

CogLab: jsPsych timeline

WEEK 4

logistics

- **project proposal**: feedback by tomorrow, will provide points based on **rubric** but ultimately no points will be taken off at this stage
- **project**: design draft (milestone #3 jsPsych) is due this weekend
- first **formative assignment** due after fall break
- **use office hours** as questions come up!

3	Sunday, September 22, 2024	Project Milestone #2 (Project Proposal) Due
4	Tuesday, September 24, 2024	<u>W4: Experiment Timeline</u>
4	Thursday, September 26, 2024	W4 continued...
4	Sunday, September 29, 2024	Project Milestone #3 (Design Draft) Due
5	Tuesday, October 1, 2024	<u>W5: Recording Data</u>
5	Thursday, October 3, 2024	W5 continued...
6	Tuesday, October 8, 2024	Fall Break!! NO CLASS
6	Thursday, October 10, 2024	<u>W6: Experiment Workflow</u>
6	Sunday, October 13, 2024	Formative Assignment (jsPsych) Due



resolving git commit issues

going back to our experiment

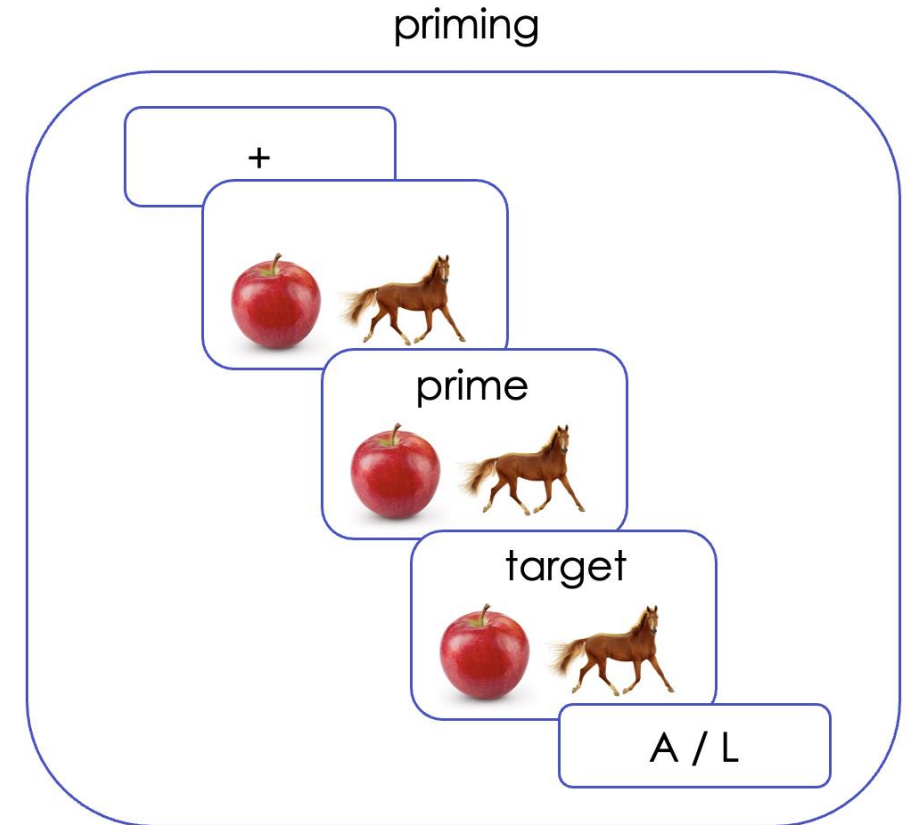
- open Visual Studio Code and open the jsPsych experiment you created last week
- also open the [index.html](#) file in your browser to remind yourself of what we did!

recap

- what we covered:
 - jsPsych 101
 - building a sequence of events/screens
- your to-dos were:
 - *prep*: <https://www.jspsych.org/7.0/overview/timeline/>
 - *apply*: project milestone #2 (proposal)

recap: priming

- 4 components:
 - fixation cross
 - image
 - prime + image
 - target + image + response



recap: priming code

```
<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,
};
```

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

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

step 6: target display and response

- we can recycle the prime trial to create a target trial
- modify the `jsPsych.run()` argument to include the full sequence of events

```
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>"  
}
```

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


experiment recap

training

association x 3

priming

sentence

space

novel word?

<response>

word

<response>

x 3

+



prime



target



A / L

today's agenda

- importing `stimuli` to jsPsych
- repeating sequence of events for different items

building stimuli files

- what do we need for this experiment?
 - a list of sentences
 - a list of prime-target pairs
 - images
- download the stimuli folder into the same directory as your `index.html` file

... > jspsych materials > stimuli ▾ 👤

Type ▾

People ▾

Modified ▾

Name ↓

Owner

📄 sentences.csv 👤

👤 me

📄 priming.csv 👤

👤 me

📁 first-jspsych-experiment 🔄 >

📁 Fall 2023 🔄 >

📁 stimuli >

📄 index.html

📄 priming.csv

📄 sentences.csv

sentences.csv

- `sentences.csv` contains the specific sentences that participants saw during the training phase organized row-wise
- what information does each column contain?

