

INTRODUCTION TO



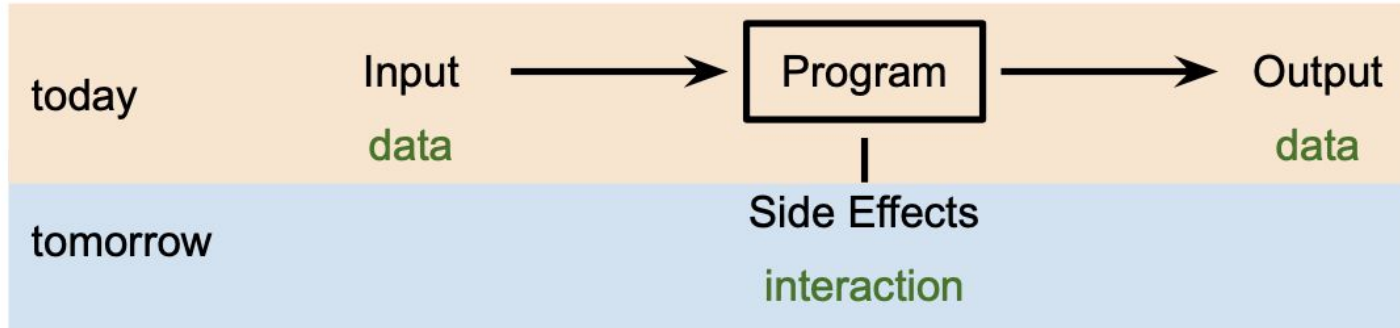
Aylin Kallmayer, Melvin Kallmayer, Leah Kumle
GRADE Workshop Frankfurt, May 11-12th 2021

YOUR TASK FOR TODAY

Program your own experiment in Python using Psychopy

Why?

YOUR TASK FOR TODAY

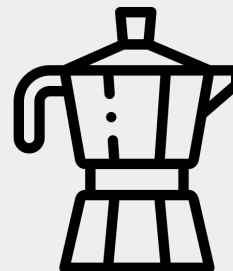
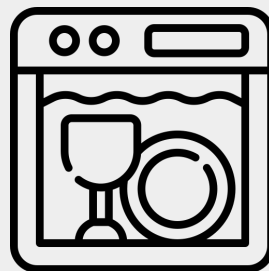




*Psychopy*³
Now running studies online

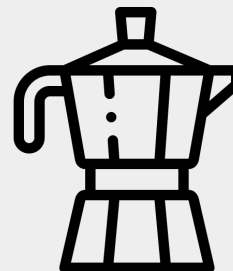
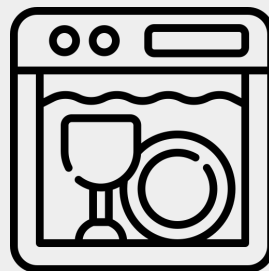
MNE
MEG + EEG ANALYSIS & VISUALIZATION

PYTORCH





PYTORCH





“PsychoPy is a free cross-platform package allowing you to run a wide range of **experiments** in the behavioral sciences (neuroscience, psychology, psychophysics, linguistics...)

This is a **community project**. Users have all the source code. Users are the developers. Users support each other.”



PsychoPy³

Now running studies online

Is this a cat?

