

CogLab: jsPsych plugins

WEEK 4

recap: Sep 19, 2023

- what we covered:
 - jsPsych 101
- your to-dos were:
 - *read*: jsPsych webpage on plugins
 - *schedule*: project meeting with me to discuss ideas

today's agenda

- building a sequence of events/screens

going back to our experiment

- open Visual Studio Xcode and open the jsPsych experiment you created
- also open the `index.html` file in your browser to remind yourself of what we did!
- comment out `trial_duration` if you have that parameter (`//`)
- loading external content
 - **src**: images & scripts
 - **href**: stylesheets
 - **rel**: relationship

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>My experiment</title>
5 <script src="https://unpkg.com/jspsych@7.3.3"></script>
6 <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.1.2"></script>
7 <link href="https://unpkg.com/jspsych@7.3.3/css/jspsych.css" rel="stylesheet" type="text/css" />
8 </head>
9 <body></body>
10 <script>
11   const jsPsych = initJsPsych();
12
13   const hello_trial = {
14     type: jsPsychHtmlKeyboardResponse,
15     stimulus: 'Hello world!',
16     choices: ['Y', 'N']
17   }
18
19   jsPsych.run([hello_trial]);
20 </script>
21 </html>
22
```

experiment recap

training

sentence

space

novel word?

<response>

association x 3

word

<response>

x 3

priming

+



prime



target



A / L

exercise: which plugin?

- in groups of 3, figure out which jsPsych plugin would be best suited for different parts of the experiment

experiment recap

training

sentence

space

novel word?

<response>

association x 3

word

<response>

x 3

priming

+



prime



target



A / L

step 1: sentence trial

- rename `hello_trial` with `sentence`
- change `const` to `var`
- replace stimulus with an actual sentence
- change `choices` to a space string
- run the sentence trial
- save and reload!

```
<script>
  const jsPsych = initJsPsych();

  var sentence = {
    type: jsPsychHtmlKeyboardResponse,
    stimulus: 'I want to see a foobly mipp.',
    choices: [' ']
  }

  jsPsych.run([sentence]);
</script>
</html>
```


step 2: attention trial

- we want to display a question and ask participants to type in a response
- we will use a new plugin called survey-text for this
- go to the plugin “install” section and load the plugin into your HTML file by adding the `<script>` tag inside `<head>`

Install 📄

Using the CDN-hosted JavaScript file:

```
<script src="https://unpkg.com/@jspsych/plugin-survey-text@1.1.2"></script>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>My experiment</title>
    <script src="https://unpkg.com/jspsych@7.3.3"></script>
    <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.1.2"></script>
    <link href="https://unpkg.com/jspsych@7.3.3/css/jspsych.css" rel="stylesheet" type="text/css" />
    <script src="https://unpkg.com/@jspsych/plugin-survey-text@1.1.2"></script>
  </head>
```

step 2: attention trial

- define a new variable called **attention** after the sentence trial
 - **type**: tells HTML to use the survey-text plugin
 - **questions**: stores what you would like to ask the participant to do
 - can store multiple questions of the same type
- add **attention** to the **run** sequence
- save and reload

```
<script>
  const jsPsych = initJsPsych();

  var sentence = {
    type: jsPsychHtmlKeyboardResponse,
    stimulus: 'I want to see a foobly mipp.',
    choices: [' ']
  }

  var attention = {
    type: jsPsychSurveyText,
    questions: [{prompt: "Type any ONE novel word from the previous sentence:"}],
  };

  jsPsych.run([sentence, attention]);
</script>
</html>
```

step 3: free association

- we want to display a particular word and ask participants to respond with the first word that comes to mind
- we can just reuse the survey-text plugin for this!
- define a new **association** trial and add it to the run sequence

```
<script>
  const jsPsych = initJsPsych();

  var sentence = {
    type: jsPsychHtmlKeyboardResponse,
    stimulus: 'I want to see a foobly mipp.',
    choices: [' ']
  }

