Workshop jsPsych

*Session 1 - **Les bases***

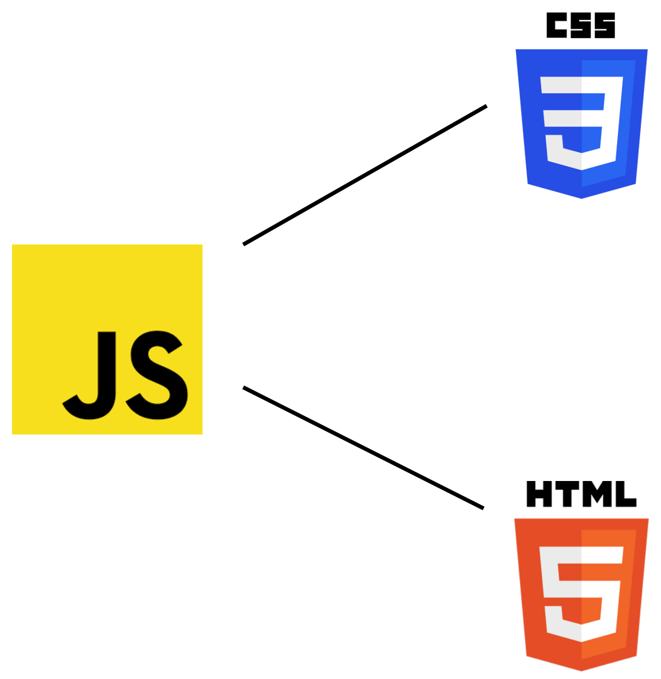
jspsych

Objectifs de la session 1

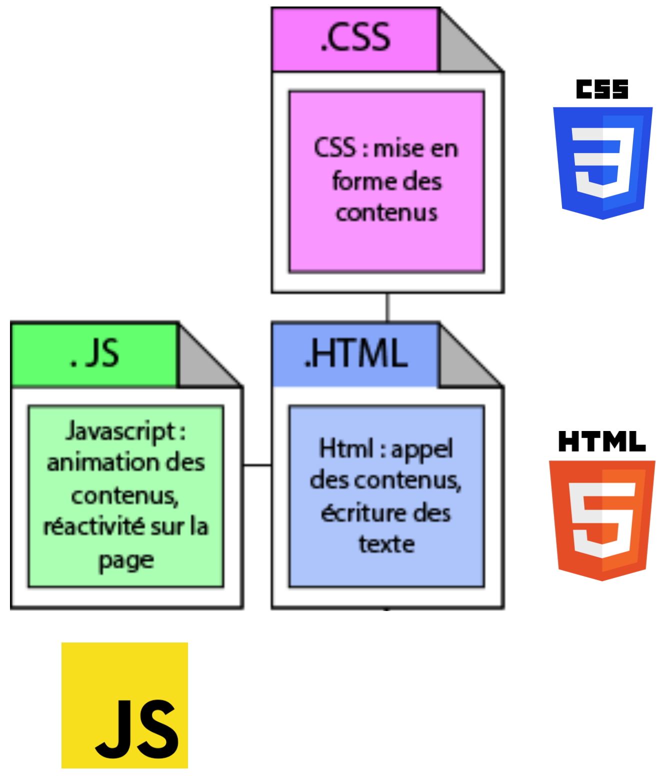
- Comprendre l’architecture *jsPsych*,
- Comprendre l’architecture web qui englobe notre code,
- Pouvoir utiliser *cognition.run* pour coder,
- Pouvoir mettre en place une expérience,
- Pouvoir récupérer les données de son expérience

