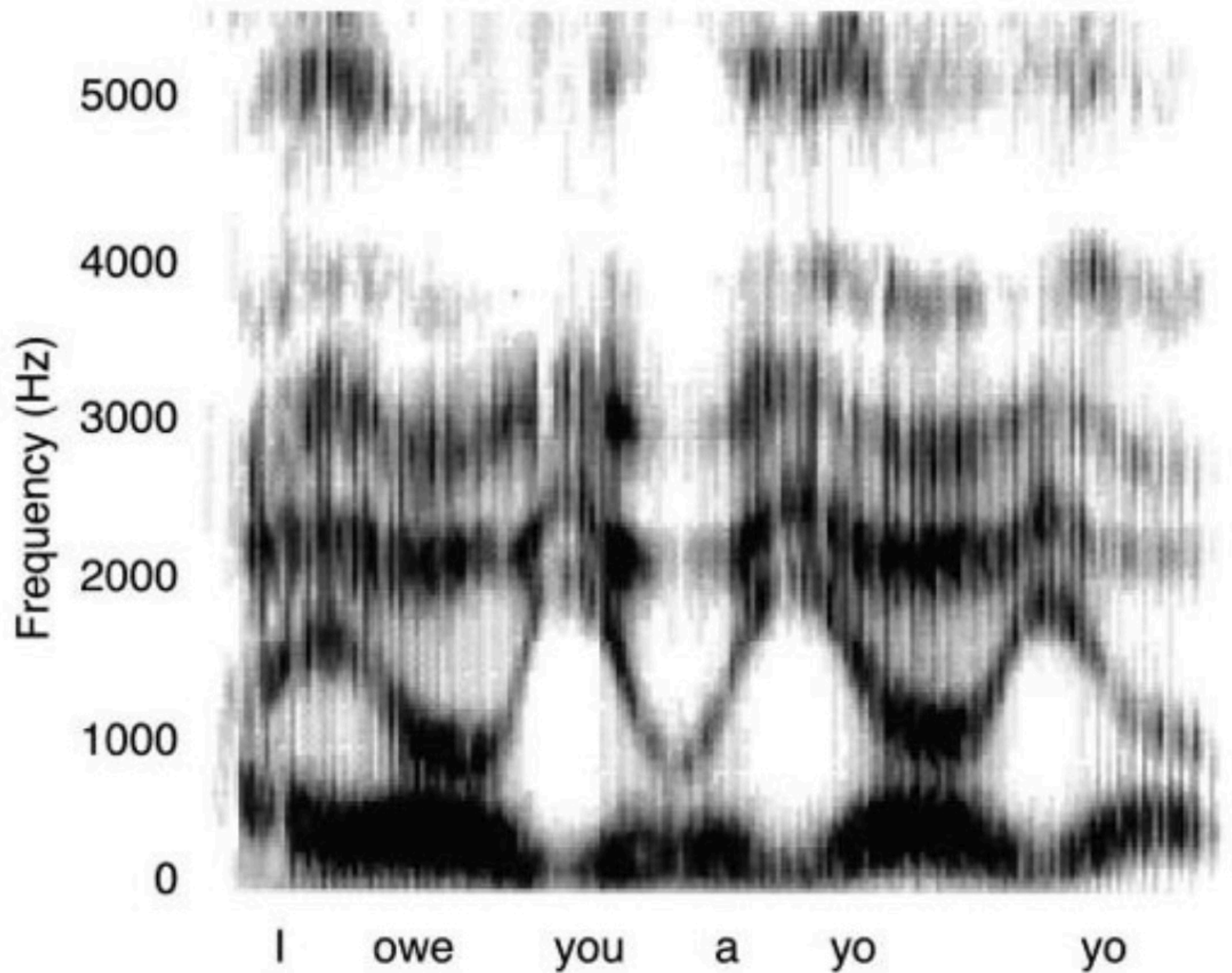
speech spectrograms

"buy spot food"