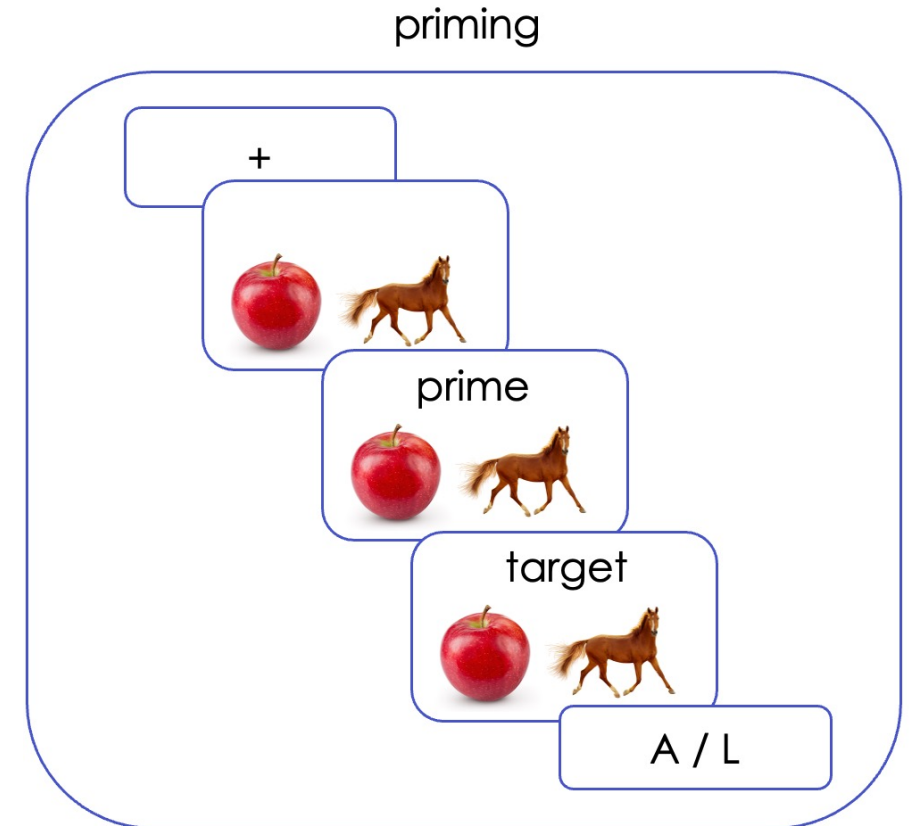
  var attention = {
    type: jsPsychSurveyText,
    questions: [{prompt: "Type any ONE novel word from the previous sentence:"}],
  };

  var association = {
    type: jsPsychSurveyText,
    questions: [{prompt: "foobly"}],
  };

  jsPsych.run([sentence, attention, association]);
</script>
</html>
```

step 4: priming

- priming has 4 separate components: a fixation cross, an image display, prime + image display, and a target + image display that leads to a response
- we will sequentially build each of these components



step 4a: fixation

- we want to display a fixation cross for a specified duration but allow no participant response
- we can use html-keyboard-response for this and limit choices to NO_KEYS
- define a **fixation** trial and add it to the run sequence
- save and reload

```
<script>
  const jsPsych = initJsPsych();

  var sentence = {
    type: jsPsychHtmlKeyboardResponse,
    stimulus: 'I want to see a foobly mipp.',
    choices: [' ']
  }

  var attention = {
    type: jsPsychSurveyText,
    questions: [{prompt: "Type any ONE novel word from the previous sentence:"}],
  };

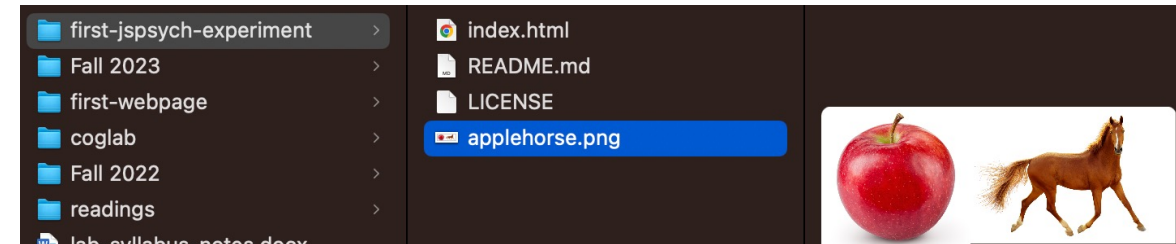
  var association = {
    type: jsPsychSurveyText,
    questions: [{prompt: "foobly"}],
  };

  var fixation = {
    type: jsPsychHtmlKeyboardResponse,
    stimulus: "+",
    choices: "NO_KEYS",
    trial_duration: 500,
  };

  jsPsych.run([sentence, attention, association, fixation]);
</script>
</html>
```

step 4b.1: image

- next, we need to display an image of a horse and an apple and not allow any responses
- we can use the image-keyboard-response plugin and set choices to NO_KEYS as before!
- first, we download and save the images `applehorse.png` and `horseapple.png` in our `first_jspsych_experiment` folder



step 4b.2: image

- next, we **load** the html-image-keyboard response plugin via `<script>` into the `<head>` tag
- finally, define an **image** trial
 - stimulus: name of the image file with the extension
 - choices: allow no response
 - duration: 500 ms
- save and reload

