PsyMSc4 Kog - Praxismodul

Python for Psychologists

Rebecca Mayer
Dominik Kraft
WS 19/20

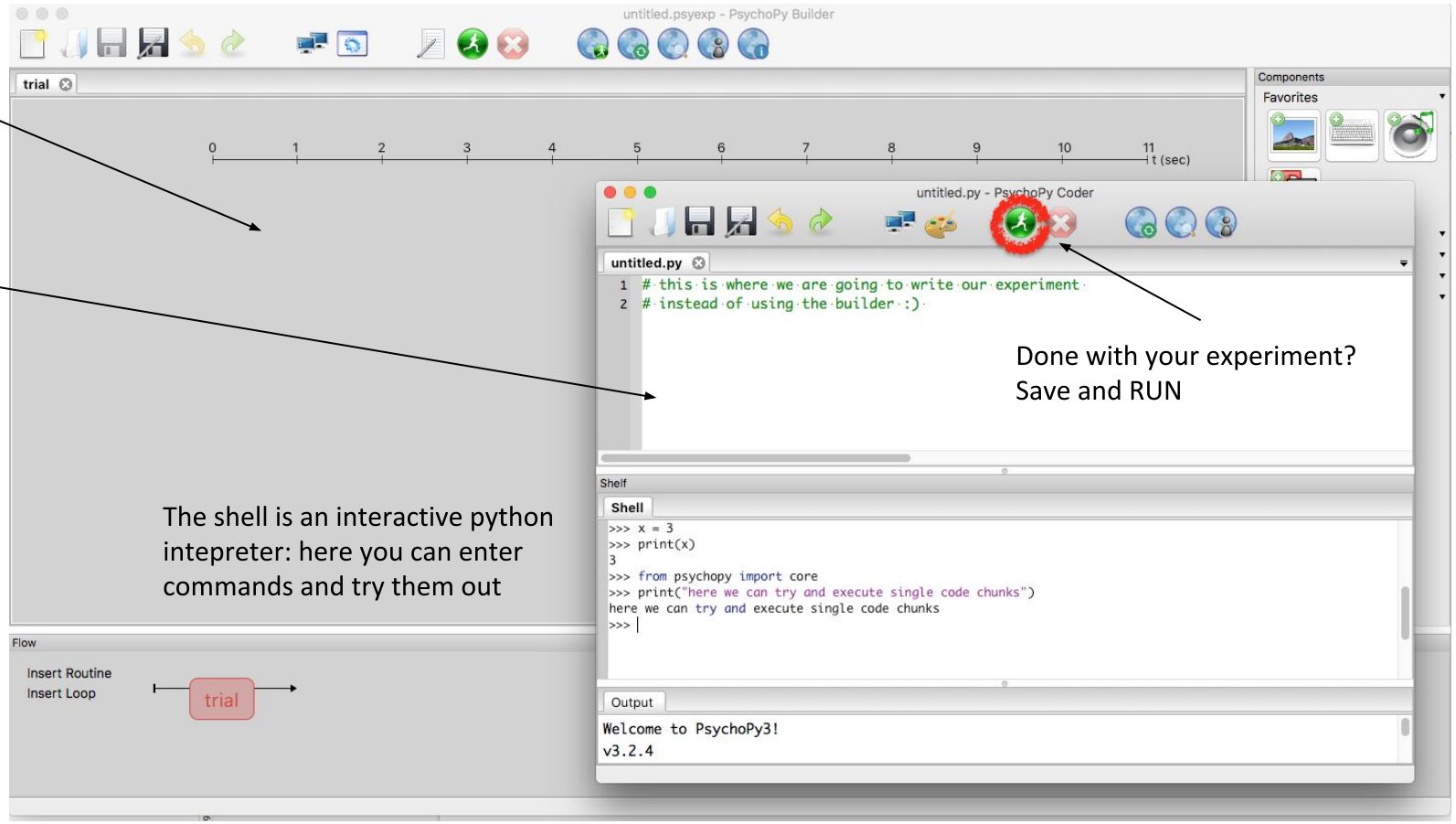
PsychoPy3

… a module for programming experiments

PsychoPy - Getting started I

Builder View

Coder View



PsychoPy - Where to get help

some useful resources:

- **PsychoPy Manual** https://www.psychopy.org/PsychoPyManual.pdf
- **API Reference Manual** https://www.psychopy.org/api/api.html
  - Contents:
    - `psychopy.core` - basic functions (clocks etc.)
    - `psychopy.visual` - many visual stimuli
    - `psychopy.clock` - Clocks and timers
    - `psychopy.data` - functions for storing/saving/analysing data

Experiment Header

```
>> #!/usr/bin/env python
```
→ tells your OS that this programme is using the python language

```
>> # -*- coding: utf-8 -*-
```
→ character encoding

```
# Import the PsychoPy libraries that you want to use
>> from psychopy import core, visual, event
```

Presenting stimuli

psychopy.**visual.window**(size=[800,600], pos=None, color=(0,0,0), colorSpace='rgb', ...)

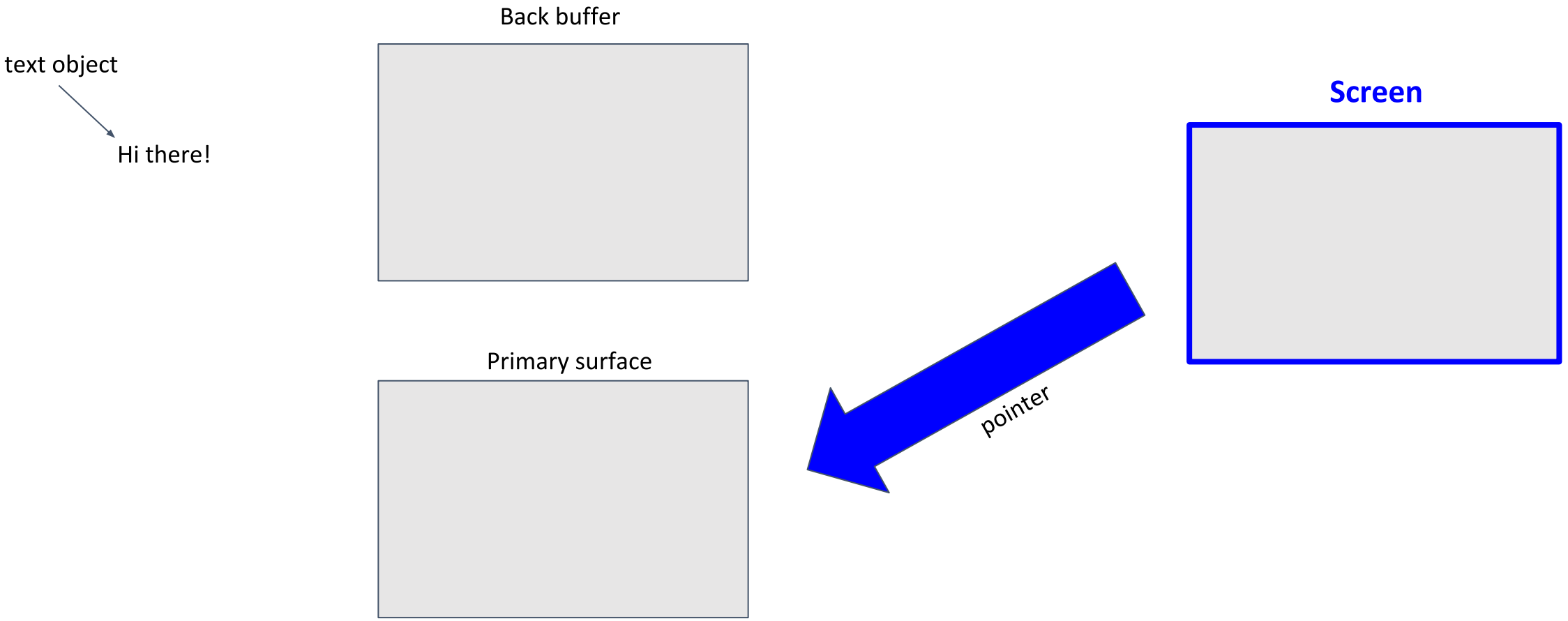
- creates a window in which stimuli etc. can be presented
- important method: .flip()
  - .flip() returns the time of presentation