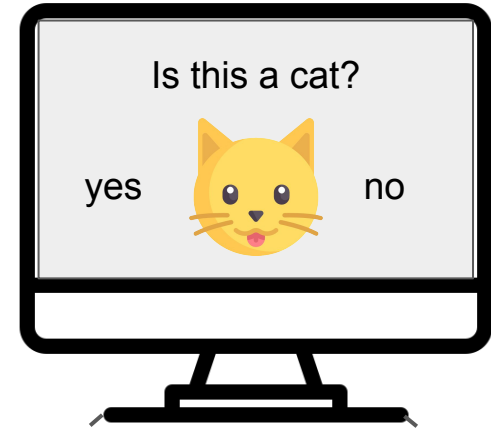
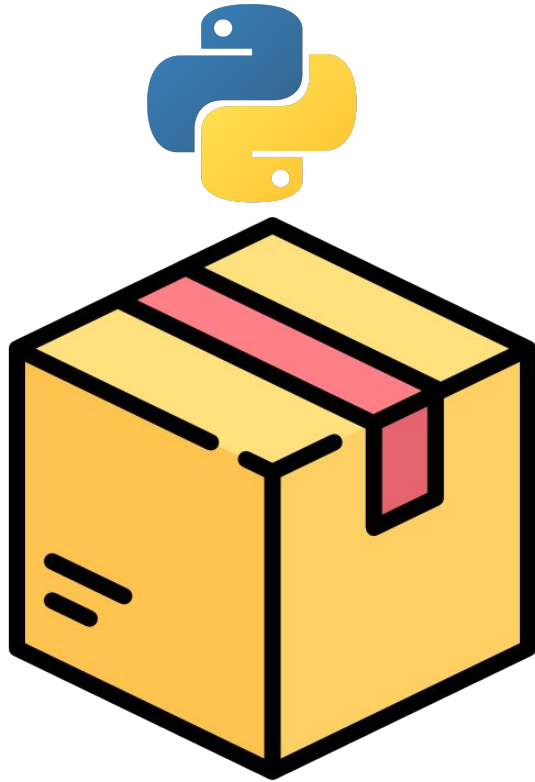


Press “y” for yes and
“n” for no



Time

PSYCHOPY



```
1 from psychopy import visual, event, core, logging, sound, monitors, gui, data
2
3 # create a window to draw in
4 win = visual.Window((800, 800), fullscr=False, monitor='testMonitor',
5 allowGUI=True, color='grey', units='pix')
6
7 visual.TextStim(win, "Success! Press space to quit", pos=(0,0)).draw()
8 win.flip()
9 event.waitKeys()
10 win.flip()
11
12 core.quit()
```