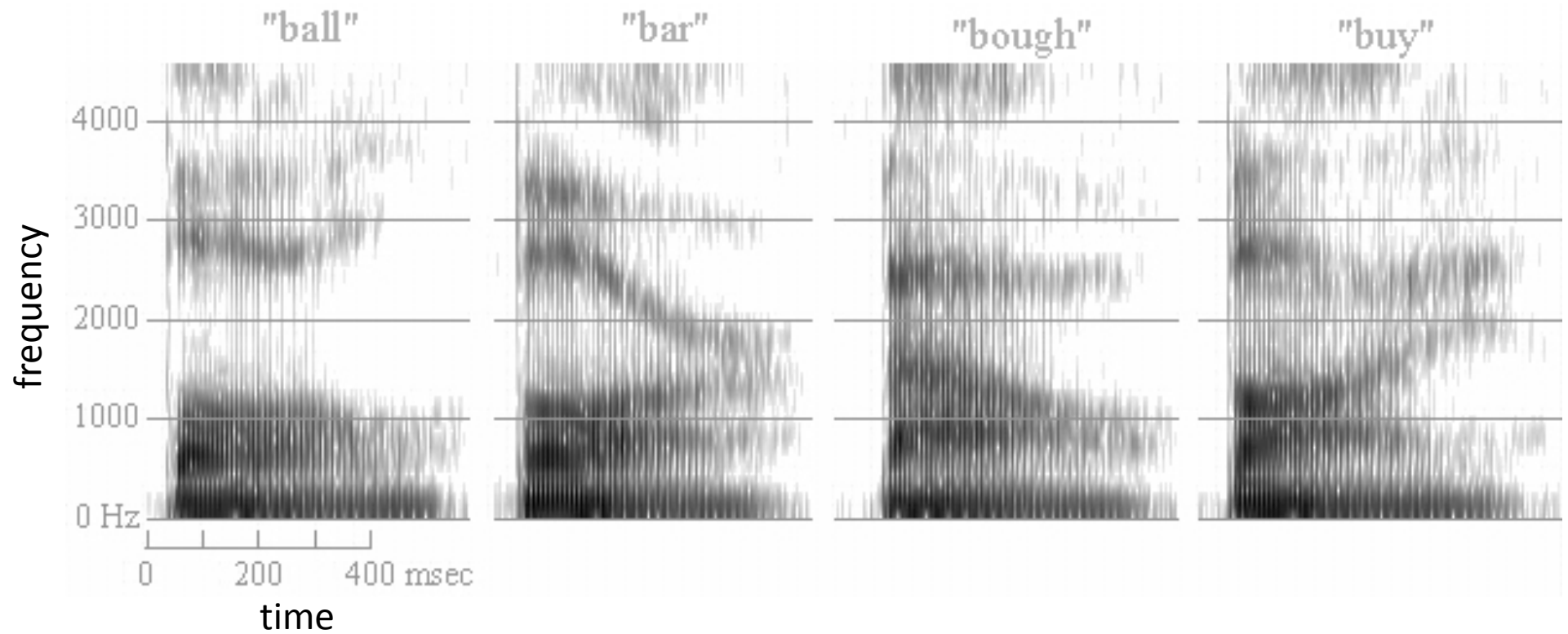


speech spectrograms