Triad	Pair	Part	sentence	novel1	novel2	novel3
1	foobly apple	training	I went to Zimziland because I heard you can get a foobly apple there.	foobly	Zimziland	
1	foobly apple	training	My sister doesn't like to have a foobly apple.	foobly		
1	foobly apple	training	Everybody I met in Zimziland would love to have a foobly apple.	foobly	Zimziland	
1	foobly apple	training	As soon as I arrived, I noticed a foobly apple.	foobly		
1	foobly apple	training	I would really love to have a foobly apple.	foobly		
1	foobly apple	training	One lady offered me a foobly apple.	foobly		
1	foobly apple	training	Maybe I would prefer a foobly apple.	foobly		
1	foobly apple	training	I went to a place where it's easy to find a foobly apple.	foobly		
1	foobly apple	training	The people I met there told me they saw a foobly apple that day.	foobly		
1	foobly apple	training	In the end, I decided I'd go looking for a foobly apple.	foobly		
1	foobly mipp	training	I didn't find a foobly mipp.	foobly	mipp	
1	foobly mipp	training	My sister told me you can get a foobly mipp there.	foobly	mipp	
1	foobly mipp	training	My friend Sally said she saw a foobly mipp once.	foobly	mipp	

importing sentences into `index.html`

- CSV is not a format that is easily understood by jsPsych
- **solution**: convert each list to a format that jsPsych understands
 - JSON is a file format that jsPsych understands
 - excel/CSV to JSON conversion can be done online pretty easily!

Triad	Pair	Part	sentence	novel1	novel2	novel3
1	foobly apple	training	I went to Zimziland because I heard you can get a foobly apple there.	foobly	Zimziland	
1	foobly apple	training	My sister doesn't like to have a foobly apple.	foobly		
1	foobly apple	training	Everybody I met in Zimziland would love to have a foobly apple.	foobly	Zimziland	
1	foobly apple	training	As soon as I arrived, I noticed a foobly apple.	foobly		
1	foobly apple	training	I would really love to have a foobly apple.	foobly		
1	foobly apple	training	One lady offered me a foobly apple.	foobly		
1	foobly apple	training	Maybe I would prefer a foobly apple.	foobly		
1	foobly apple	training	I went to a place where it's easy to find a foobly apple.	foobly		
1	foobly apple	training	The people I met there told me they saw a foobly apple that day.	foobly		
1	foobly apple	training	In the end, I decided I'd go looking for a foobly apple.	foobly		
1	foobly mipp	training	I didn't find a foobly mipp.	foobly	mipp	
1	foobly mipp	training	My sister told me you can get a foobly mipp there.	foobly	mipp	
1	foobly mipp	training	My friend Sally said she saw a foobly mipp once.	foobly	mipp	

converting excel to JSON

- go to <https://csvjson.com/csv2json>
- upload **sentences.csv**
- click convert

CSV or TSV > JSON

To get started, upload or paste your data from Excel (saved as CSV or TSV).

Upload a CSV file

Choose File No file chosen

Or paste your CSV here

```
noticed a foobly apple.",foobly,,  
1,foobly apple,training,I would really love to  
have a foobly apple.,foobly,,  
1,foobly apple,training,One lady offered me a  
foobly apple.,foobly,,  
1,foobly apple,training,Maybe I would prefer a  
foobly apple.,foobly,,  
1,foobly apple,training,I went to a place where  
it's easy to find a foobly apple.,foobly,,  
1,foobly apple,training,The people I met there  
told me they saw a foobly apple that  
day.,foobly,,  
1,foobly apple,training,"In the end, I decided  
I'd go looking for a foobly apple.",foobly,,  
1,foobly mipp,training,I didn't find a foobly  
mipp.,foobly,mipp,  
1,foobly mipp,training,My sister told me you can  
get a foobly mipp there.,foobly,mipp,  
1.foobly minn.training.Mv friend Sally said she
```

> Convert

⊗ Clear

saving a .js file

- download and save the file as `all_sentences.js` to the same folder as `index.html`
- open the file in VS Code

Separator

Auto-detect ▾

Parse numbers

Parse JSON

Transpose

Output: Array Hash Minify

JSON

```
[
  {
    "Triad": 1,
    "Pair": "foobly apple",
    "Part": "training",
    "sentence": "I went to Zimziland because I heard you can get a
foobly apple there.",
    "novel1": "foobly",
    "novel2": "Zimziland",
    "novel3": ""
  },
  {
    "Triad": 1,
    "Pair": "foobly apple",
    "Part": "training",
    "sentence": "My sister doesn't like to have a foobly apple.",
    "novel1": "foobly",
    "novel2": "",
    "novel3": ""
  }
]
```