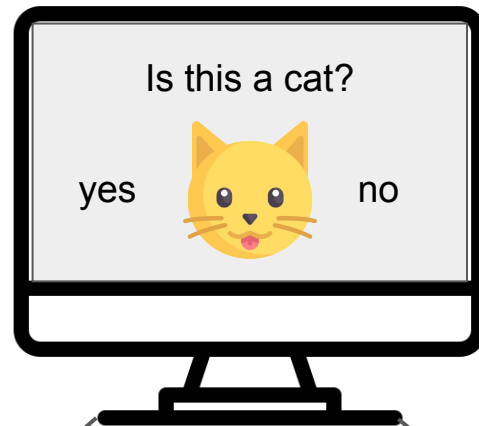


PSYCHOPY

In [2]: `# THIS IS A COMMENT
variable_1 = 5
variable_2 = 3

result = variable_1 + variable_2
print(result)`

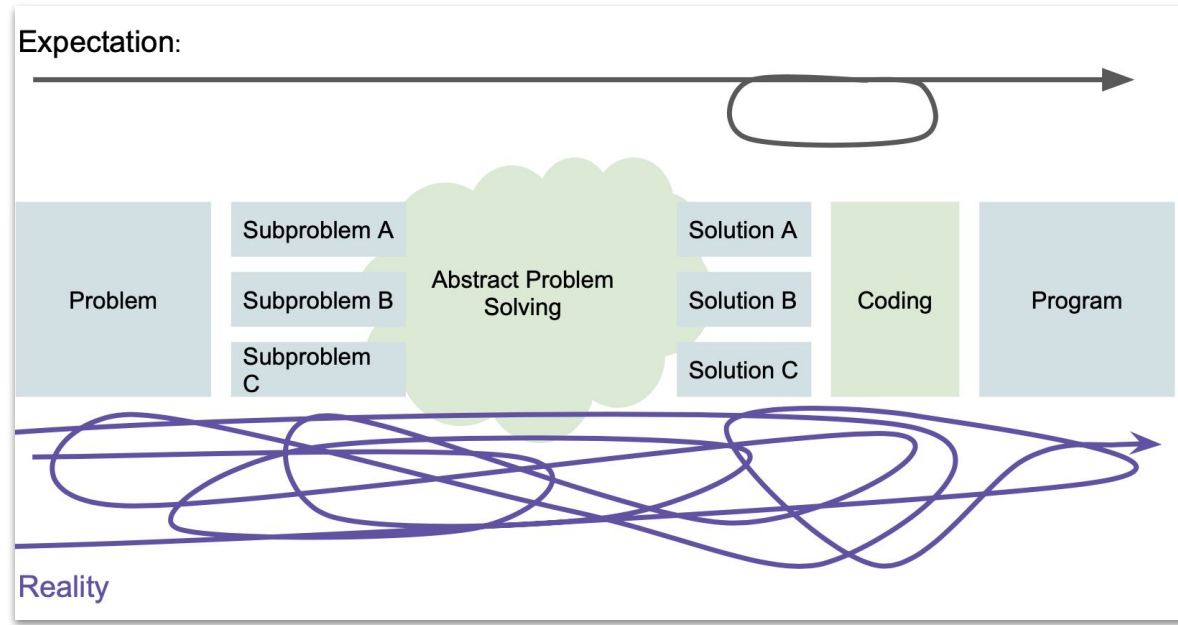
8



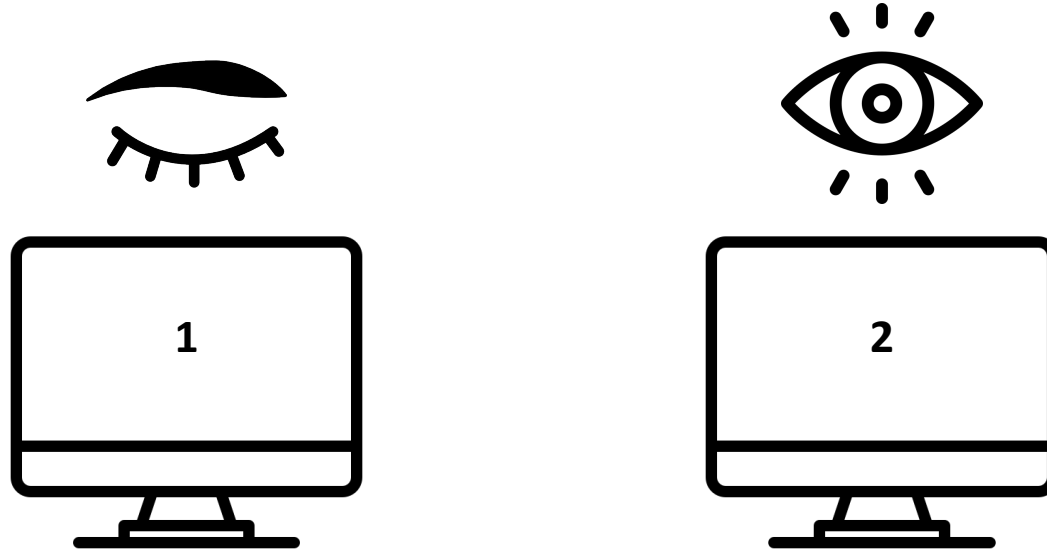
```
1 from psychopy import visual, event, core, logging, sound, monitors, gui, data
2
3 # create a window to draw in
4 win = visual.Window((800, 800), fullscr=False, monitor='testMonitor',
5 allowGUI=True, color='grey', units='pix')
6
7 visual.TextStim(win, "Success! Press space to quit", pos=(0,0)).draw()
8 win.flip()
9 event.waitKeys()
10 win.flip()
11
12 core.quit()
```



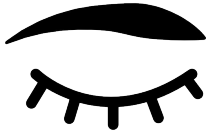
NO PAIN, NO GAIN (OR SOMETHING...)