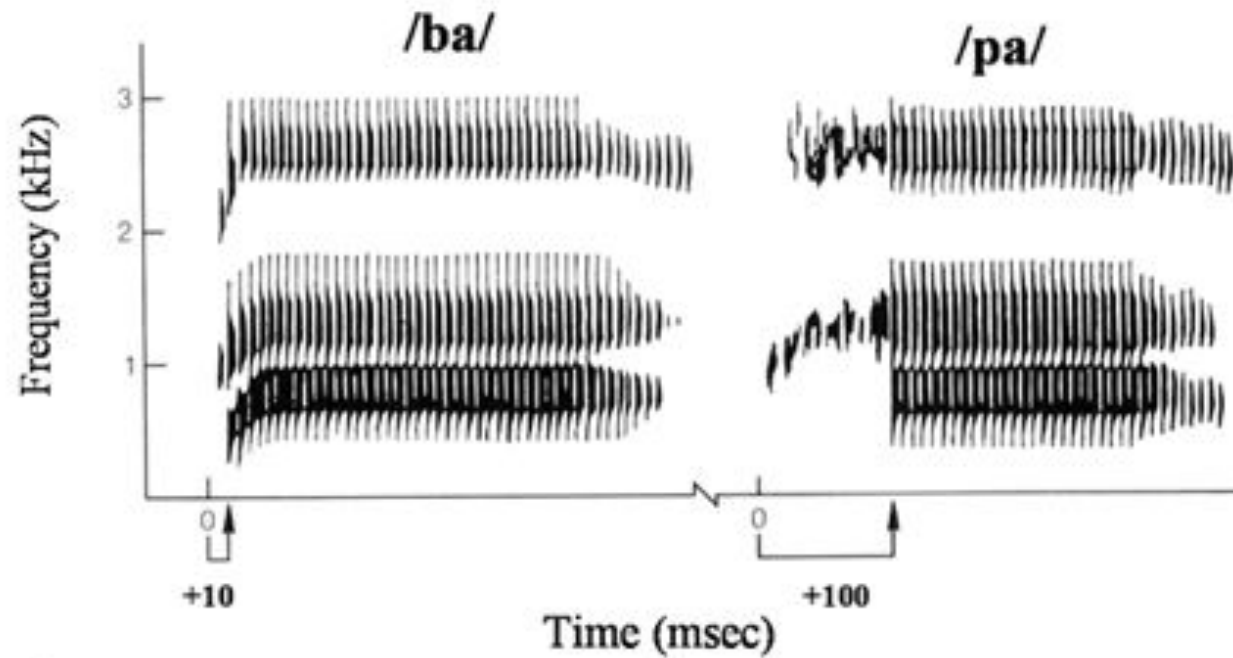
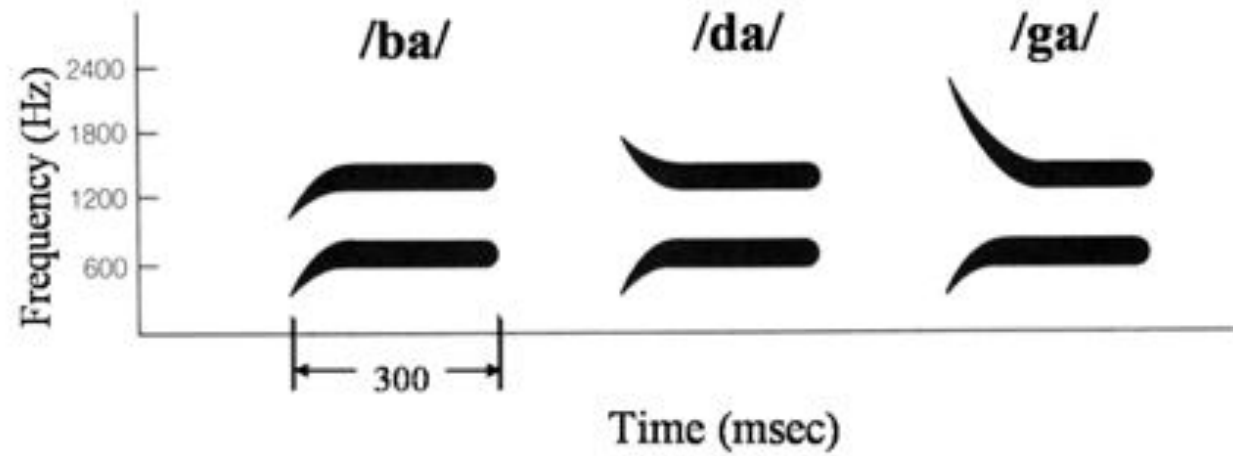
"I owe you a yo-yo"