L’architecture jsPsych

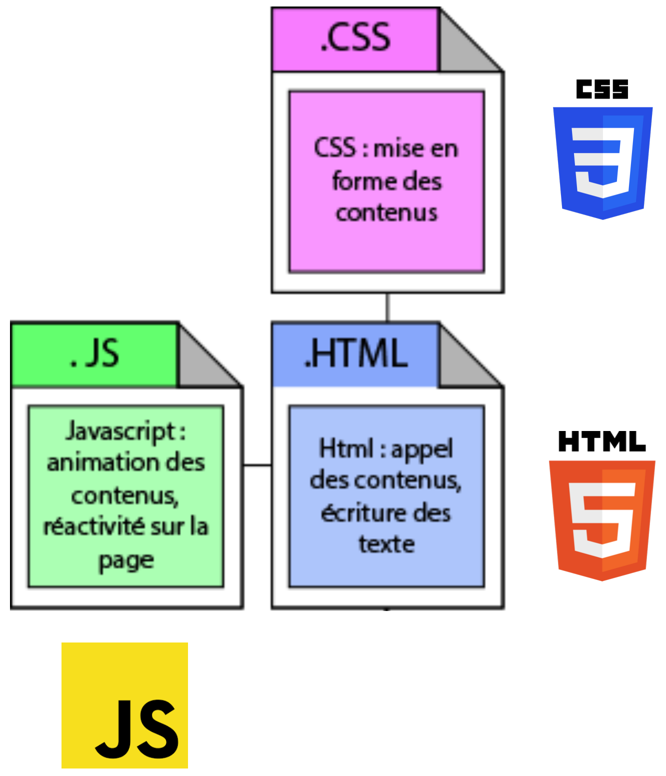
- Les trois languages du web

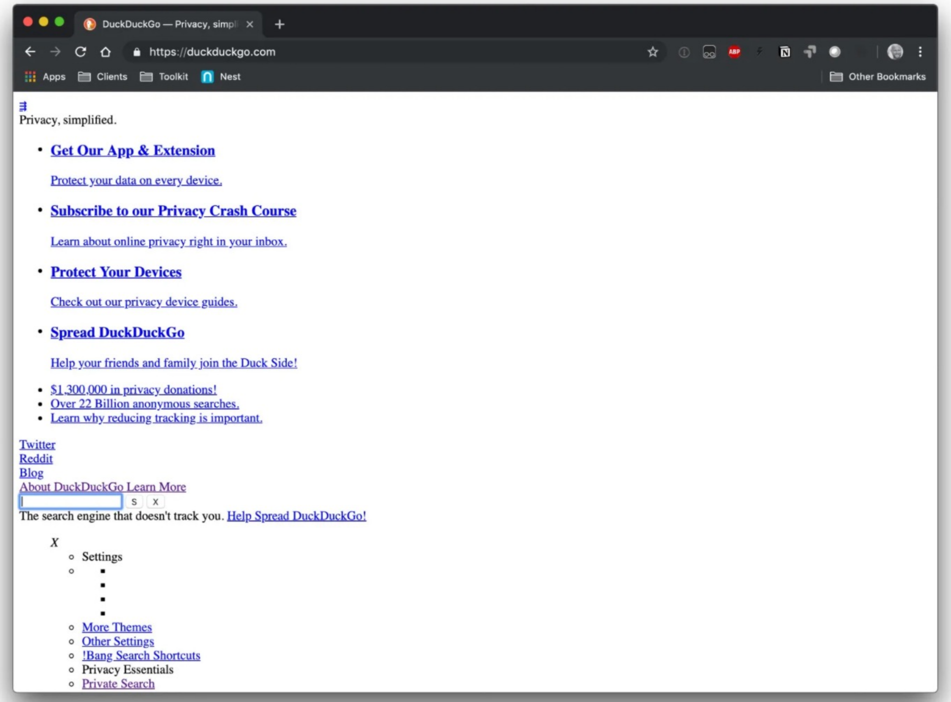


L’architecture jsPsych


.CSS
CSS : mise en forme des contenus
CSS
.JS
Javascript : animation des contenus, réactivité sur la page
.HTML
Html : appel des contenus, écriture des texte
HTML
JS

L’architecture jsPsych


.CSS
CSS : mise en forme des contenus
CSS
.JS
Javascript : animation des contenus, réactivité sur la page
.HTML
Html : appel des contenus, écriture des texte
HTML
JS



Privacy, simplified.

- **Get Our App & Extension**

  Protect your data on every device.
- **Subscribe to our Privacy Crash Course**

  Learn about online privacy right in your inbox.
- **Protect Your Devices**

  Check out our privacy device guides.
- **Spread DuckDuckGo**

  Help your friends and family join the Duck Side!

- $1,300,000 in privacy donations!
- Over 22 Billion anonymous searches.
- Learn why reducing tracking is important.

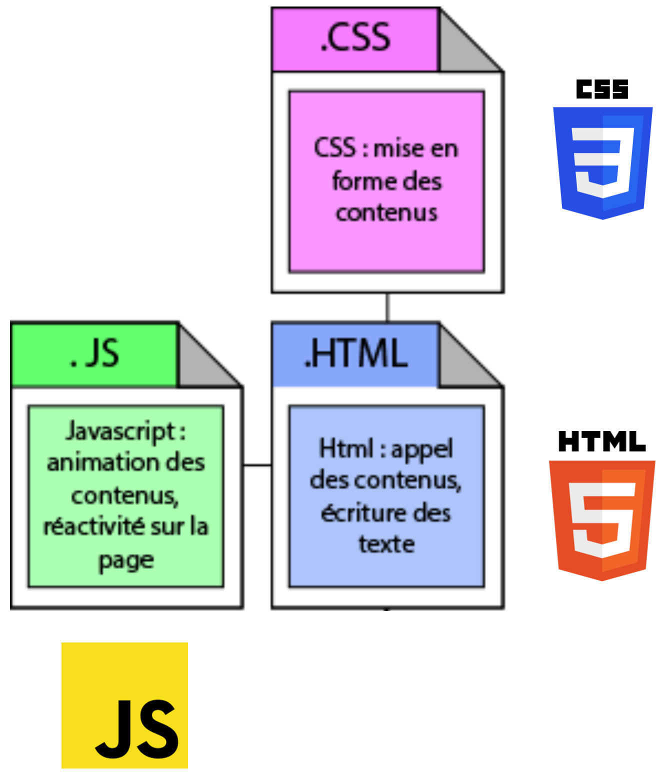
Twitter
Reddit
Blog
About DuckDuckGo Learn More