Download

Save

Copy to clipboard

examining `all_sentences.js`

- notice that this file starts with a square bracket: this is a Javascript **array/object**
- each row in your CSV is defined by one `{}`, and each column and its value is denoted by `column-name : value`
- what is the name of the column that contains the sentence?
- to use it in your experiment you need to assign a name to this array
- add `var list_of_sentences =` before the square bracket
- save the file

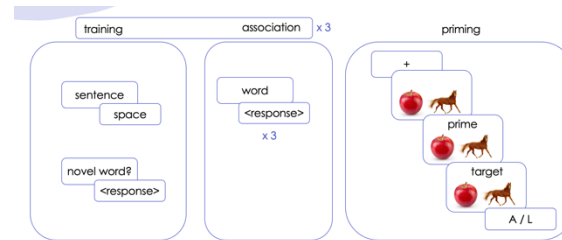
```
[
  {
    "Triad": 1,
    "Pair": "foobly apple",
    "Part": "training",
    "sentence": "I went to Zimziland because I heard you can get a foobly apple there.",
    "novel1": "foobly",
    "novel2": "Zimziland",
    "novel3": ""
  },
  {
    "Triad": 1,
    "Pair": "foobly apple",
    "Part": "training",
    "sentence": "My sister doesn't like to have a foobly apple.",
    "novel1": "foobly",
    "novel2": "",
    "novel3": ""
  }
],
```

```
var list_of_sentences = [
  {
    "Triad": 1,
    "Pair": "foobly apple",
    "Part": "training",
    "sentence": "I went to Zimziland because
    "novel1": "foobly",
    "novel2": "Zimziland",
    "novel3": "NOT_FOUND"
  },
]
```


loading `all_sentences.js` into `index.html`

- import the `all_sentences.js` file using the `<script>` tag inside the `<head>` tag as before
- now think about which trials in the experiment need to have these sentences, and **which parameter will we need to modify for this**

```
<!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="jspsych/modified-image-plugin.js"></script>
    <script src="all_sentences.js"></script>
  </head>
  <body></body>
</html>
```



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

modifying our plugin

- we need to modify the **sentence** plugin trial (**stimulus** parameter) to import the list of sentences instead of only displaying one sentence
- we use a specific function called `jsPsych.timelineVariable()` with the argument being the **name of the column** that contains our sentences
- what if I wanted to instead display the word pair?

```
var sentence = {  
  type: jsPsychHtmlKeyboardResponse,  
  stimulus: jsPsych.timelineVariable('sentence'),  
  choices: [' ']  
}
```

```
var list_of_sentences = [  
  {  
    "Triad": 1,  
    "Pair": "foobly apple",  
    "Part": "training",  
    "sentence": "I went to Zimziland because",  
    "novel1": "foobly",  
    "novel2": "Zimziland",  
    "novel3": "NOT_FOUND"  
  },  
  ]
```

creating a timeline

- next, we need to connect the `jsPsych.timelineVariable()` function inside the plugin trial to the `.js` file we created using a `timeline`
- here, we are defining a timeline that consists of the `sentence` plugin trial, uses the `sentences` variable we defined inside `sentences.js`

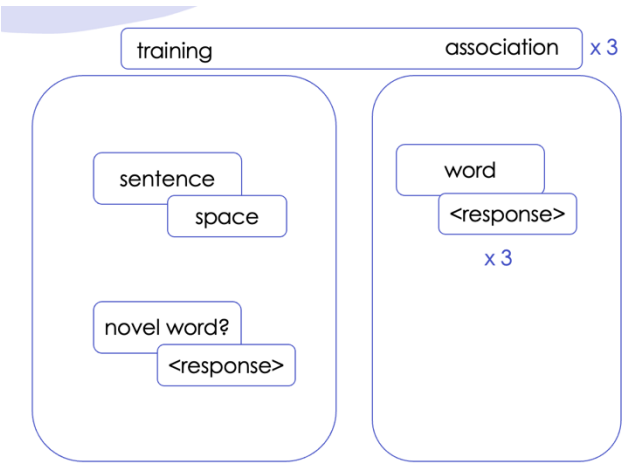
```
var sentence = {  
  type: jsPsychHtmlKeyboardResponse,  
  stimulus: jsPsych.timelineVariable('sentence'),  
  choices: [' ']  
}
```

```
var training_procedure = {  
  timeline: [sentence],  
  timeline_variables: list_of_sentences,  
};
```

```
var list_of_sentences = [  
  {  
    "Triad": 1,  
    "Pair": "foobly apple",  
    "Part": "training",  
    "sentence": "I went to Zimziland because",  
    "novel1": "foobly",  
    "novel2": "Zimziland",  
    "novel3": "NOT_FOUND"  
  },  
  ]
```

creating a timeline

- try running just the `training_procedure`
- save and reload `index.html`
- randomize the order of sentences, reload