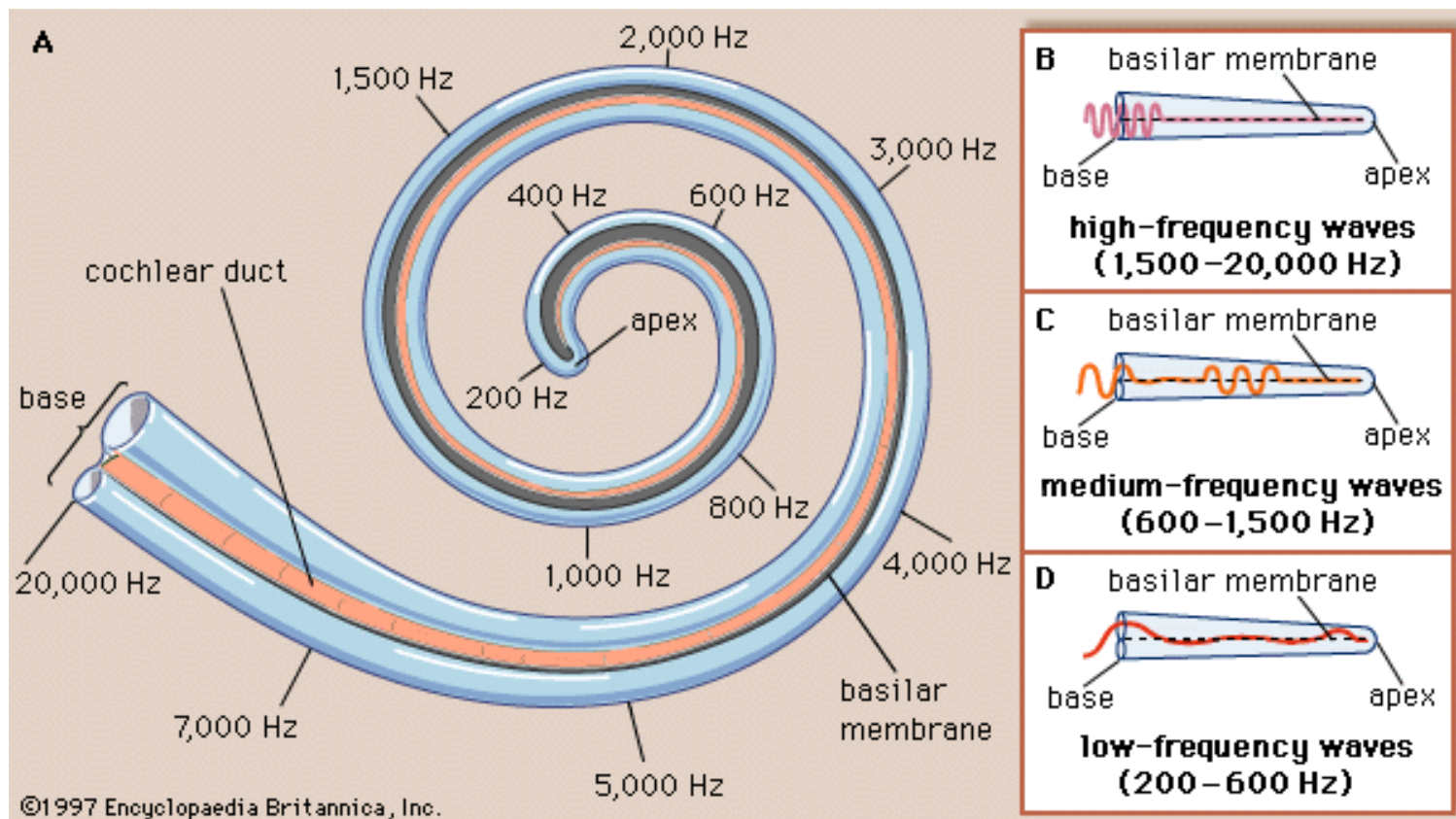
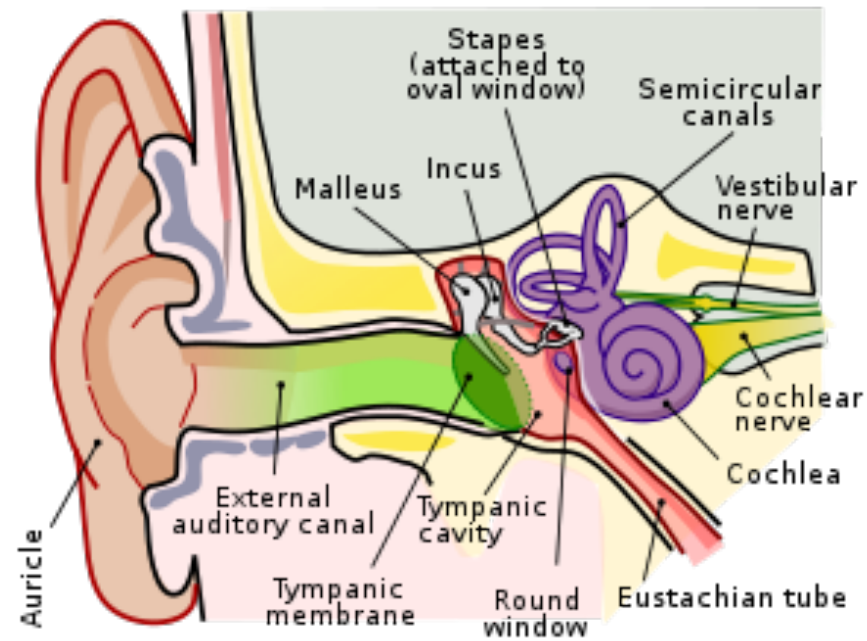
speech spectrograms