```
<html>
  <head>
    <title>My experiment</title>
    <script src="https://unpkg.com/jspsych@7.3.3"></script>
    <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.1.2"></script>
    <link href="https://unpkg.com/jspsych@7.3.3/css/jspsych.css" rel="stylesheet" type="text/css" />
    <script src="https://unpkg.com/@jspsych/plugin-survey-text@1.1.2"></script>
    <script src="https://unpkg.com/@jspsych/plugin-image-keyboard-response@1.1.2"></script>
  </head>
```

```
var fixation = {
  type: jsPsychHtmlKeyboardResponse,
  stimulus: "+",
  choices: "NO_KEYS",
  trial_duration: 500,
};

var image = {
  type: jsPsychImageKeyboardResponse,
  stimulus: "applehorse.png",
  choices: "NO_KEYS",
  trial_duration: 500
};

jsPsych.run([sentence, attention, association, fixation, image]);
```

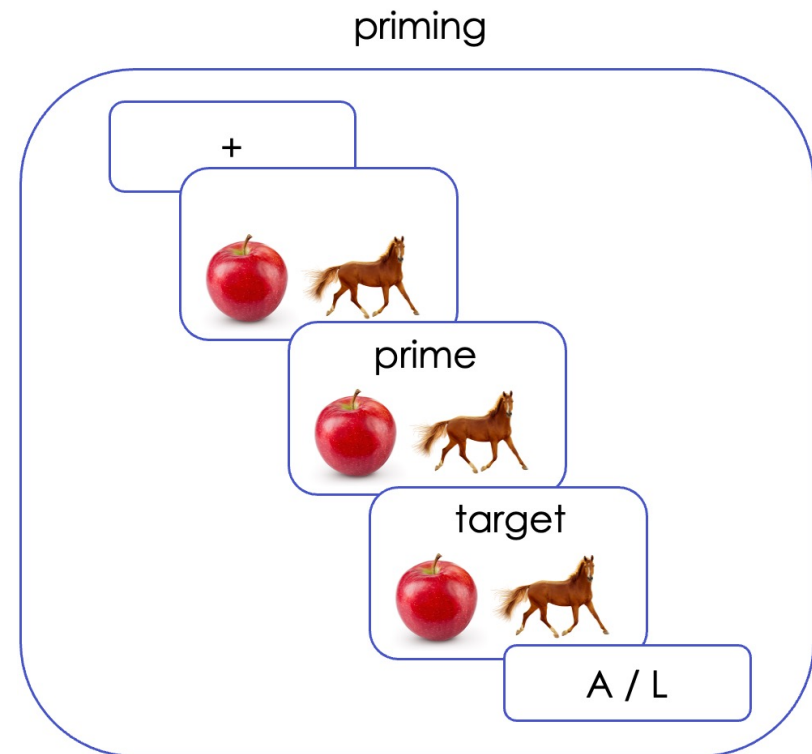
step 4b.2: image

- the image is too big for the screen and its dimensions need to be adjusted
- we can use other **parameters** of the image-keyboard-response plugin to change the dimensions
- adjust the **stimulus_width** and set **maintain_aspect_ratio** to true
- save and reload!