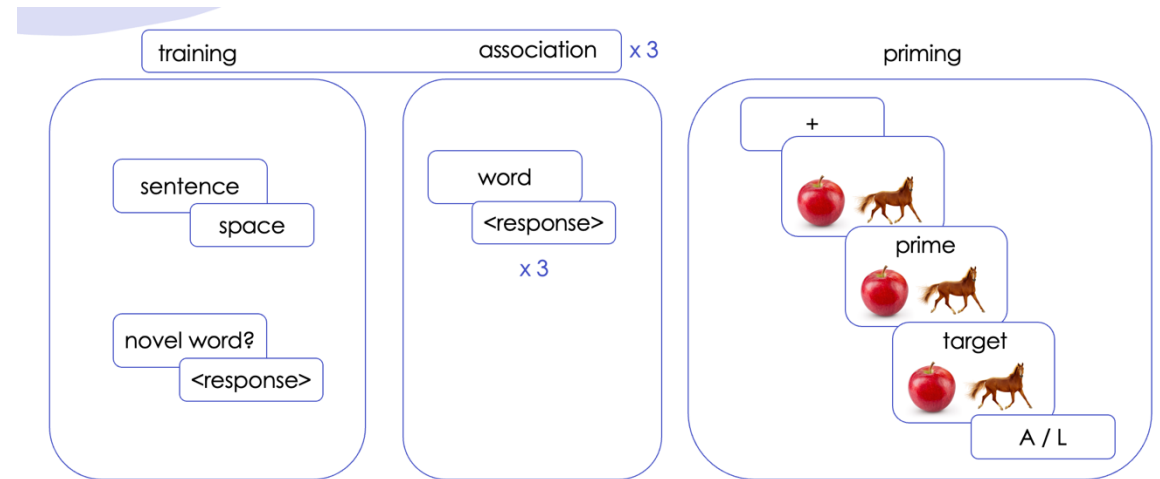
```
var training_procedure = {  
  timeline: [sentence],  
  timeline_variables: list_of_sentences,  
};
```

```
jsPsych.run([training_procedure]);
```

```
var training_procedure = {  
  timeline: [sentence],  
  timeline_variables: list_of_sentences,  
  randomize_order: true  
};
```

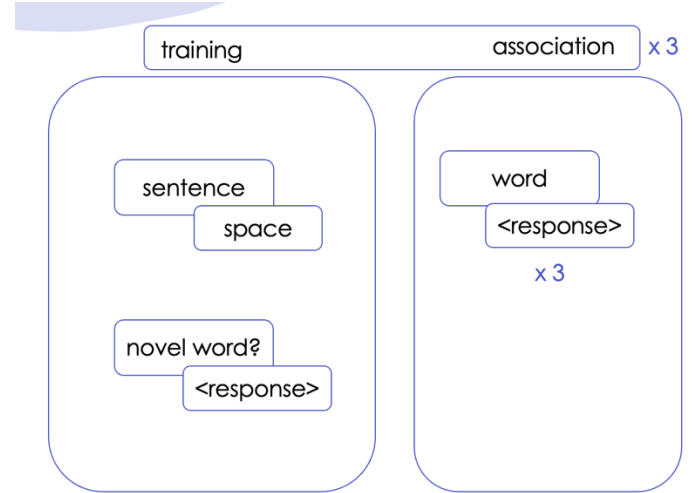
adding to the timeline

- we need to add the **association** trials after the sentences
- we also need a way to display an **attention check** every few trials



creating association stimuli

- what words were presented as cues for the association task?
- create a new file called `association.js` inside the same folder as `index.html`
- using the same format as `all_sentences.js`, create 4 rows with the column name as cue and the value inside the column being the novel words



```
var association_cues = [
  {
    cue: "foobly"
  },
  {
    cue: "mipp"
  },
  {
    cue: "dodish"
  },
  {
    cue: "geck"
  }
]
```