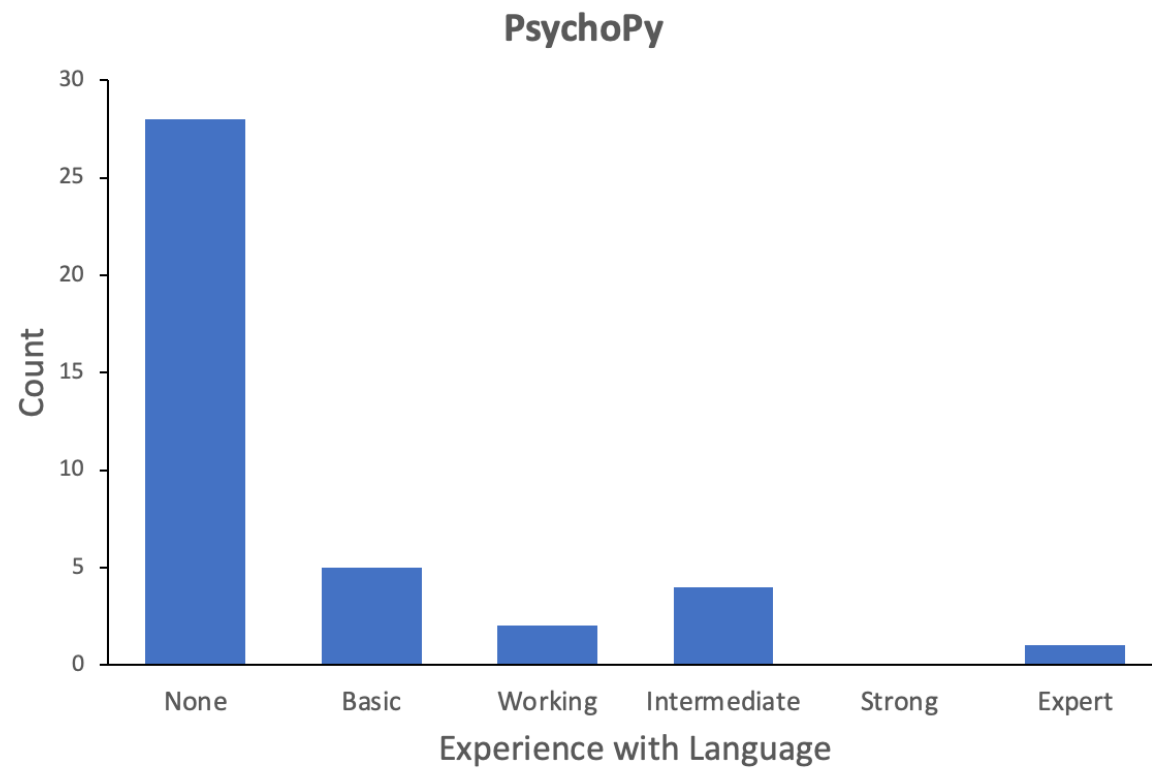
speech spectrograms



cochlea

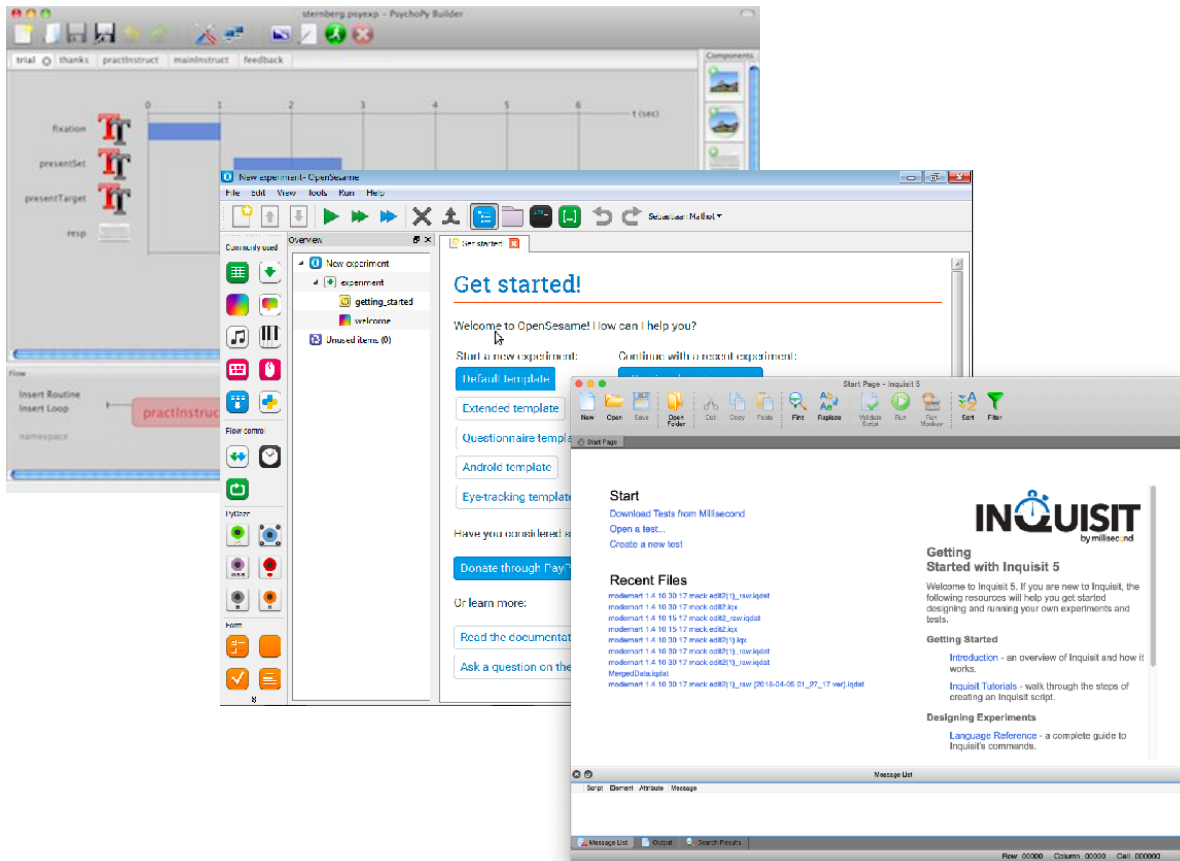
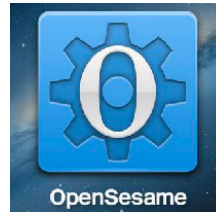