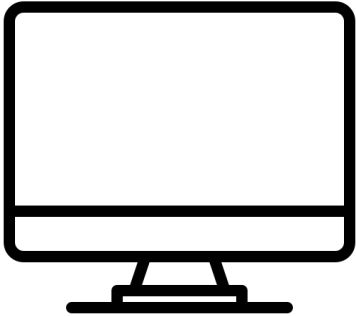
THE PSYCHOPY FLOW



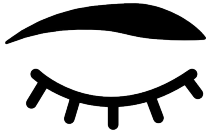
THE PSYCHOPY FLOW



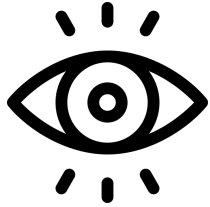
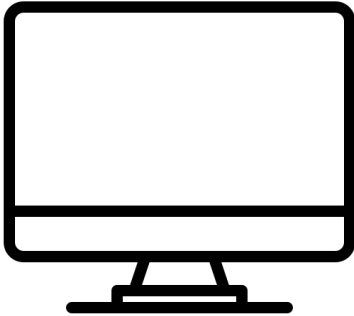
**Drawing on hidden
window**



THE PSYCHOPY FLOW



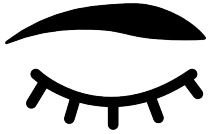
**Drawing on hidden
window**



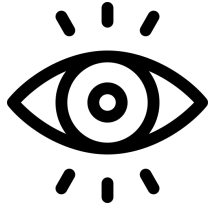
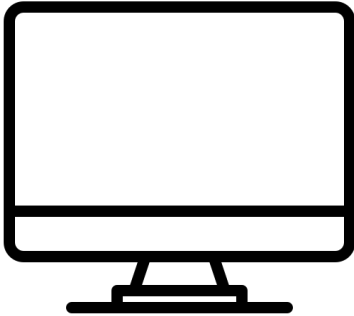
Flipping



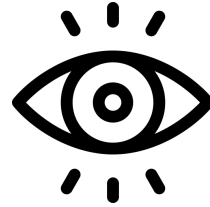
THE PSYCHOPY FLOW



**Drawing on hidden
window**



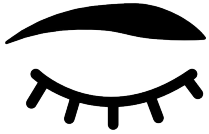
Flipping



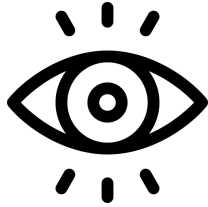
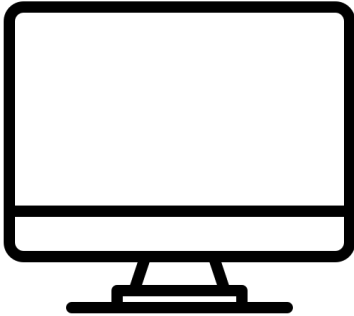
Waiting (for action)



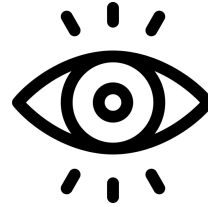
THE PSYCHOPY FLOW



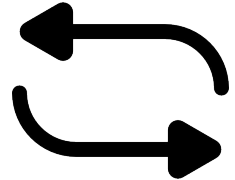
Drawing on hidden
window



Flipping

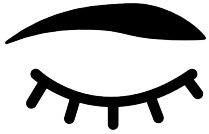


Waiting (for action)

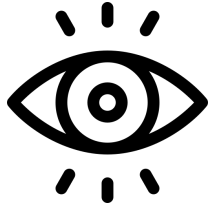
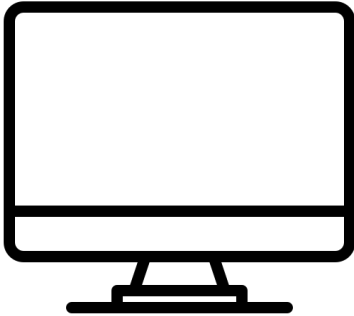


REPEAT!