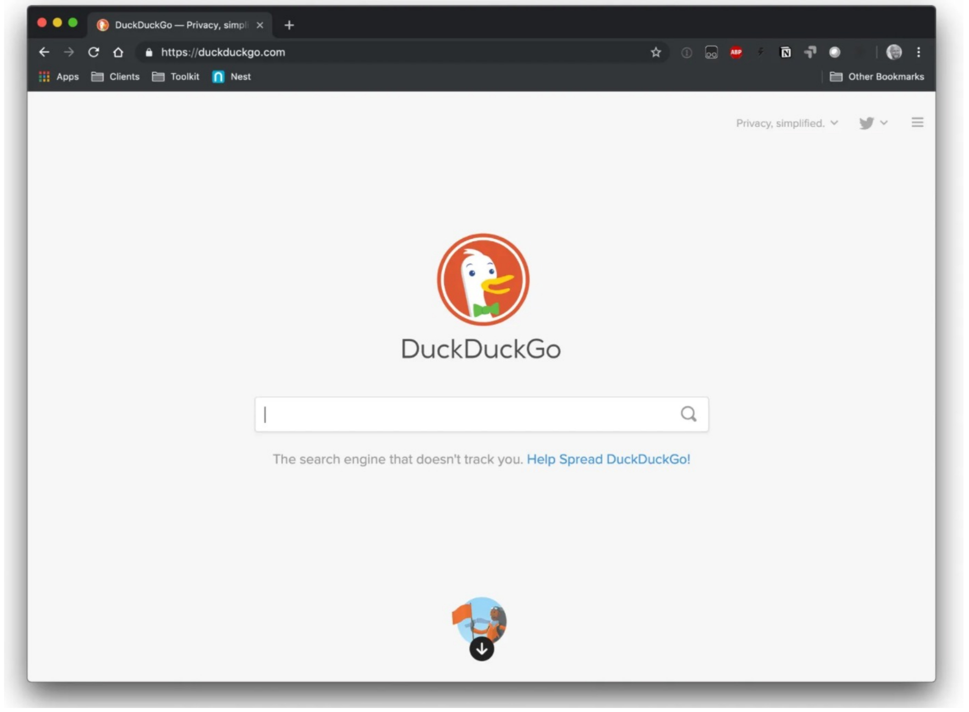
S X

The search engine that doesn't track you. Help Spread DuckDuckGo!

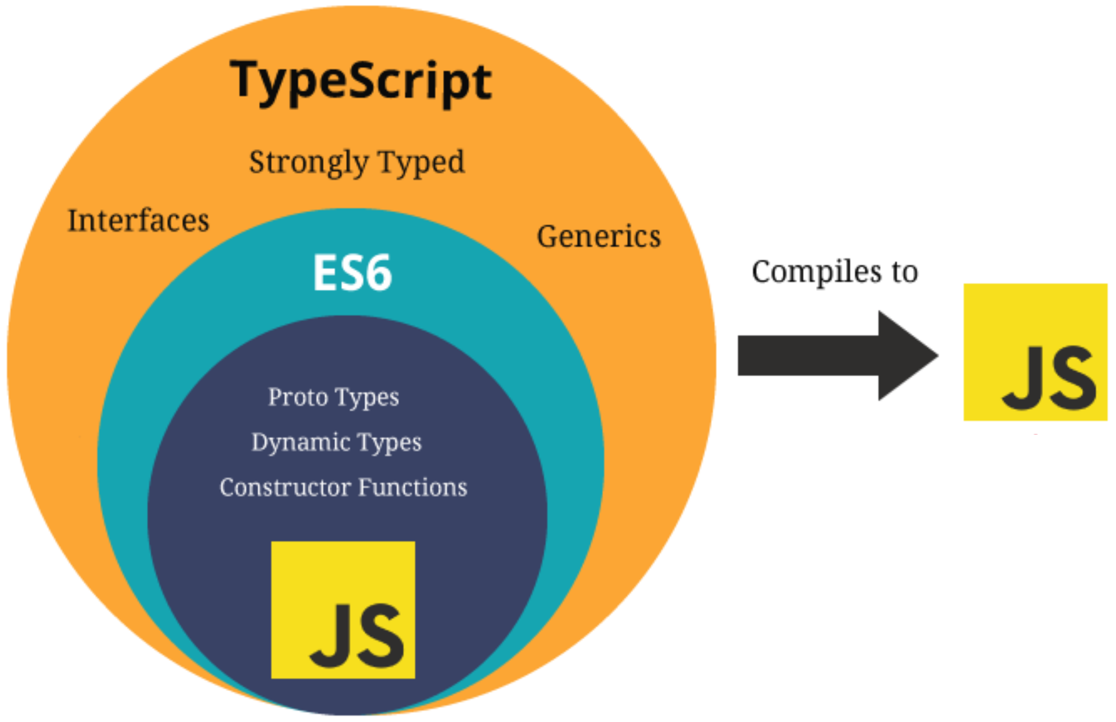
X
- Settings
- More Themes
- Other Settings
- !Bang Search Shortcuts
- Privacy Essentials
- Private Search