PsychoPy for Experiments



GUI experiment builders

Open Source / Free

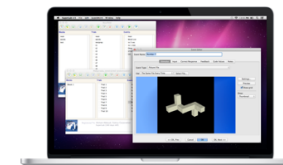


Not free

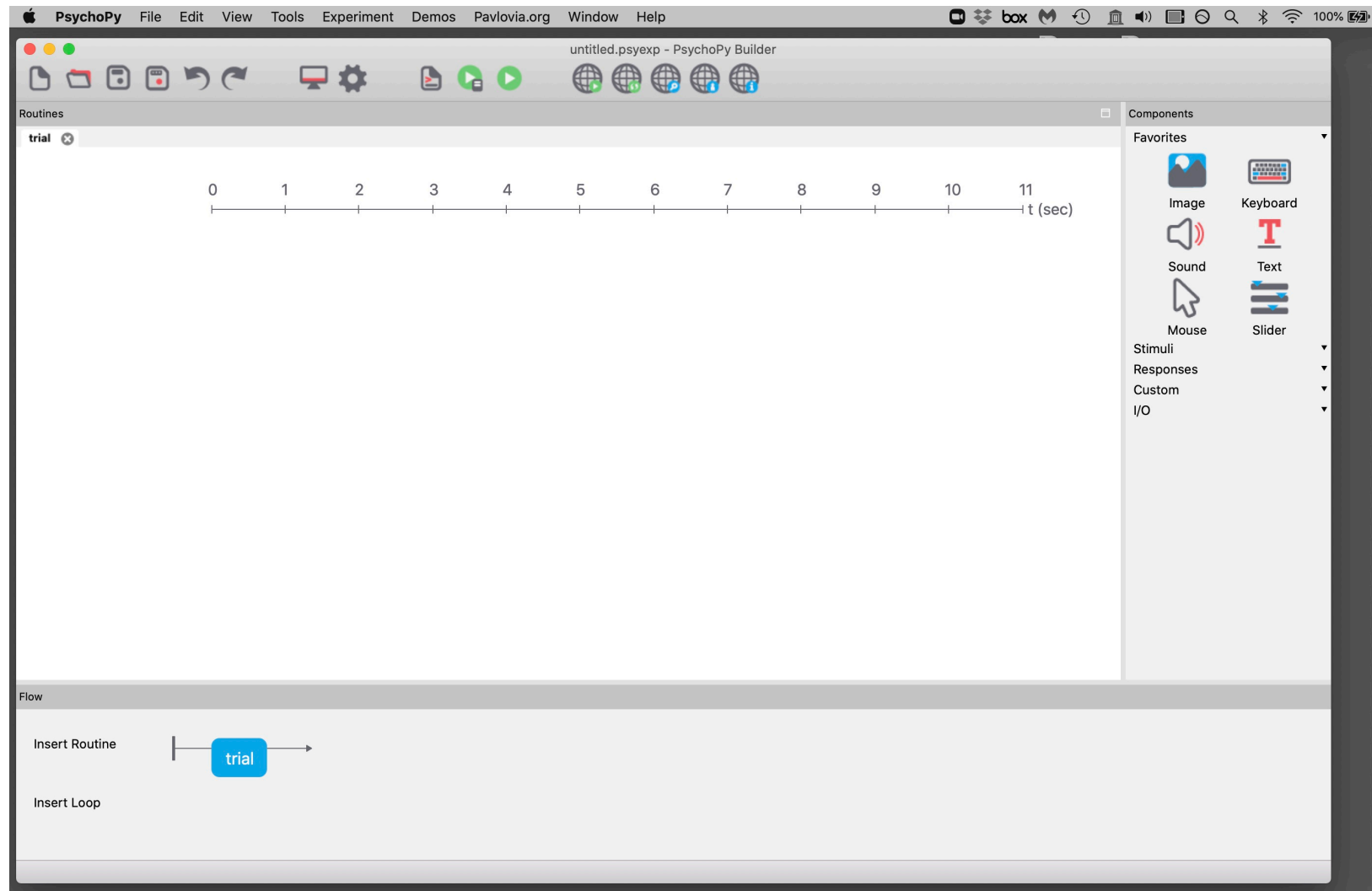


Presentation

SuperLab 5



PsychoPy



you can use the PsychoPy app to build experiments with a GUI,
but we are going to program them using the Python libraries

libraries/packages in programming languages

Python



Matlab

Psychtoolbox-3

Javascript



Creating Experiments



PsychoPy documentation

Basic Concepts and PsychoPy Tutorial

<https://www.psychopy.org/coder/>

PsychoPy API

<https://www.psychopy.org/api/>

PsychoPy Manual

<https://www.psychopy.org/PsychoPyManual.pdf>

PsychoPy Videos

<https://www.youtube.com/playlist?list=PLuqBA9VDSXk7Z06RtJ6Gh6Y5YznVrFrK6>

test1.py

PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],      size of window in pixels  
    screen=0,  
    fullscr=False,  
    allowGUI=False,  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

<https://www.psychopy.org/api/visual/window.html#psychopy.visual.Window>

PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,  
    allowGUI=False,  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

which monitor?
0 = main monitor
1 = external monitor

<https://www.psychopy.org/api/visual/window.html#psychopy.visual.Window>

PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,          full screen or not  
    allowGUI=False,  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

<https://www.psychopy.org/api/visual/window.html#psychopy.visual.Window>

PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,  
    allowGUI=False,      adds window interface  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

<https://www.psychopy.org/api/visual/window.html#psychopy.visual.Window>

PsychoPy example

recommendation: for development (and homework)

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,      set full screen to False  
    allowGUI=True,      can X out of the window  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