importing associations.js into index.html

- add a `<script>` tag to include the .js file into your experiment
- now modify your association plugin trial to import the column that contains the novel words inside the association.js file

```
<!DOCTYPE html>
<html>
  <head>
    <title>My experiment</title>
    <script src="https://unpkg.com/jspsych@7.3.3">
    <script src="https://unpkg.com/@jspsych/plugin
    <link href="https://unpkg.com/jspsych@7.3.3/cs
    <script src="https://unpkg.com/@jspsych/plugin
    <script src="jspsych/modified-image-plugin.js"
    <script src="all_sentences.js"></script>
    <script src="association.js"></script>
  </head>
  <body></body>
  <script>
```

```
var association_cues = [
  {
    cue: "foobly"
  },
  {
    cue: "mipp"
  },
  {
    cue: "dodish"
  },
  {
    cue: "geck"
  }
]
```

```
var association = {
  type: jsPsychSurveyText,
  questions: [{prompt: jsPsych.timelineVariable('cue')}],
};
```

creating a procedure for association

- we define `association_procedure` that pulls the variable from `associations.js`

```
<!DOCTYPE html>
<html>
  <head>
    <title>My experiment</title>
    <script src="https://unpkg.com/jspsych@7.3.3">
    <script src="https://unpkg.com/jspsych/plugin
    <link href="https://unpkg.com/jspsych@7.3.3/cs
    <script src="https://unpkg.com/jspsych/plugin
    <script src="jspsych/modified-image-plugin.js"
    <script src="all_sentences.js"></script>
    <script src="association.js"></script>
  </head>
  <body></body>
  <script>
```

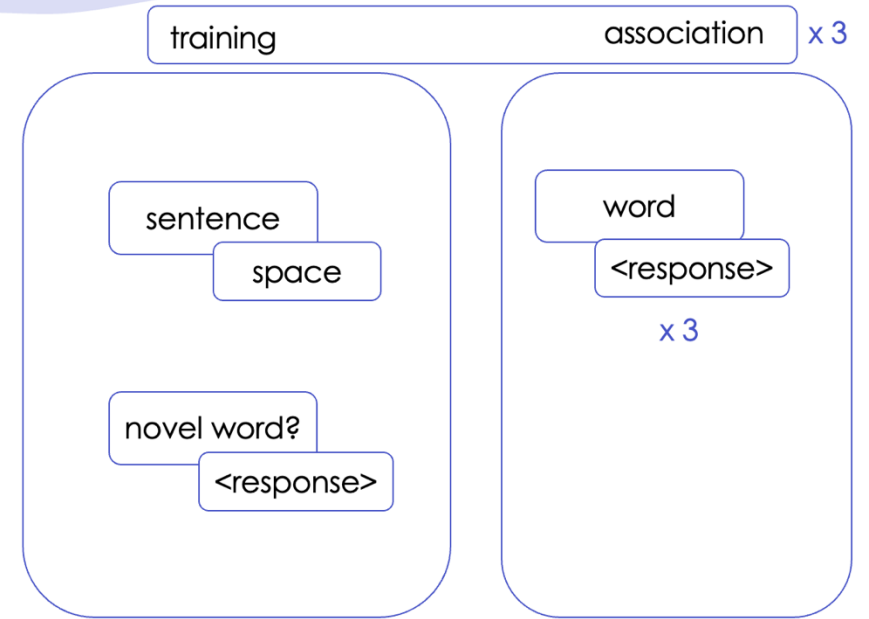
```
var association_cues = [
  {
    cue: "foobly"
  },
  {
    cue: "mipp"
  },
  {
    cue: "dodish"
  },
  {
    cue: "geck"
  }
]
```

```
var association = {
  type: jsPsychSurveyText,
  questions: [{prompt: jsPsych.timelineVariable('cue')}],
};
```

```
var association_procedure = {
  timeline: [association],
  timeline_variables: association_cues,
};
```


repeating associations

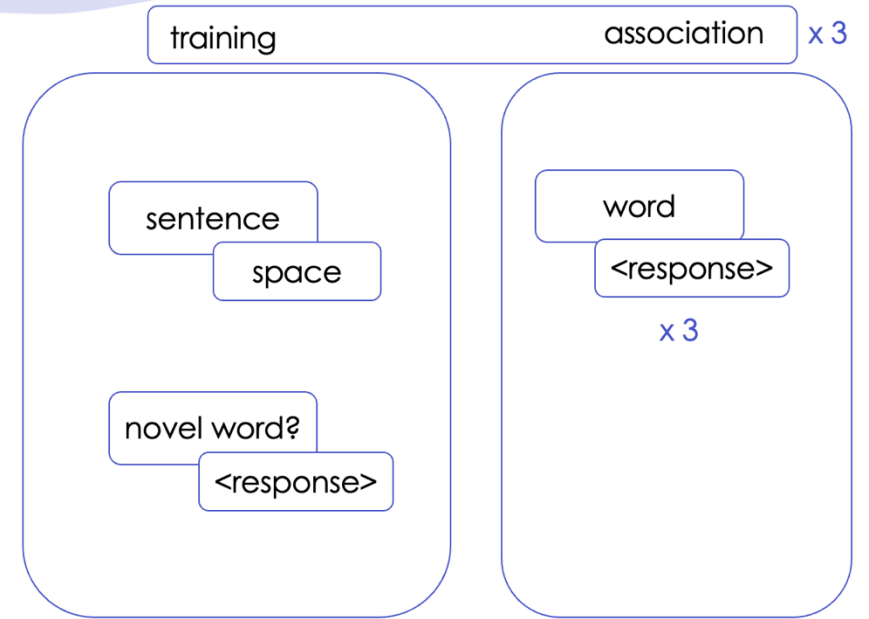
- we also want the association procedure to be repeated three times and in random order



```
var association_procedure= {  
  timeline: [association],  
  timeline_variables: association_cues,  
  randomize_order: true,  
  repetitions: 3  
};
```