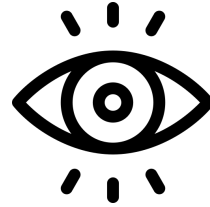
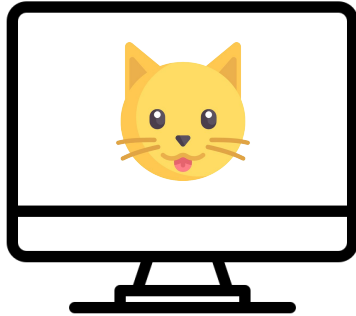
THE PSYCHOPY FLOW



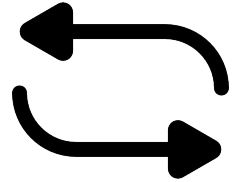
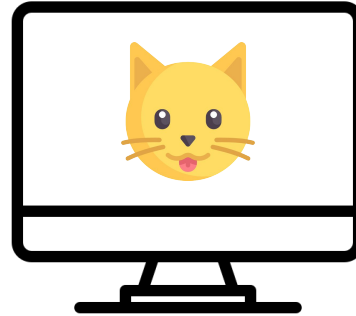
Drawing on hidden
window



Flipping



Waiting (for action)



REPEAT!



Part 1: Variables, Types, Lists, Operators

We will start with the very basics. You will see that with each section you complete, you can gather more detailed insight of our group.

A) Using Python as a calculator

Try converting the following expression into something that Python can understand: Your result should be ~ 1353.362985



```
In [ ]: 1 # enter equation here
        2
```

Comments: Have you noticed the text following the '#' sign? That is called a comment. Comments are invisible to Python and we use them to make notes for ourselves that make it easier to understand the code.

It is good practice to ALWAYS comment your code. Trust me, you will regret it later if you don't.

B) Assign your first variables

Store your age (as a numerical variable) and your name (as a string).

Useful: You can always use the 'print()' function to get Python to return a variable.

```
In [ ]: 1 my_name =
        2 my_age =
        3
```

Editor

gaze_cue_tut_part1.py

gaze_cue_tut_part2.py

gaze_cue_tut_part2_solutions.py

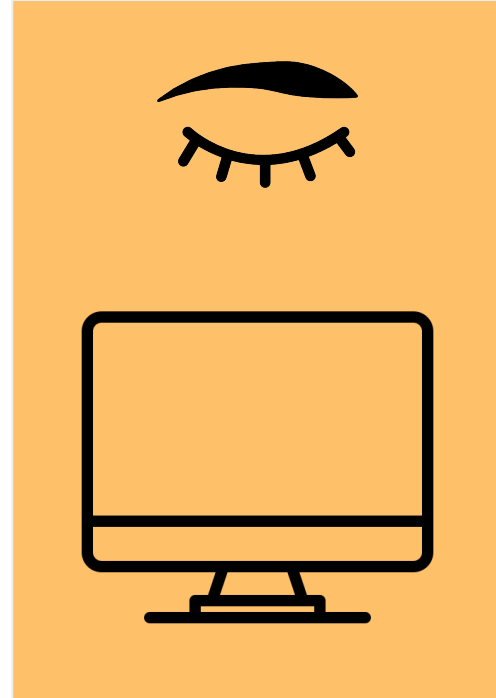
gaze_cue_demo.py



```
26
27 # Global event key to quit the experiment ("shutdown key").
28 event.globalKeys.add(key='q', func=core.quit)
29
30 # prepare output file
31 filename = "data.csv"
32 file = open(filename, "w", encoding='utf8')
33 writer = csv.writer(file, delimiter = ",")
34
35 # write the column names of the variables we want to log
36 writer.writerow(["subject_id", "gaze", "target", "position", "congruency", "response", "rt", "correct_response", "correct"])
37
38 # prepare dialogue box to collect participant name
39 win.fullscr = False
40 explInfo = {"Participant name": ""}
41 dlg = gui.DlgFromDict(explInfo, title='Gaze Cue Experiment')
42 if dlg.OK:
43     subject_id = explInfo["Participant name"]
44     win.fullscr = True
45 else:
46     core.quit() # the user hit cancel so exit
```