L’architecture jsPsych

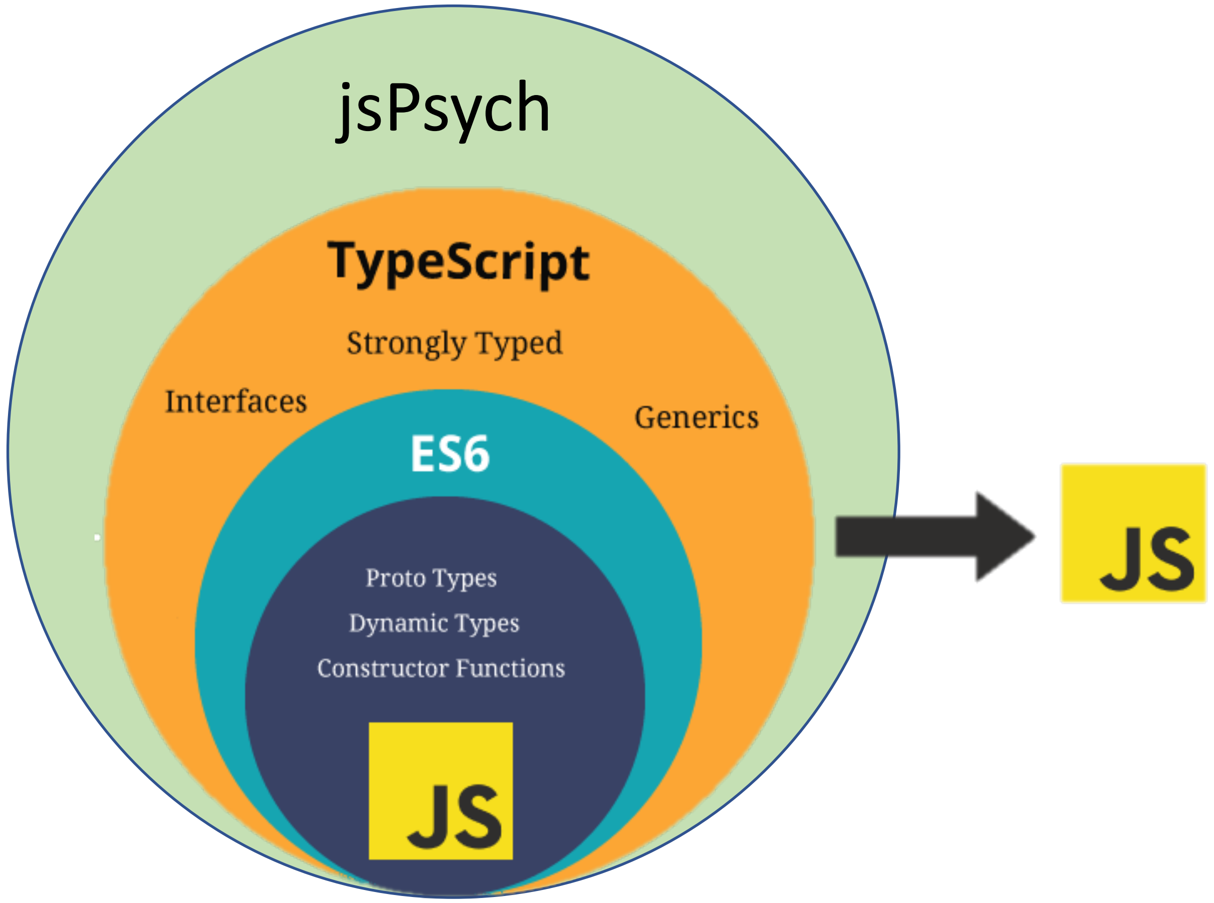

.CSS
CSS : mise en forme des contenus
CSS
.JS
Javascript : animation des contenus, réactivité sur la page
.HTML
Html : appel des contenus, écriture des texte
HTML
JS


DuckDuckGo — Privacy, simpl
https://duckduckgo.com
Apps
Clients
Toolkit
Nest
Other Bookmarks
Privacy, simplified.
DuckDuckGo
The search engine that doesn't track you. Help Spread DuckDuckGo!

L’architecture jsPsych

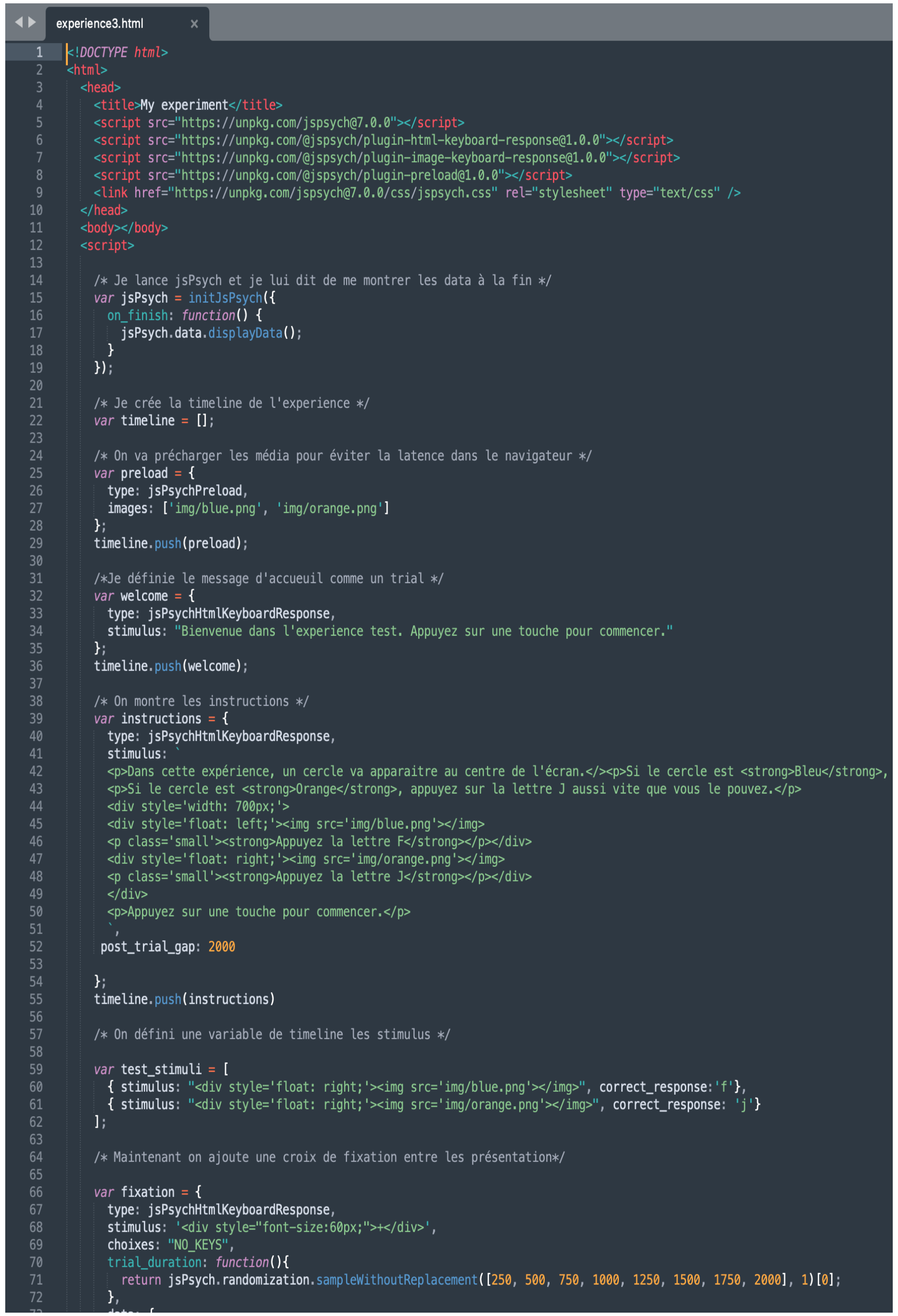


L'architecture jsPsych



jsPsych

- Présentation de la syntaxe



experience3.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My experiment</title>
    <script src="https://unpkg.com/jspsych@7.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-image-keyboard-response@1.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-preload@1.0.0"></script>
    <link href="https://unpkg.com/jspsych@7.0.0/css/jspsych.css" rel="stylesheet" type="text/css" />
  </head>
  <body></body>
  <script>