Presenting stimuli

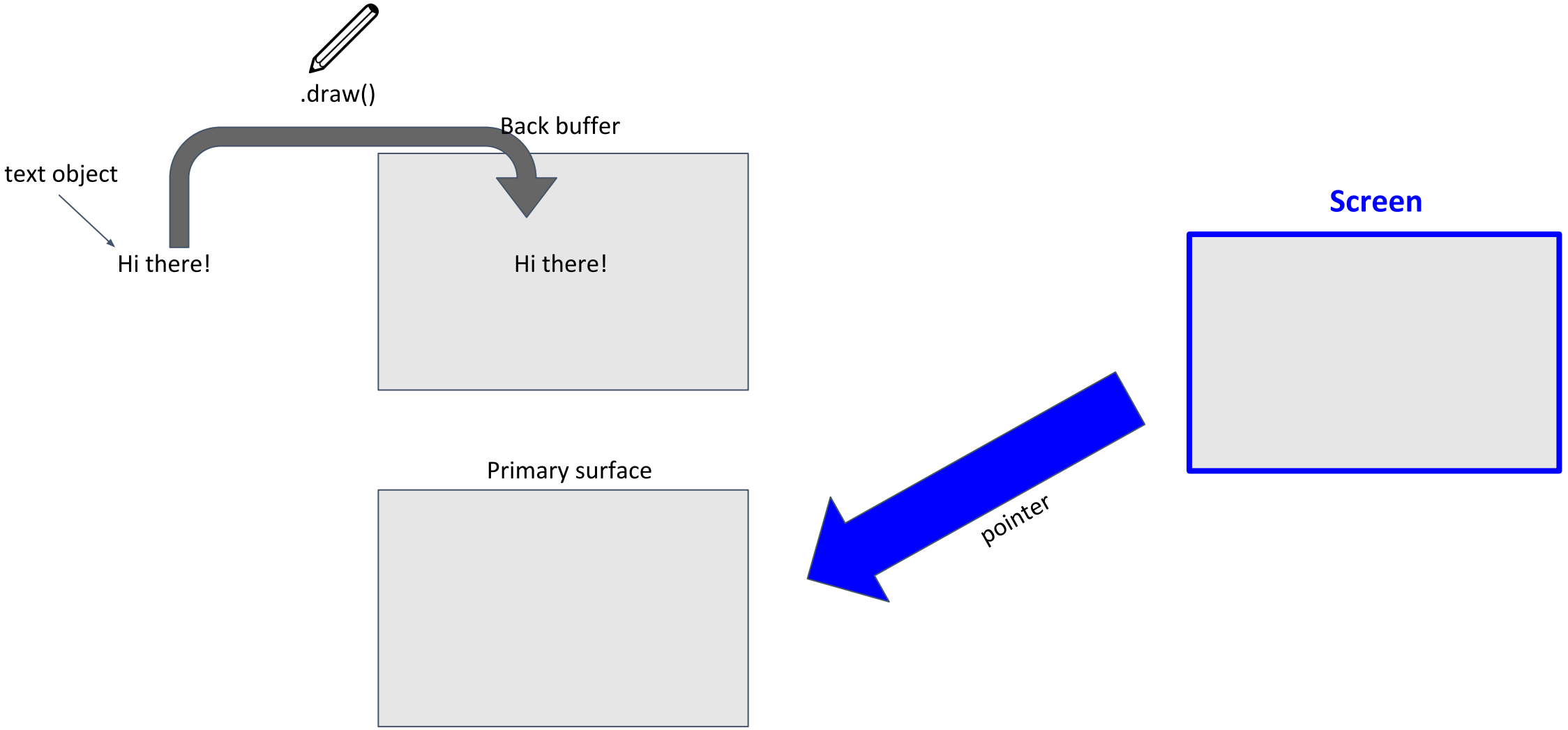
psychopy.**visual.TextStim**(window, text=”Hello world”, color=(0,0,0), colorSpace=’rgb’, pos=(0,0), ...)