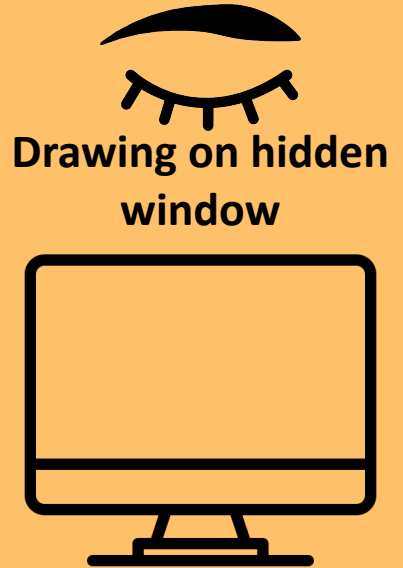
THE PSYCHOPY FLOW - IN CODE

```
from psychopy import visual  
  
win = visual.Window((800, 800))
```



THE PSYCHOPY FLOW - IN CODE

```
from psychopy import visual  
  
win = visual.Window((800, 800))  
  
visual.TextStim(win, "Hello World!").draw()
```



THE PSYCHOPY FLOW - IN CODE

```
from psychopy import visual  
  
win = visual.Window((800, 800))  
  
visual.TextStim(win, "Hello World!").draw()  
win.flip()
```



Flipping



THE PSYCHOPY FLOW - IN CODE

```
from psychopy import visual, core

win = visual.Window((800, 800))

visual.TextStim(win, "Hello World!").draw()
win.flip()

# option 1:
core.wait(1)
```

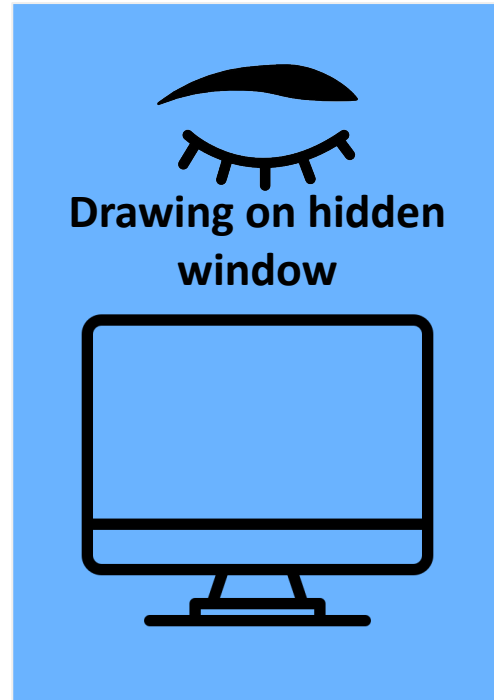


Flipping



THE PSYCHOPY FLOW - IN CODE

```
from psychopy import visual, core  
  
win = visual.Window((800, 800))  
  
visual.ImageStim(win, "cat.png").draw()
```

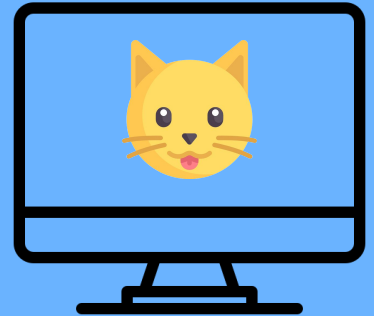


THE PSYCHOPY FLOW - IN CODE

```
from psychopy import visual, core  
  
win = visual.Window((800, 800))  
  
visual.ImageStim(win, "cat.png").draw()  
win.flip()
```



Flipping



THE PSYCHOPY FLOW - IN CODE

```
from psychopy import visual, core, event

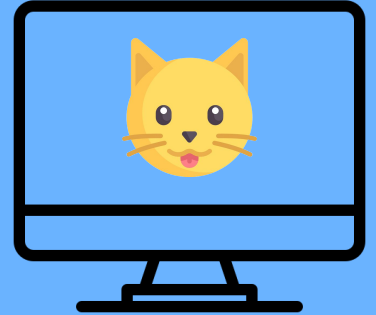
win = visual.Window((800, 800))

visual.ImageStim(win, "cat.png").draw()
win.flip()

# option 2:
event.waitKeys()
```