    /* Je lance jsPsych et je lui dit de me montrer les data à la fin */
    var jsPsych = initJsPsych({
      on_finish: function() {
        jsPsych.data.displayData();
      }
    });

    /* Je crée la timeline de l'experience */
    var timeline = [];

    /* On va précharger les média pour éviter la latence dans le navigateur */
    var preload = {
      type: jsPsychPreload,
      images: ['img/blue.png', 'img/orange.png']
    };
    timeline.push(preload);

    /*Je définie le message d'accueuil comme un trial */
    var welcome = {
      type: jsPsychHtmlKeyboardResponse,
      stimulus: "Bienvenue dans l'experience test. Appuyez sur une touche pour commencer."
    };
    timeline.push(welcome);

    /* On montre les instructions */
    var instructions = {
      type: jsPsychHtmlKeyboardResponse,
      stimulus: `
      <p>Dans cette expérience, un cercle va apparaitre au centre de l'écran.</><p>Si le cercle est <strong>Bleu</strong>,
      <p>Si le cercle est <strong>Orange</strong>, appuyez sur la lettre J aussi vite que vous le pouvez.</p>
      <div style='width: 700px;'>
      <div style='float: left;'><img src='img/blue.png'></img>
      <p class='small'><strong>Appuyez la lettre F</strong></p></div>
      <div style='float: right;'><img src='img/orange.png'></img>
      <p class='small'><strong>Appuyez la lettre J</strong></p></div>
      </div>
      <p>Appuyez sur une touche pour commencer.</p>
      `,
     post_trial_gap: 2000

    };
    timeline.push(instructions)

    /* On défini une variable de timeline les stimulus */

    var test_stimuli = [
      { stimulus: "<div style='float: right;'><img src='img/blue.png'></img>", correct_response:'f'},
      { stimulus: "<div style='float: right;'><img src='img/orange.png'></img>", correct_response: 'j'}
    ];

    /* Maintenant on ajoute une croix de fixation entre les présentation*/

    var fixation = {
      type: jsPsychHtmlKeyboardResponse,
      stimulus: '<div style="font-size:60px;">+</div>',
      choixes: "NO_KEYS",
      trial_duration: function(){
        return jsPsych.randomization.sampleWithoutReplacement([250, 500, 750, 1000, 1250, 1500, 1750, 2000], 1)[0];
      },
```

jsPsych - syntaxe

Sublime text, Visual studio code, …

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;









jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;
```

Experiment code to type here

```
jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;
```

Trial 1

```
jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;
```

Trial 1

Trial 2

```
jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;
```

Trial 1

Trial 2

…

Trial n

```
jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;
```

Trial 1

```
jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;

var trial1;

jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;

var trial1 = {
   type:     jsPsychPlugin
   param1:
   param2:
   param3:
   …
};



jsPsych.run(timeline) ;
```

jsPsych - syntaxe

jsPsychPlugin.js

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;

var trial1 = {
   type:     jsPsychPlugin
   param1:
   param2:
   param3:
   …
};

jsPsych.run(timeline) ;
```

jsPsych - syntaxe

jsPsychPlugin.js

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;

var trial1 = {
  type:    jsPsychPlugin
  param1:
  param2:
  param3:
  …
};

timeline.push(trial1);


jsPsych.run(timeline) ;
```

jsPsych - syntaxe

Petit détour par la timeline :

Array = [];

jsPsych - syntaxe

Petit détour par la timeline :

```
Array = [];
Array = [a, b, c, d, e, f];
```

jsPsych - syntaxe

Petit détour par la timeline :

Array = [];

Array = [a, b, c, d, e, f];

Array.*push*(**G**);   *méthode <u>.push()</u>*

jsPsych - syntaxe

Petit détour par la timeline :

Array = [];

Array = [a, b, c, d, e, f];

Array.*push*(**G**); *méthode <u>.push()</u>*

Array = [a, b, c, d, e, f, **G**]

jsPsych - syntaxe

Petit détour par la timeline :

Array = [];

Array = [a, b, c, d, e, f];

Attention : les indices commencent à 0 !

Array.*push*(**G**);

méthode .push()

Array = [a, b, c, d, e, f, **G**]

jsPsych - syntaxe

Petit détour par la timeline :

Array = [];

Array = [a, b, c, d, e, f];

Attention : les indices commencent à 0 !