- creates a text object
- important method: .draw()

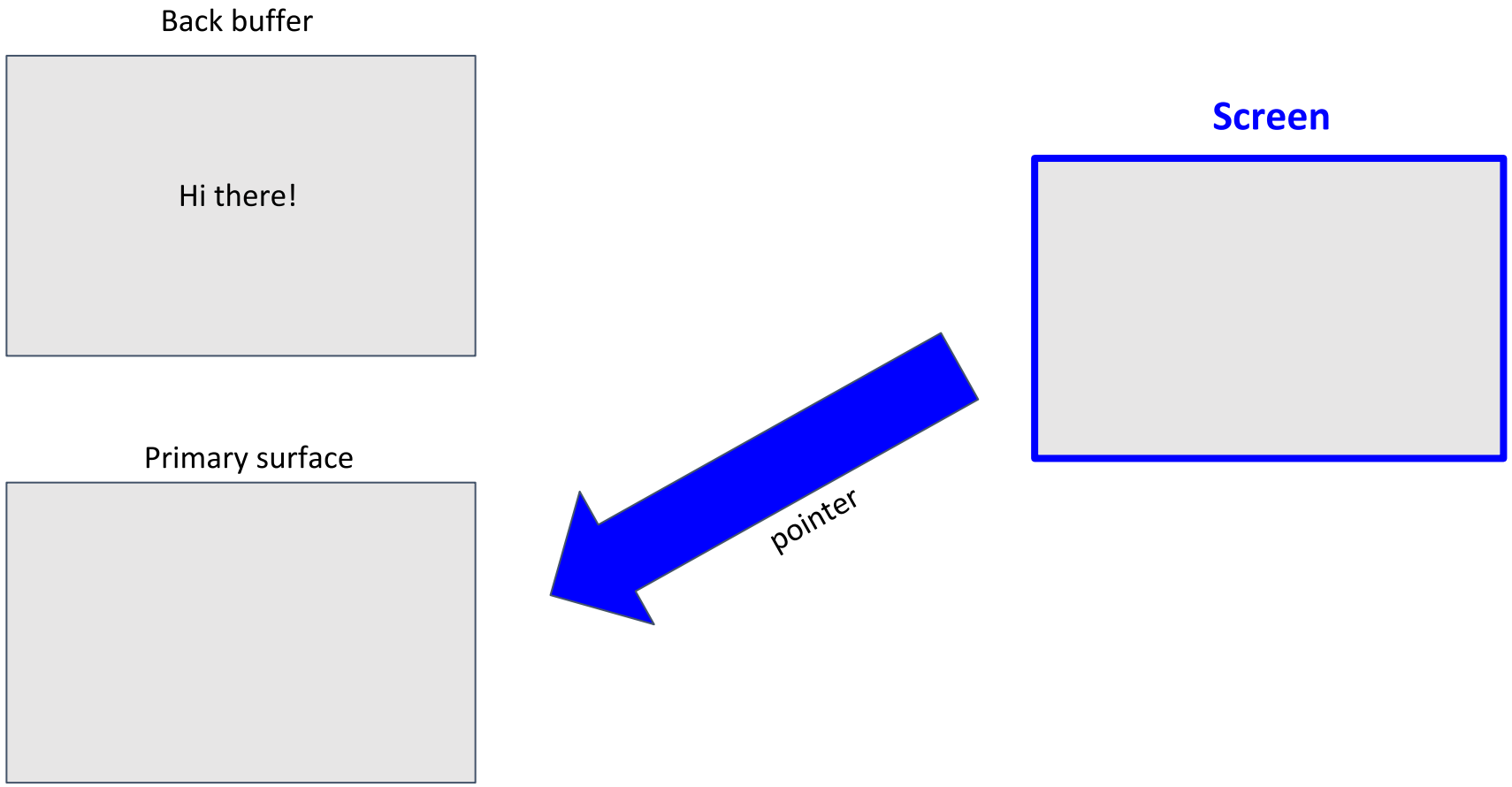
Presenting stimuli



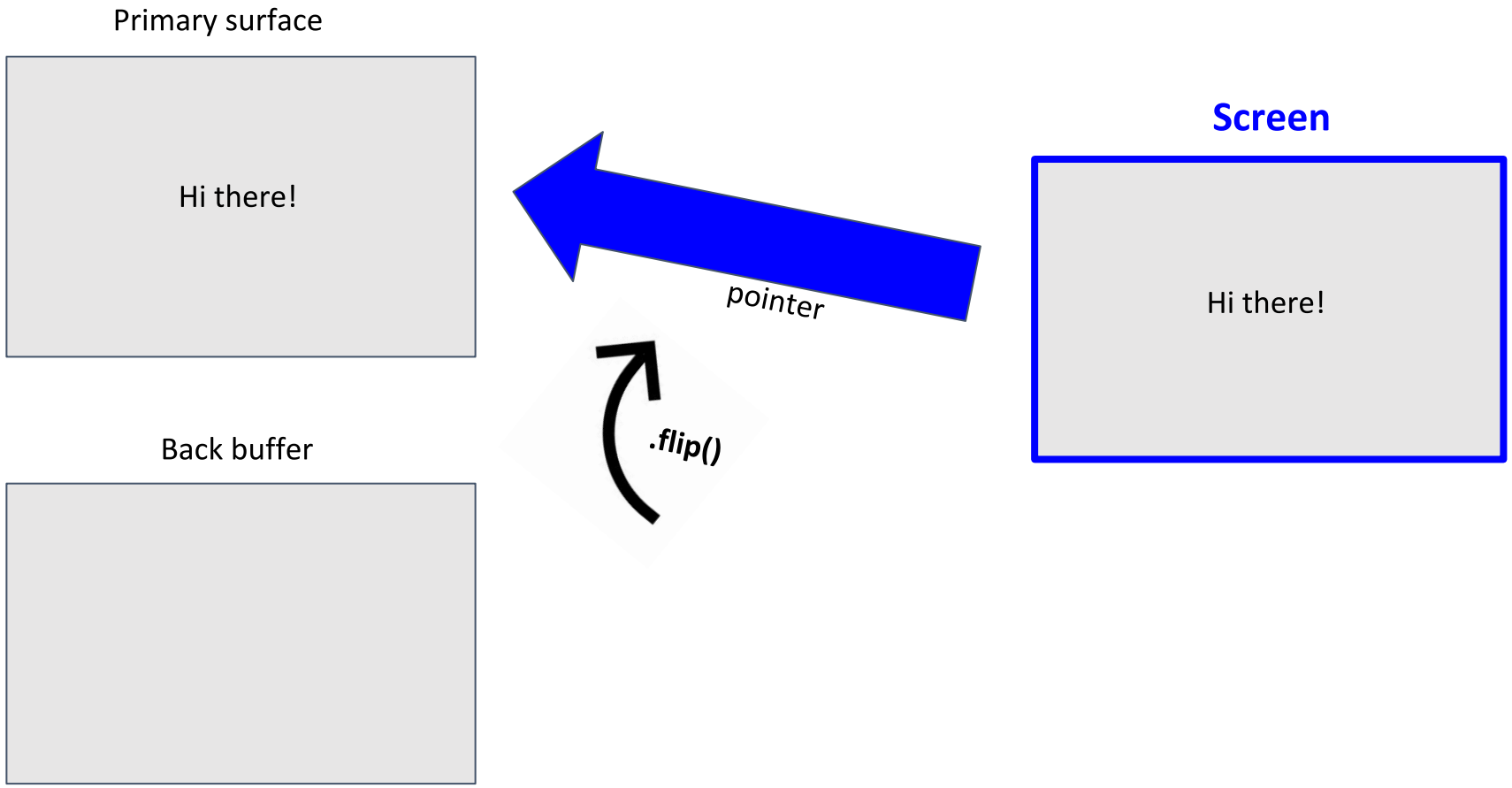
Presenting stimuli



Presenting stimuli



Presenting stimuli



Example

```
>> from psychopy import core, visual
# Create a window
>> win = visual.Window([400,300], monitor="testMonitor")
# Create a text stimulus for a certain window
>> message = visual.TextStim(win, text="Hi there!")
# Draw the stimulus to the window.
>> message.draw()
# Flip backside of the window. Puts picture immediately on the screen (but only for a fraction of time)
>> win.flip()
# Pause 5 s, so you get a chance to see it!
>> core.wait(5.0)
# Finally, close the window
>> win.close()
```

Timing stimuli

```
# tells psychopy to wait t seconds, before doing anything else
>> core.wait(t)

# initialize a clock for your experiment
>> clocki = core.clock()

# returning the current time of your clock
>> clocki.getTime()

# reset your clock (we named it clocki)
>> clocki.reset()