PsychoPy example

recommendation: for actual experiments

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=True,      set full screen to True  
    allowGUI=False,    do not allow X out  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,  
    allowGUI=False,  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

**testMonitor is a "starter"
can create monitor specs
in the PsychoPy app**

<https://www.psychopy.org/api/visual/window.html#psychopy.visual.Window>

PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,  
    allowGUI=False,  
    monitor='testMonitor',  
    units='deg',           possible units: 'norm',  
    color='gray' )        'cm', 'deg', 'pix', 'height'
```

<https://www.psychopy.org/general/units.html#units>

PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,  
    allowGUI=False,  
    monitor='testMonitor',  
    units='deg',  
    color='gray' ) set background color
```


PsychoPy example

psychopy has several modules with the package

```
from psychopy import visual, core, event
```

<https://www.psychopy.org/api/index.html>

Window object containing attributes and methods

creating a Window is a basic first step for an experiment

```
mywin = visual.Window(  
    size=[800,600],  
    screen=0,  
    fullscr=False,  
    allowGUI=False,  
    monitor='testMonitor',  
    units='deg',  
    color='gray' )
```

<https://www.psychopy.org/api/visual/window.html#psychopy.visual.Window>

PsychoPy example

core contains a variety of basic functions (simple timing, quit)

```
core.wait(1.0)
```

<https://www.psychopy.org/api/core.html>

```
mywin.close()
```

```
core.quit()
```

PsychoPy example

```
core.wait(1.0)
```

closes the window

```
mywin.close()
```

```
core.quit()
```

PsychoPy example

```
core.wait(1.0)
```

```
mywin.close()
```

```
core.quit()
```

does PsychoPy tidying up and quits Python

test2.py

another example

```
from psychopy import visual, core, event

mywin = visual.Window(size=[800,600], screen=0,
                      fullscr=False, allowGUI=True,
                      monitor='testMonitor', units='deg',
                      color='gray')
```

another example

```
grating = visual.GratingStim(win=mywin,  
    mask='circle', size=3, pos=[-4,0], sf=3)
```

an example function for creating a variety of stimuli

```
fixation = visual.GratingStim(win=mywin,  
    size=0.5, pos=[0,0], sf=0, rgb=-1)
```

another example

```
while True:
```

```
    grating.setPhase(0.05, '+' )
```

```
    grating.draw( )
```

```
    fixation.draw( )
```

```
    mywin.flip( )
```

**continuous stimulus updating
and keyboard polling**

```
    if len(event.getKeys( )) > 0:
```

```
        break
```

```
    event.clearEvents( )
```


another example

```
while True:
```

```
    grating.setPhase(0.05, '+' )    update the grating
```

```
    grating.draw()
```

```
    fixation.draw()
```

```
    mywin.flip()
```

```
    if len(event.getKeys()) > 0:
```

```
        break
```

```
    event.clearEvents()
```

another example

```
while True:
    grating.setPhase(0.05, '+' )
    grating.draw( )           draw grating to back buffer
    fixation.draw( )
    mywin.flip( )

    if len(event.getKeys( ) ) > 0:
        break
    event.clearEvents( )
```

another example

```
while True:
    grating.setPhase(0.05, '+' )
    grating.draw( )
    fixation.draw( )           draw fixation to back buffer
    mywin.flip( )

    if len(event.getKeys( )) > 0:
        break
    event.clearEvents( )
```

another example

```
while True:
    grating.setPhase(0.05, '+' )
    grating.draw( )
    fixation.draw( )
    mywin.flip( )           flip to make back buffer visible

    if len(event.getKeys( ) ) > 0:
        break
    event.clearEvents( )
```

another example

```
while True:
    grating.setPhase(0.05, '+' )
    grating.draw( )
    fixation.draw( )
    mywin.flip( )
                                check for keyboard press
    if len(event.getKeys( ) ) > 0:
        break
    event.clearEvents( )
```

another example

```
while True:
    grating.setPhase(0.05, '+' )
    grating.draw( )
    fixation.draw( )
    mywin.flip( )

    if len(event.getKeys( )) > 0:
        break
    event.clearEvents( )
        clear event buffer
```

another example

closes the window

```
mywin.close()
```

```
core.quit()
```

another example

```
mywin.close()
```

```
core.quit()
```

does PsychoPy tidying up and quits Python

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

PsychoPy - test7.py

test7

Project PsychoPy ~/Desktop/mss/Course/P6219 Scientific Computing for Psychology/Python/PsychoPy

- data
- images
- models
- notebooks
- ebbinghaus.py
- Jordingray.bmp
- main.py
- mondrian.py
- README.md
- requirements.txt
- test1.py
- test2.py
- test3.py
- test4.py
- test5.py
- test6.py
- test7.py
- Week11a.zip