Array.*push*(**G**);

méthode <u>.push()</u>

```
Array = [a, b, c, d, e, f, G]
         /  /  /  /  /  /  /
         0  1  2  3  4  5  6
```

jsPsych - syntaxe

Petit détour par la timeline :

Array = [];

jsPsych - syntaxe

Petit détour par la timeline :

```
timeline = [];
```

jsPsych - syntaxe

Petit détour par la timeline :

timeline = [];

timeline = [*trial 1, trial2, trial3, …*];

jsPsych - syntaxe

Petit détour par la timeline :

timeline = [];

timeline = [*trial 1, trial2, trial3, …*];

timeline.push(trialN);

jsPsych - syntaxe

Petit détour par la timeline :

timeline = [];

timeline = [*trial 1,* [trial2A, trial2B, …]*, trial3, …*];

timeline.push(trialN);

jsPsych - syntaxe

Petit détour par la timeline :

timeline = [];

timeline = [*trial 1,* [trial2A, trial2B, [...], ...]*, trial3, ...*];

timeline.push(trialN);

jsPsych - syntaxe

Petit détour par la timeline :

timeline = [];

$$\text{timeline = [}\textit{trial 1,}\text{ }\underbrace{\text{[trial2A, trial2B, }\underbrace{\text{[…]}}\text{, …]}}\textit{, trial3, …}\text{];}$$

timeline.push(trialN);

jsPsych - syntaxe

<script>

var jsPsych = initJsPsych({}) ;

var timeline = [] ;

Trial 1

Trial 2

…

Trial n

jsPsych.run(timeline) ;

jsPsych - syntaxe

<script>

var jsPsych = initJsPsych({}) ;

var timeline = [] ;

Trial 1

Timeline.push

Trial 2

Timeline.push

Trial n

Timeline.push

jsPsych.run(timeline) ;

jsPsych - syntaxe

jsPsychPlugin.js

```
var jsPsych = initJsPsych({}) ;                                                    <script>

var timeline = [] ;

var trial1 = {
   type:    jsPsychPlugin
   param1:
   param2:
   param3:
   …
};

timeline.push(trial1);


jsPsych.run(timeline) ;
```

jsPsych

- Paramètres communs des plugins :

- *on_start, on_load, on_finish*
- *data*
- *post_trial_gap*
- *css_classes*

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;

var trial1 = {
   type:     jsPsychPlugin
   param1:
   …
};

Timeline.push

jsPsych.run(timeline) ;
```

jsPsych - syntaxe

<script>

```
var jsPsych = initJsPsych({}) ;

var timeline = [] ;
```

on_start:

on_load:

```
var trial1 = {
   type:     jsPsychPlugin
   param1:
   …
};
```

on_finish:

Timeline.push

```
jsPsych.run(timeline) ;
```

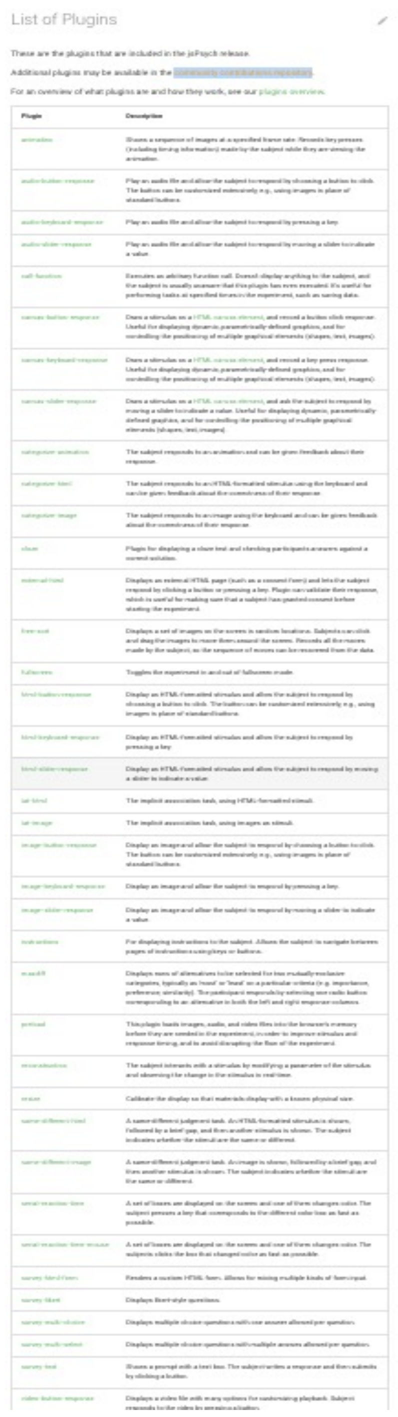
jsPsych - syntaxe