```

Recording responses

psychopy.**event.waitKeys**(maxWait=inf, keyList=None, timeStamped=False, ...)

- waits maxWait seconds for either any key to be pressed or for one of the keys contained in keyList to be pressed (if specified)
  - (Moves on, if the key is pressed before maxWait is reached)
- timeStamped
  - False: only the key pressed is returned (as a list); if no key is pressed, None is returned
  - True: the key pressed is returned as well as the time (as a tuple) at which the key press occured; again, None is returned, if no key was pressed
  - Clock: give the name of your clock to have the clock time saved instead; again, None is returned if no key was pressed

Exercise

1) Import the libraries core, visual and event from the psychopy module
2) Create a window of 500 x 500. Also, close the window again at the end of your script.
3) Now, create a text stimulus (a letter of your choice) that is presented on the screen (use the default settings).
4) Use .waitKeys() and the maxWait argument to a) record the response and, thus b) show the stimulus for 3 seconds. For now, don't specify any keyList and set timeStamped to False. Also, save the key to a variable (fyi: the output of .waitKeys() is of length 1). Note: you should also cover the case of misses, otherwise you might get an error when trying to assign the output. Hint: If no key has been pressed you the return value is None.
5) Now, as a second step, also record the reaction time. As a first step, simply set timeStamped to True. Store both the key and the reaction time in separate variables. Note, that now the output of wait.Keys() is of length 2, so you will have to "offer" a tuple of two variables, which the output can be assigned to.
6) Finally, write a loop. In each iteration, have another letter presented to the screen (check the setText method). Also, store all keys and rts in a list.
7) Save the presentation times of your stimuli (that is, at what time they have been presented).