```
var image = {  
  type: jsPsychImageKeyboardResponse,  
  stimulus: "applehorse.png",  
  choices: "NO_KEYS",  
  trial_duration: 500,  
  stimulus_width: 500,  
  maintain_aspect_ratio: true,  
};
```


step 5: prime display

- we can now reuse the image trial to create our prime word trial
- all we need to add is an extra parameter that can display some text with the image: what could this be?
- review the [parameters for the plugin](#)



step 5a: prime display

- define a **prime** trial that uses the **prompt** parameter to display a word with an image
- save and reload!

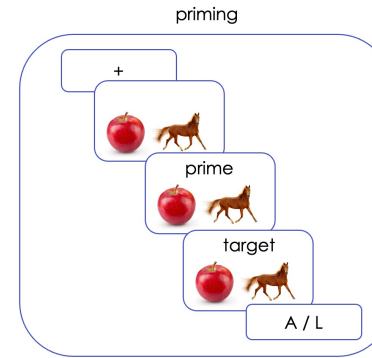
```
var prime = {  
  type: jsPsychImageKeyboardResponse,  
  stimulus: "applehorse.png",  
  //trial_duration: 300,  
  choices: "NO_KEYS",  
  stimulus_width: 500,  
  maintain_aspect_ratio: true,  
  prompt: "foobly"  
}  
  
jsPsych.run([sentence, attention, association, fixation, image, prime]);  
  
</script>  
</html>
```



foobly

step 5b: prime display

- issue 1: the prime is being displayed on the side and not in a new line
 - **solution**: add `
` before and after the word
- issue 2: the prime is being displayed below the image, not above it
 - solution: modified/customized plugin!



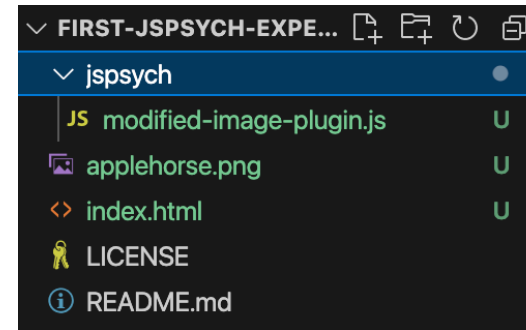
```
var prime = {
  type: jsPsychImageKeyboardResponse,
  stimulus: "applehorse.png",
  //trial_duration: 300,
  choices:"NO_KEYS",
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: "<br>foobly<br>"
}

jsPsych.run([sentence, attention, association, fixation, image, prime]);

</script>
</html>
```

step 5c: modified prime display

- create a new subfolder called “jspsych” inside your main folder
- download the [modified_image_plugin.js](#) file and save it inside the jspsych folder
- **replace** the `<script>` tag for loading image-keyboard-response to instead load this modified plugin file
- save and reload!



```
<!DOCTYPE html>
<html>
  <head>
    <title>My experiment</title>
    <script src="https://unpkg.com/jspsych@7.3.3"></script>
    <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.1.2"></script>
    <link href="https://unpkg.com/jspsych@7.3.3/css/jspsych.css" rel="stylesheet" type="text/css" />
    <script src="https://unpkg.com/@jspsych/plugin-survey-text@1.1.2"></script>
    <script src="https://unpkg.com/@jspsych/plugin-image-keyboard-response@1.1.2"></script>
    <script src="jspsych/modified-image-plugin.js"></script>
  </head>
  <body></body>
  <script>
    const jsPsych = initJsPsych();
```

foobly



testing specific screens

- it can be annoying to run through the entire sequence to review your changes to one of the screens
- **solution**: only run the specific trials you want to test!

```
var prime = {  
  type: jsPsychImageKeyboardResponse,  
  stimulus: "applehorse.png",  
  //trial_duration: 300,  
  choices:"NO_KEYS",  
  stimulus_width: 500,  
  maintain_aspect_ratio: true,  
  prompt: "<br>foobly<br>"  
}  
  
jsPsych.run([prime]);  
  
</script>  
</html>
```

making image & prime trials comparable

- to remove the jarring effect of going from the image trial to the prime display, you can add the `
` tags to prompt parameter for the image trial as well

```
var image = {
  type: jsPsychImageKeyboardResponse,
  stimulus: "applehorse.png",
  choices: "NO_KEYS",
  trial_duration: 500,
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: "<br><br>"
};

var prime = {
  type: jsPsychImageKeyboardResponse,
  stimulus: "applehorse.png",
  //trial_duration: 300,
  choices: "NO_KEYS",
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: "<br>foobly<br>"
}

jsPsych.run([image,prime]);

</script>
</html>
```

other issues/questions

- what if we wanted to increase the font of text?
 - **solution**: try span!
- what if we want to display a series of sentences? what if we want to show different words?
 - **solution**: using experiment timelines, storing stimuli in excel/CSV and importing it into jsPsych