Waiting (for action)



THE PSYCHOPY FLOW

```
1  from psychopy import visual, core, event
2
3  win = visual.Window((800, 800), pos=(0,0))
4
5  visual.TextStim(win, "Hello world!").draw()
6  win.flip()
7  core.wait(1)
8  visual.ImageStim(win, "cat.png").draw()
9  win.flip()
10 event.waitKeys()
```

THE PSYCHOPY FLOW

```
1 from psychopy import visual, core, event
2
3 win = visual.Window((800, 800), pos=(0,0))
4
5 visual.TextStim(win, "Hello world!").draw()
6 win.flip()
7 core.wait(1)
8 visual.ImageStim(win, "cat.png").draw()
9 win.flip()
10 event.waitKeys()
```

Drawing on hidden
window



THE PSYCHOPY FLOW

```
1 from psychopy import visual, core, event
2
3 win = visual.Window((800, 800), pos=(0,0))
4
5 visual.TextStim(win, "Hello world!").draw()
6 win.flip()
7 core.wait(1)
8 visual.ImageStim(win, "cat.png").draw()
9 win.flip()
10 event.waitKeys()
```

Drawing on hidden
window



Flipping + waiting



THE PSYCHOPY FLOW

```
1 from psychopy import visual, core, event
2
3 win = visual.Window((800, 800), pos=(0,0))
4
5 visual.TextStim(win, "Hello world!").draw()
6 win.flip()
7 core.wait(1)
8 visual.ImageStim(win, "cat.png").draw()
9 win.flip()
10 event.waitKeys()
```

Drawing on hidden
window



Flipping + waiting



Drawing on hidden
window



THE PSYCHOPY FLOW

```
1 from psychopy import visual, core, event
2
3 win = visual.Window((800, 800), pos=(0,0))
4
5 visual.TextStim(win, "Hello world!").draw()
6 win.flip()
7 core.wait(1)
8 visual.ImageStim(win, "cat.png").draw()
9 win.flip()
10 event.waitKeys()
```

Drawing on hidden
window



Flipping + waiting



Drawing on hidden
window



Flipping + waiting



PSYCHOPY DOCUMENTATION

[Reference Manual \(API\) — PsychoPy v2021.1](#)

THE EXPERIMENT - GAZE CUE

Friesen and Kingstone (1998)

F



H



THE EXPERIMENT - GAZE CUE

Friesen and Kingstone (1998)

F



H



THE EXPERIMENT - GAZE CUE

Friesen and Kingstone (1998)

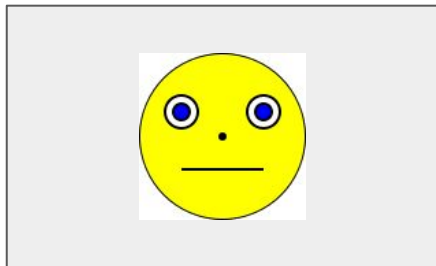
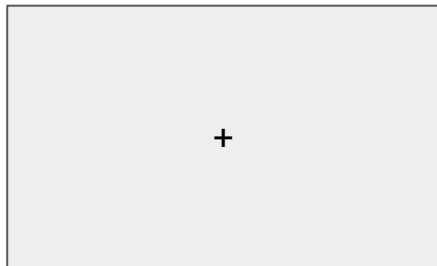
F



H



neutral gaze



THE EXPERIMENT - GAZE CUE

Friesen and Kingstone (1998)

F



H



neutral gaze



gaze cue
left / right



THE EXPERIMENT - GAZE CUE

Friesen and Kingstone (1998)