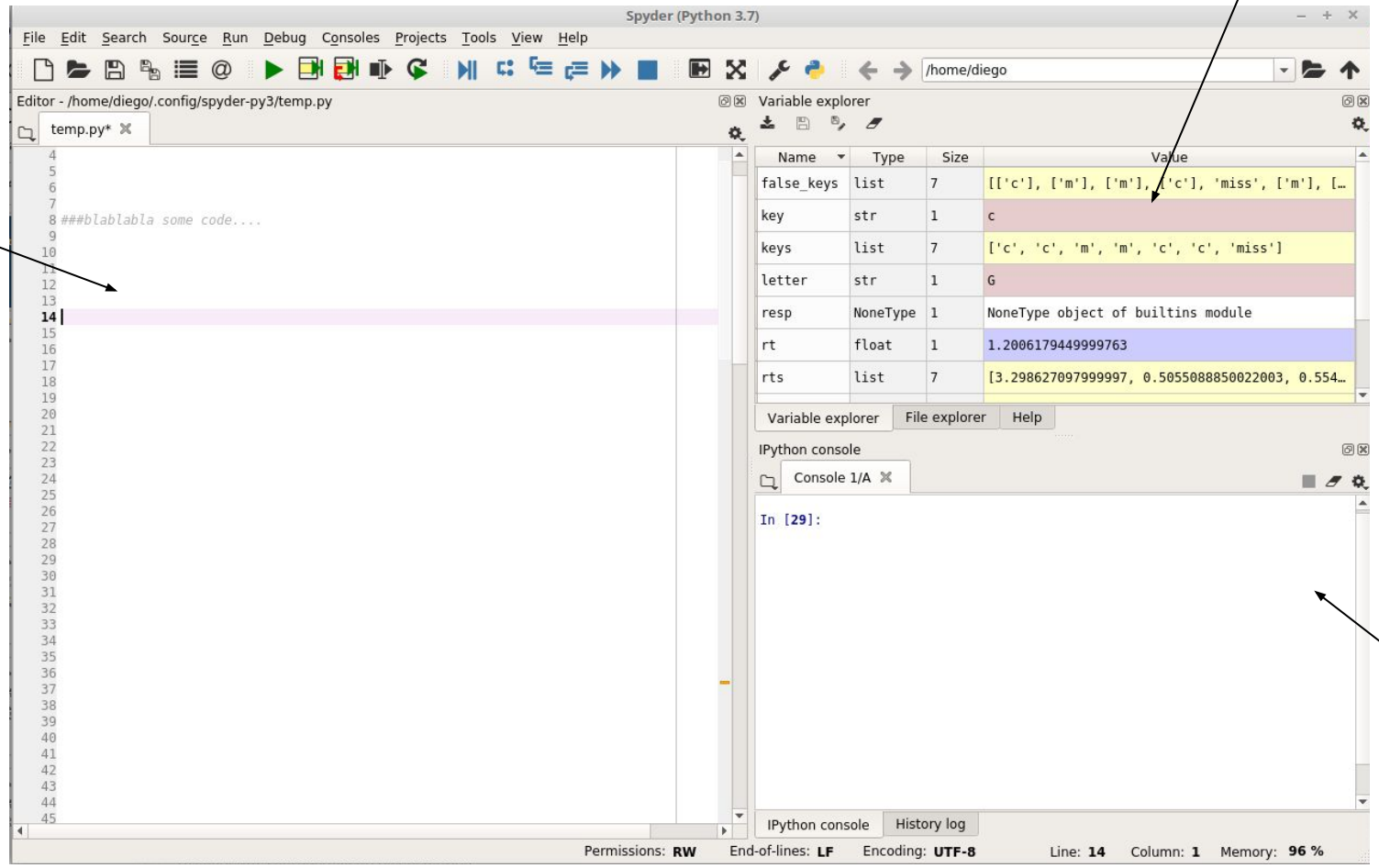
Timing stimuli

```
# tells psychopy to wait t seconds, before doing anything else
>> core.wait(t)

# initialize a clock for your experiment, your timer starts right away!
>> clocki = core.clock()

# returning the current time of your clock
>> clocki.getTime()

# reset your clock (we named it clocki)
>> clocki.reset()
```

The Spyder IDE

variable explorer/...

editor



console