```
var jsPsych = initJsPsych({}) ;                                   <script>

var timeline = [] ;
on_start:
on_load:
var trial1 = {
   type:     jsPsychPlugin
   param1:
   …
};
on_finish:
Post_trial_gap: (ITI)
                    Timeline.push
jsPsych.run(timeline) ;
```

jsPsych

- Présentation des principaux plugins



List of Plugins

These are the plugins that are included in the jsPsych release.

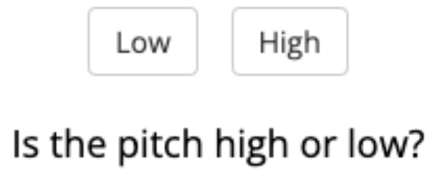
Additional plugins may be available in the community contributions repository.

For an overview of what plugins are and how they work, see our plugins overview.

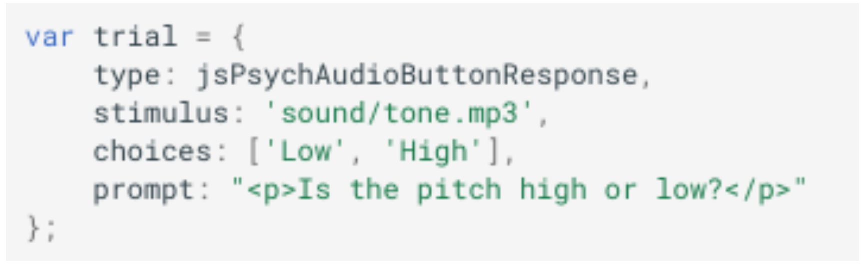
Plugin	Description
animation	Shows a sequence of images at a specified frame rate. Records key presses (including timing information) made by the subject while they are viewing the animation.
audio-button-response	Play an audio file and allow the subject to respond by choosing a button to click. The button can be customized extensively, e.g., using images in place of standard buttons.
audio-keyboard-response	Play an audio file and allow the subject to respond by pressing a key.
audio-slider-response	Play an audio file and allow the subject to respond by moving a slider to indicate a value.
call-function	Executes an arbitrary function call. Doesn't display anything to the subject, and the subject is usually unaware that this plugin has even executed. It's useful for performing tasks at specified times in the experiment, such as saving data.
canvas-button-response	Draw a stimulus on a HTML canvas element, and record a button click response. Useful for displaying dynamic, parametrically-defined graphics, and for controlling the positioning of multiple graphical elements (shapes, text, images).
canvas-keyboard-response	Draw a stimulus on a HTML canvas element, and record a key press response. Useful for displaying dynamic, parametrically-defined graphics, and for controlling the positioning of multiple graphical elements (shapes, text, images).
canvas-slider-response	Draw a stimulus on a HTML canvas element, and ask the subject to respond by moving a slider to indicate a value. Useful for displaying dynamic, parametrically defined graphics, and for controlling the positioning of multiple graphical elements (shapes, text, images).
categorize-animation	The subject responds to an animation and can be given feedback about their response.
categorize-html	The subject responds to an HTML-formatted stimulus using the keyboard and can be given feedback about the correctness of their response.
categorize-image	The subject responds to an image using the keyboard and can be given feedback about the correctness of their response.
cloze	Plugin for displaying a cloze test and checking participants answers against a correct solution.
external-html	Displays an external HTML page (such as a consent form) and lets the subject respond by clicking a button or pressing a key. Plugin can validate their response, which is useful for making sure that a subject has granted consent before starting the experiment.
free-sort	Displays a set of images on the screen in random locations. Subjects can click and drag the images to move them around the screen. Records all the moves made by the subject, so the sequence of moves can be recovered from the data.
fullscreen	Toggles the experiment in and out of fullscreen mode.
html-button-response	Display an HTML-formatted stimulus and allow the subject to respond by choosing a button to click. The button can be customized extensively, e.g., using images in place of standard buttons.
html-keyboard-response	Display an HTML-formatted stimulus and allow the subject to respond by pressing a key.
html-slider-response	Display an HTML-formatted stimulus and allow the subject to respond by moving a slider to indicate a value.
iat-html	The implicit association task, using HTML-formatted stimuli.
iat-image	The implicit association task, using images as stimuli.
image-button-response	Display an image and allow the subject to respond by choosing a button to click. The button can be customized extensively, e.g., using images in place of standard buttons.
image-keyboard-response	Display an image and allow the subject to respond by pressing a key.
image-slider-response	Display an image and allow the subject to respond by moving a slider to indicate a value.
instructions	For displaying instructions to the subject. Allows the subject to navigate between pages of instructions using keys or buttons.
maxdiff	Displays rows of alternatives to be selected for two mutually-exclusive categories, typically as 'most' or 'least' on a particular criteria (e.g. importance, preference, similarity). The participant responds by selecting one radio button corresponding to an alternative in both the left and right response columns.
preload	This plugin loads images, audio, and video files into the browser's memory before they are needed in the experiment, in order to improve stimulus and response timing, and to avoid disrupting the flow of the experiment.
reconstruction	The subject interacts with a stimulus by modifying a parameter of the stimulus and observing the change in the stimulus in real-time.
resize	Calibrate the display so that materials display with a known physical size.
same-different-html	A same-different judgment task. An HTML-formatted stimulus is shown, followed by a brief gap, and then another stimulus is shown. The subject indicates whether the stimuli are the same or different.
same-different-image	A same-different judgment task. An image is shown, followed by a brief gap, and then another stimulus is shown. The subject indicates whether the stimuli are the same or different.
serial-reaction-time	A set of boxes are displayed on the screen and one of them changes color. The subject presses a key that corresponds to the different color box as fast as possible.
serial-reaction-time-mouse	A set of boxes are displayed on the screen and one of them changes color. The subjects clicks the box that changed color as fast as possible.
survey-html-form	Renders a custom HTML form. Allows for mixing multiple kinds of form input.
survey-likert	Displays likert-style questions.
survey-multi-choice	Displays multiple choice questions with one answer allowed per question.
survey-multi-select	Displays multiple choice questions with multiple answers allowed per question.
survey-text	Shows a prompt with a text box. The subject writes a response and then submits by clicking a button.
video-button-response	Displays a video file with many options for customizing playback. Subject responds to the video by pressing a button.

jsPsych

- 1) Boutons:



Is the pitch high or low?

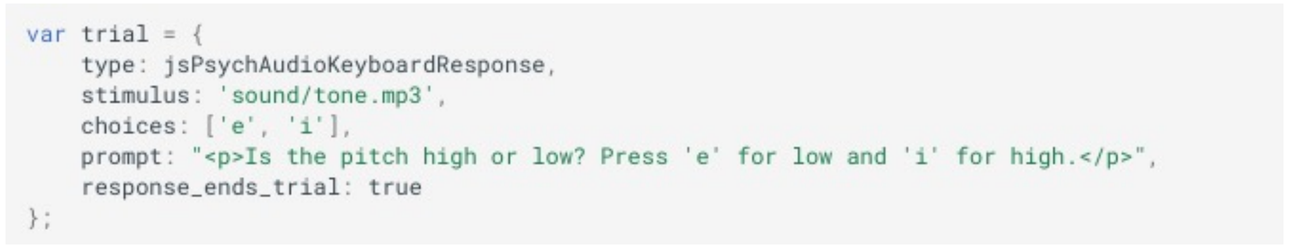