putting it all together

- the sentence + (attention) + association sequence occurs three times in the experiment
- we achieve this by defining yet another timeline variable



```
var training_plus_association = {  
  timeline: [training_procedure, association_procedure],  
  repetitions: 3  
}  
  
jsPsych.run([training_plus_association]);
```

testing hacks

- running through all 40 sentences can be annoying
- **solution**: you could reduce the `trial_duration` when you're testing different parts of your experiment

```
var sentence = {  
  type: jsPsychHtmlKeyboardResponse,  
  stimulus: jsPsych.timelineVariable('sentence'),  
  choices: [' '],  
  trial_duration: 10  
}
```

experiment recap

training

association x 3

priming

sentence

space

novel word?

<response>

word

<response>

x 3

+



prime



target



A / L

explore priming.csv

block_number	triad	target_image	part	prime_word	target_word	type	relatedness	correct_resp	image_path	correct_key	
practice		1	apple-horse	priming	boff	apple	novel	novel	1	applehorse.png	A
practice		2	apple-horse	priming	nuppical	horse	novel	novel	3	applehorse.png	L
practice		2	apple-horse	priming	boff	horse	novel	novel	3	applehorse.png	L
practice		1	horse-apple	priming	nuppical	apple	novel	novel	3	horseapple.png	L
practice		1	apple-horse	priming	boff	apple	novel	novel	1	applehorse.png	A
practice		2	apple-horse	priming	nuppical	horse	novel	novel	3	applehorse.png	L
practice		2	apple-horse	priming	boff	horse	novel	novel	3	applehorse.png	L
practice		1	horse-apple	priming	nuppical	apple	novel	novel	3	horseapple.png	L
	1	1	apple-horse	priming	foobly	apple	direct	related	1	applehorse.png	A
	1	1	apple-horse	priming	mipp	apple	shared	related	1	applehorse.png	A
	1	1	apple-horse	priming	dodish	apple	direct	unrelated	1	applehorse.png	A
	1	1	apple-horse	priming	geck	apple	shared	unrelated	1	applehorse.png	A
	1	1	apple-horse	priming	nuppical	apple	novel	novel	1	applehorse.png	A
	1	1	apple-horse	priming	boff	apple	novel	novel	1	applehorse.png	A
	1	2	apple-horse	priming	foobly	horse	direct	unrelated	3	applehorse.png	L
	1	2	apple-horse	priming	mipp	horse	shared	unrelated	3	applehorse.png	L
	1	2	apple-horse	priming	dodish	horse	direct	related	3	applehorse.png	L
	1	2	apple-horse	priming	geck	horse	shared	related	3	applehorse.png	L
	1	2	apple-horse	priming	nuppical	horse	novel	novel	3	applehorse.png	L
	1	2	apple-horse	priming	boff	horse	novel	novel	3	applehorse.png	L
	1	1	horse-apple	priming	foobly	apple	direct	related	3	horseapple.png	L
	1	1	horse-apple	priming	mipp	apple	shared	related	3	horseapple.png	L

creating priming.js

- convert the priming.CSV file to priming.js using <https://csvjson.com/csv2js> on
- download and save as priming.js
- define `var practice_stimuli` and `var test_stimuli`

```
1 var practice_stimuli = [  
2   {  
3     "block_number": "practice",  
4     "triad": 1,  
5     "target_image_pair": "apple-horse",  
6     "part": "priming",  
7     "prime_word": "boff",  
8     "target_word": "apple",  
9     "type": "novel",  
10    "relatedness": "novel",  
11    "correct_response": 1,  
12    "image_path": "applehorse.png",  
13    "correct_key": "A"  
14  },  
15  {  
16    "block_number": "practice",  
17    "triad": 2,  
18    "target_image_pair": "apple-horse",  
19    "part": "priming",  
20    "prime_word": "nuppical",
```

```
101    "relatedness": "novel",  
102    "correct_response": 3,  
103    "image_path": "horseapple.png",  
104    "correct_key": "L"  
105  }  
106 ]  
107  
108  
109 var test_stimuli = [  
110   {  
111     "block_number": 1,  
112     "triad": 1,
```

load into index.html

- use the `<script>` tags as before
- change the stimulus parameters for the `image` plugin trial to the column that stores the names of the images that need to be displayed