```
var image = {
  type: jsPsychImageKeyboardResponse,
  stimulus: "applehorse.png",
  choices: "NO_KEYS",
  trial_duration: 500,
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: "<span style = 'font-size:200%'><br><br></span>"
};

var prime = {
  type: jsPsychImageKeyboardResponse,
  stimulus: "applehorse.png",
  //trial_duration: 300,
  choices:"NO_KEYS",
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: "<span style = 'font-size:200%'><br>foobly<br></span>"
}
```

step 6: target display and response

- we can recycle the prime trial to create a target trial

```
var prime = {  
  type: jsPsychImageKeyboardResponse,  
  stimulus: "applehorse.png",  
  //trial_duration: 300,  
  choices:"NO_KEYS",  
  stimulus_width: 500,  
  maintain_aspect_ratio: true,  
  prompt: "<span style = 'font-size:200%'><br>foobly<br></span>"  
}
```

```
var target = {  
  type: jsPsychImageKeyboardResponse,  
  stimulus: "applehorse.png",  
  choices:['A', 'L'],  
  stimulus_width: 500,  
  maintain_aspect_ratio: true,  
  prompt: "<span style= 'font-size:170%'>apple<br><br></span>"  
}
```

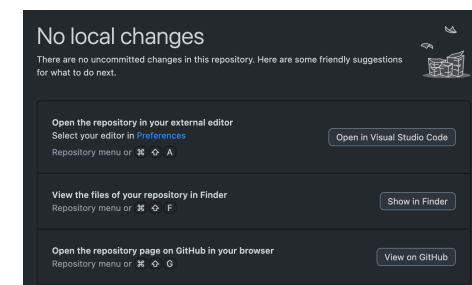
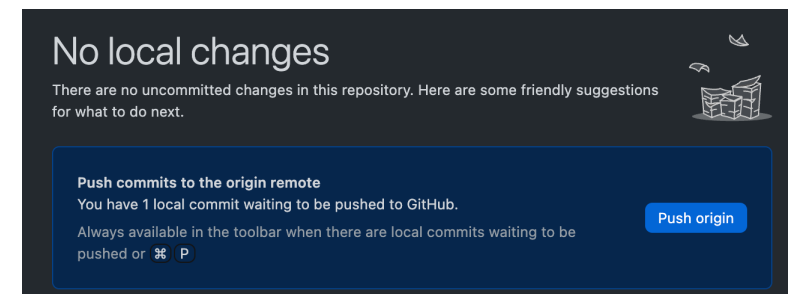
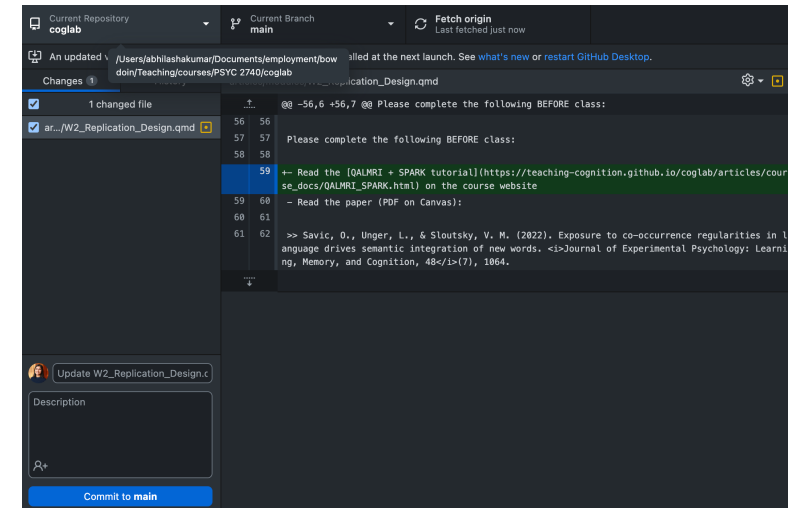

putting it all together

- modify the `jsPsych.run()` argument to include the full sequence of events
- how would you modify the run sequence?

```
jsPsych.run([sentence, attention, association, fixation, image, prime]);
```

saving your progress so far...

- save your index.html file
- open GitHub Desktop
- review changes, commit, and push
- check if changes have reflected online!



next class

- **before** class

- *prep*: <https://www.jspsych.org/7.0/overview/timeline/>
- *try*: Week 4 quiz
- *apply*: project milestone #3 (proposal)

- **during** class

- items from excel/CSV to jsPsych
- running a sequence of trials (timeline)
- feedback