External Libraries

Scratches and Consoles

```
7 from psychopy import gui, core
8
9
10 # create gui interface
11 subgui = gui.Dlg()
12 subgui.addField('Subject Number:')
13 subgui.addField('Session Number:')
14 subgui.addField('Condition:', choices=['Experimental', 'Control'])
15 subgui.addField('Condition:', choices=['Experimental', 'Control'])
16
17 # show the gui
18 ok = subgui.show()
19 print(ok)
20
21 while not ok:
22     ok = subgui.show()
23
24 # extract data from gui
25 print('Subject Number : ', subgui.data[0])
26 print('Session Number : ', subgui.data[1])
27
28 core.quit()
```

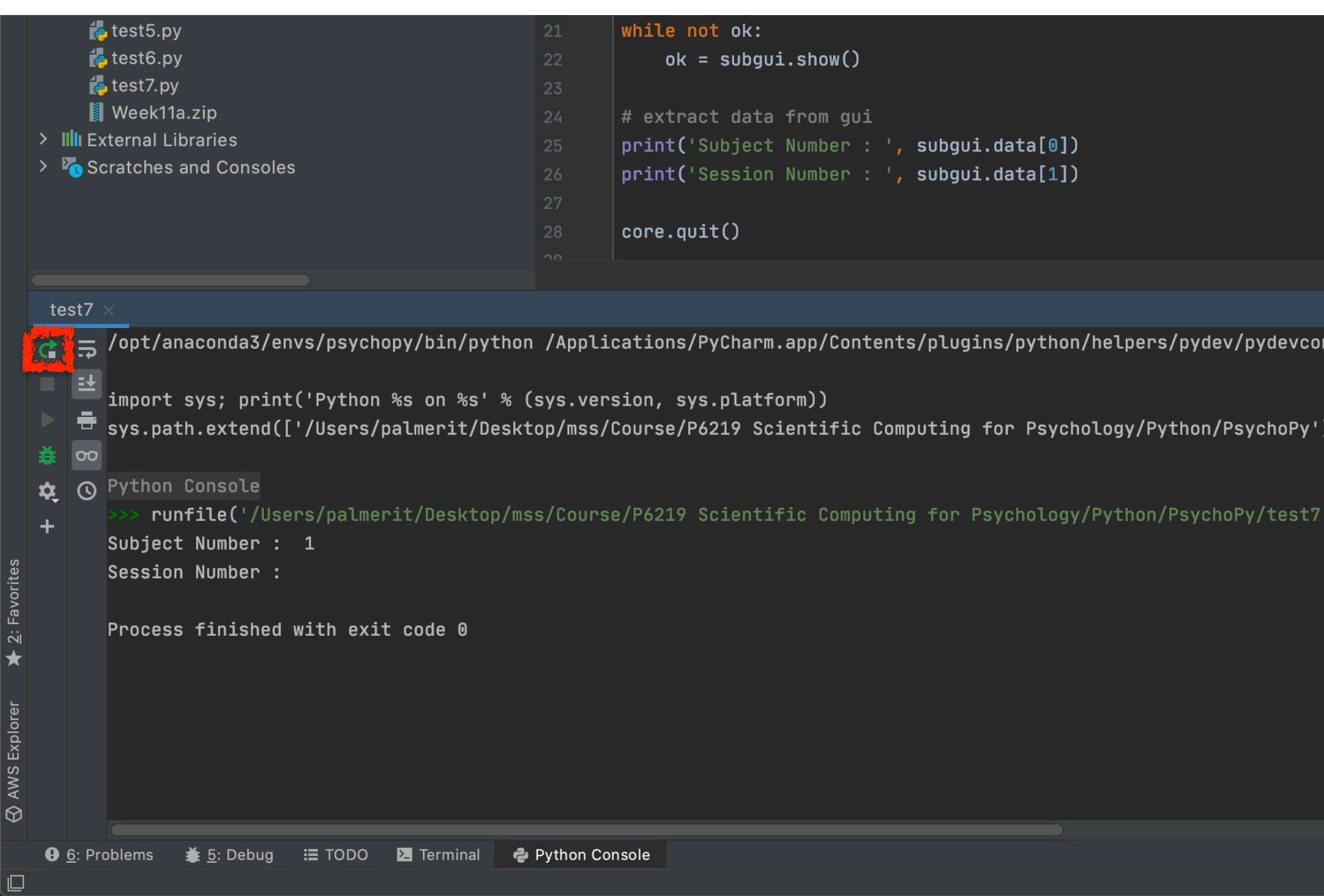
test7

```
/opt/anaconda3/envs/psychopy/bin/python /Applications/PyCharm.app/Contents/plugins/python/helpers/pydev/pydevconsole.py --mode=client --po
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/Users/palmerit/Desktop/mss/Course/P6219 Scientific Computing for Psychology/Python/PsychoPy'])
Python Console
>>> runfile('/Users/palmerit/Desktop/mss/Course/P6219 Scientific Computing for Psychology/Python/PsychoPy/test7.py', wdir='/Users/palmerit
Subject Number : 1
Session Number :

Process finished with exit code 0
```

Special Variables

1:1 LF UTF-8 AWS: No credentials selected 4 spaces Python 3.6 (psychopy)



Rerun does not create a new image