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My experiment</title>
5     <script src="https://unpkg.com/jspsych@7.3.3"
6     <script src="https://unpkg.com/@jspsych/plugin-
7     <link href="https://unpkg.com/jspsych@7.3.3/c
8     <script src="https://unpkg.com/@jspsych/plugin-
9     <script src="jspsych/modified-image-plugin.js
10    <script src ="sentences.js"></script>
11    <script src ="association.js"></script>
12    <script src ="priming.js"></script>
13  </head>
```

```
1 var practice_stimuli = [
2   {
3     "block_number": "practice",
4     "triad": 1,
5     "target_image_pair": "apple-horse",
6     "part": "priming",
7     "prime_word": "boff",
8     "target_word": "apple",
9     "type": "novel",
10    "relatedness": "novel",
11    "correct_response": 1,
12    "image_path": "applehorse.png",
13    "correct_key": "A"
14  },
15  {
16    "block_number": "practice",
17    "triad": 2,
18    "target_image_pair": "apple-horse",
19    "part": "priming",
20    "prime_word": "nuppical",
```

```
var image = {
  type: jsPsychImageKeyboardResponse,
  stimulus: jsPsych.timelineVariable('image_path'),
  choices: "NO_KEYS",
  trial_duration: 500,
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: "<span style = 'font-size:200%'"
};
```

modifying prime and target trials: 1

- change the stimulus being displayed
- do we need to change anything else?
- we also need to change the **prompt** being displayed, as our prompt contains the prime/target words

```
var prime = {  
  type: jsPsychImageKeyboardResponse,  
  stimulus: jsPsych.timelineVariable('image_path'),  
  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: jsPsych.timelineVariable('image_path'),  
  choices: ['A', 'L'],  
  stimulus_width: 500,  
  maintain_aspect_ratio: true,  
  prompt: "<span style = 'font-size:200%'><br>foobly<br></span>"  
}
```


modifying prime and target trials: 1

- inside `priming.js`, what column names contain the words to be displayed on prime and target trials?
- modify the prompt accordingly

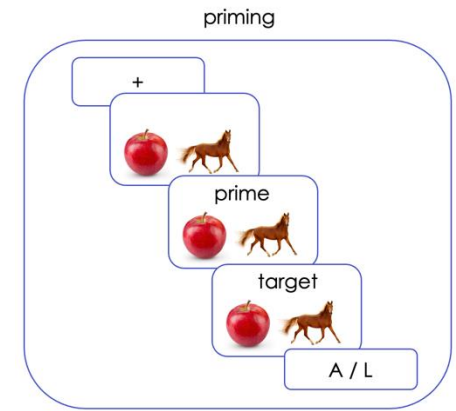
```
1 var practice_stimuli = []
2
3 {
4   "block_number": "practice",
5   "triad": 1,
6   "target_image_pair": "apple-horse",
7   "part": "priming",
8   "prime_word": "boff",
9   "target_word": "apple",
10  "type": "novel",
11  "relatedness": "novel",
12  "correct_response": 1,
13  "image_path": "applehorse.png",
14  "correct_key": "A"
15 },
16 {
17   "block_number": "practice",
18   "triad": 2,
19   "target_image_pair": "apple-horse",
20   "part": "priming",
21   "prime_word": "nuppical",
```

```
var prime = {
  type: jsPsychImageKeyboardResponse,
  stimulus: jsPsych.timelineVariable('image_path'),
  trial_duration: 300,
  choices: "NO_KEYS",
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: jsPsych.timelineVariable('prime_word')
}

var target = {
  type: jsPsychImageKeyboardResponse,
  stimulus: jsPsych.timelineVariable('image_path'),
  choices: ['A', 'L'],
  stimulus_width: 500,
  maintain_aspect_ratio: true,
  prompt: jsPsych.timelineVariable('target_word')
}
```

creating a priming procedure

- create a timeline variable that has a sequence of events that all use the priming.js file
- run this priming procedure
- save and reload index.html in your browser



```
var priming_proc = {  
  timeline: [fixation, image, prime, target],  
  timeline_variables: practice_stimuli,  
  randomize_order: true  
};
```