F



H



neutral gaze



gaze cue
left / right



response screen
gaze cue
target: F / H, left / right

F



X

THE EXPERIMENT - GAZE CUE

Friesen and Kingstone (1998)

neutral gaze



gaze cue
left / right



response screen
gaze cue
target: F / H, left / right

F



X

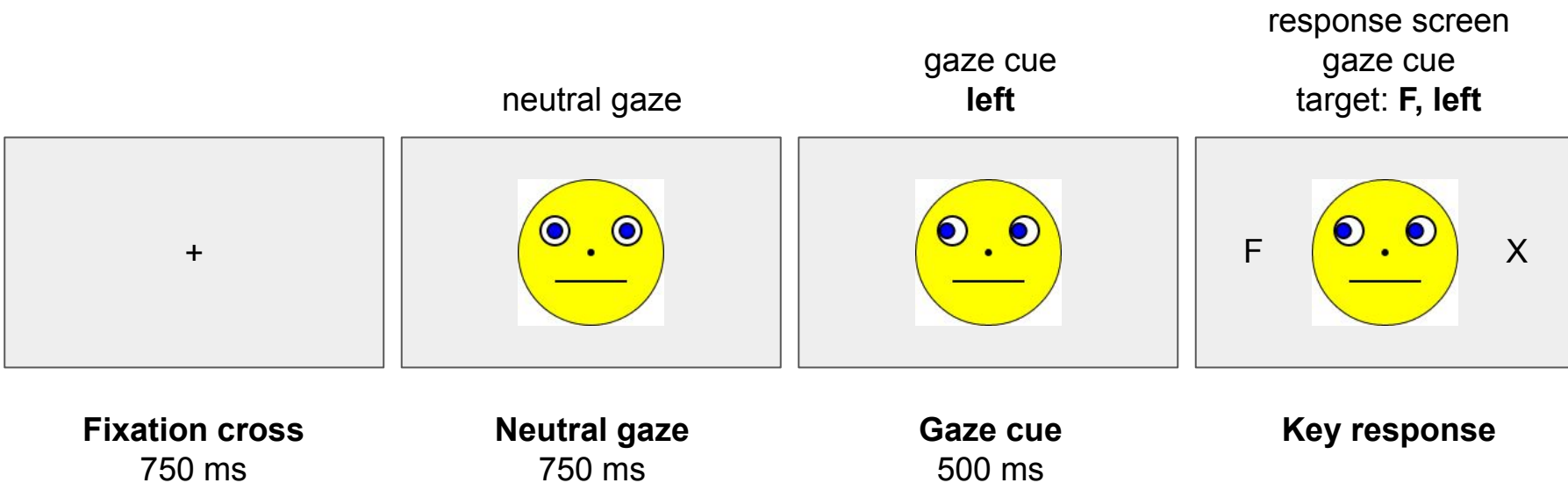
Fixation cross
750 ms

Neutral gaze
750 ms

Gaze cue
500 ms

Key response

PART 1



PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

Defining the function “saving to memory”

PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

Defining the function “saving to memory”

Calling the function “accessing from memory”

PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

```
def mysecondfunction(name):  
    # prints name and returns first  
    # letter of name  
  
    print(name)  
  
    return name[0]
```

PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

```
def mysecondfunction(name):  
    # prints name and returns first  
    # letter of name  
  
    print(name)  
  
    return name[0]
```

PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

```
def mysecondfunction(name):  
    # prints name and returns first  
    # letter of name  
  
    print(name)  
  
    return name[0]
```

PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

```
def mysecondfunction(name):  
    # prints name and returns first  
    # letter of name  
  
    print(name)  
  
    return name[0]
```

```
first_letter = mysecondfunction("aylin")  
> aylin
```

PART 1

```
def myfunction():  
    # prints "hello world"  
  
    print("hello world")
```

```
myfunction()  
> hello world
```

```
def mysecondfunction(name):  
    # prints name and returns first  
    # letter of name  
  
    print(name)
```

```
    return name[0]
```

```
first_letter = mysecondfunction("aylin")  
> aylin
```

```
print(first_letter)  
> "a"
```

Icon credits

All icons made by "<https://www.freepik.com>"

References

Friesen, C. K., & Kingstone, A. (1998). The eyes have it! Reflexive orienting is triggered by nonpredictive gaze. *Psychonomic Bulletin & Review*, 5, 490–495. [doi:10.3758/BF03208827](https://doi.org/10.3758/BF03208827)