```
jsPsych.run([priming_proc]);
```

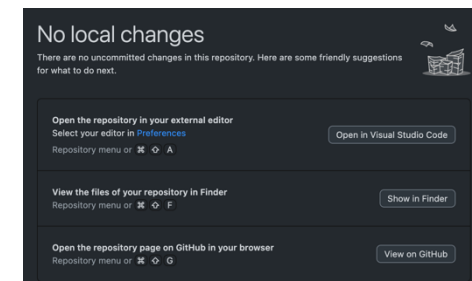
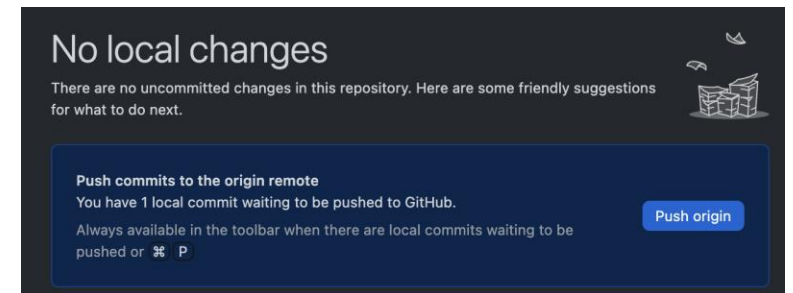
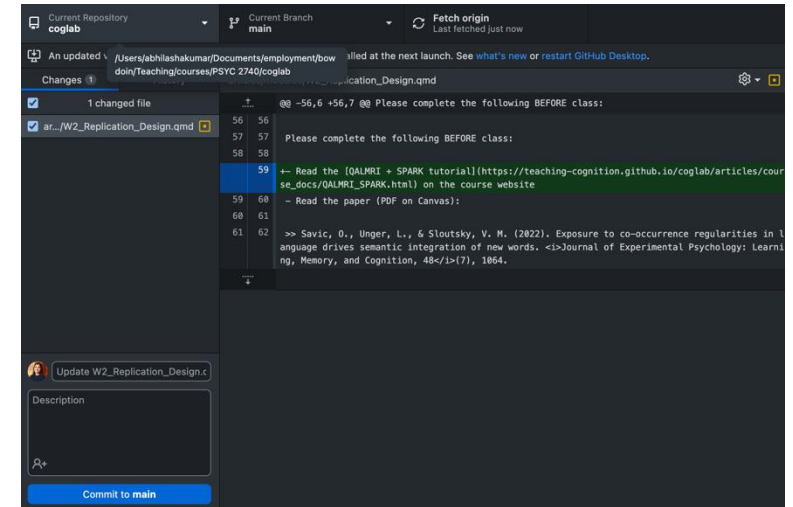
sum of the parts!

- now, we have a version of the experiment where training sentences are presented, free association happens, and the priming task is conducted
- **modify the run statement** to see the current experiment workflow

```
jsPsych.run([training_plus_association, priming_proc]);
```

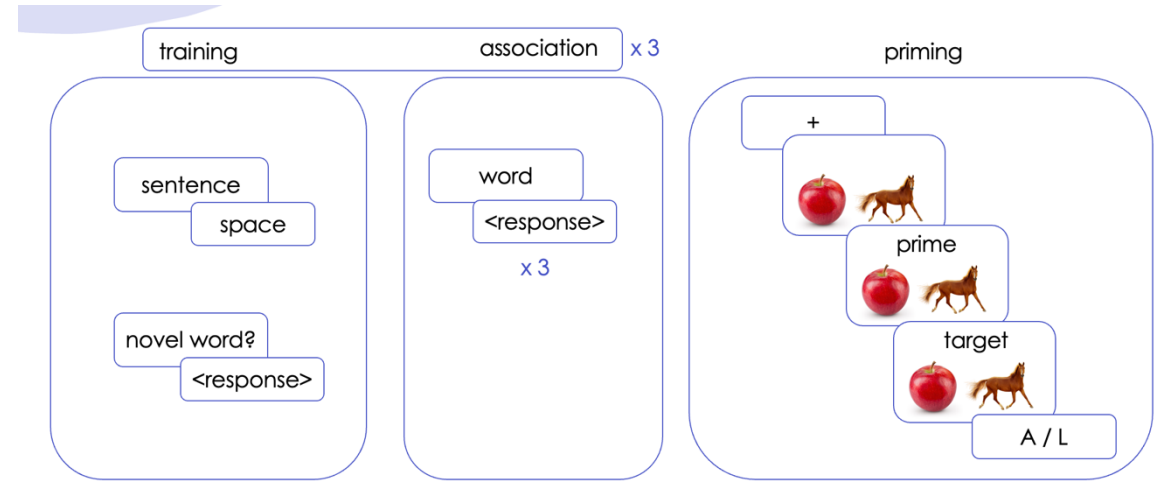
saving your progress so far...

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



outstanding issues

- fixing position & style of prime/target words
- adding instruction screens
- attention checks
- feedback
- recording data



next class

- **before** class

- *prep*: conditional timelines and providing feedback
- *prep*: design draft (project milestone #3)

- **during** class

- fixing position & style of prime/target words
- adding instruction screens
- attention checks
- feedback