```
var trial = {
    type: jsPsychAudioButtonResponse,
    stimulus: 'sound/tone.mp3',
    choices: ['Low', 'High'],
    prompt: "<p>Is the pitch high or low?</p>"
};
```

jsPsych

- 2) Keyboard:

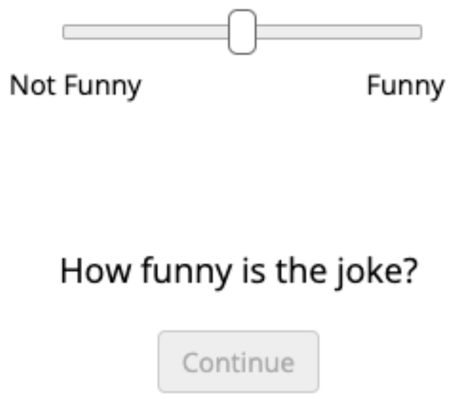
Is the pitch high or low? Press 'e' for low and 'i' for high.

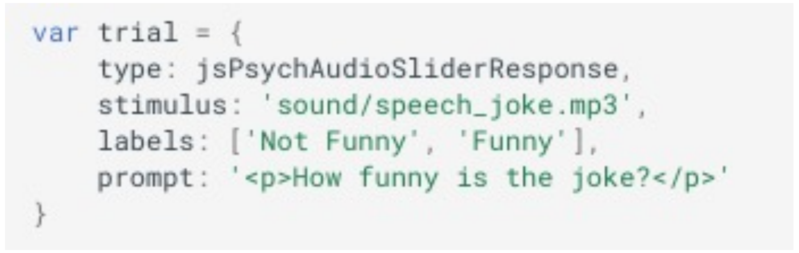


```
var trial = {
    type: jsPsychAudioKeyboardResponse,
    stimulus: 'sound/tone.mp3',
    choices: ['e', 'i'],
    prompt: "<p>Is the pitch high or low? Press 'e' for low and 'i' for high.</p>",
    response_ends_trial: true
};
```

jsPsych

- 3) Intéractions:



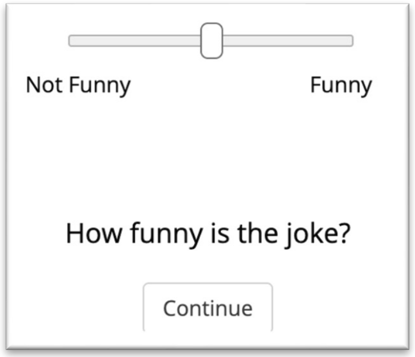


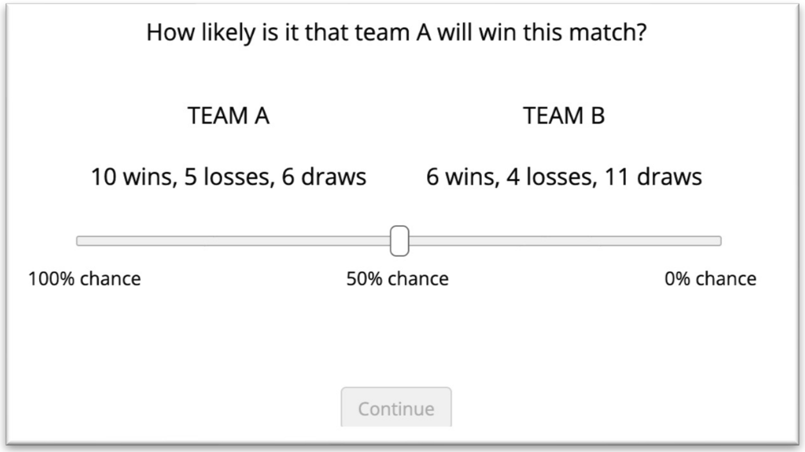
```
var trial = {
    type: jsPsychAudioSliderResponse,
    stimulus: 'sound/speech_joke.mp3',
    labels: ['Not Funny', 'Funny'],
    prompt: '<p>How funny is the joke?</p>'
}
```

jsPsych

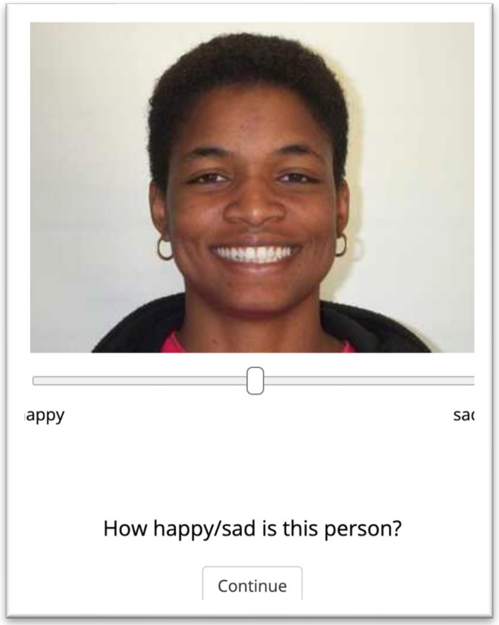
Ces grandes catégories fonctionnent avec les trois médias :

audio, vidéo, image




How likely is it that team A will win this match?

TEAM A	TEAM B
10 wins, 5 losses, 6 draws	6 wins, 4 losses, 11 draws



jsPsych

- Les questionnaires:

Likert,
Multi-choice,
Multi-select,
Free-text,
Drop-down

html

jsPsych

- D’autres plugins importants:

Preload,

Browser check,

Full-screen,

Instructions,

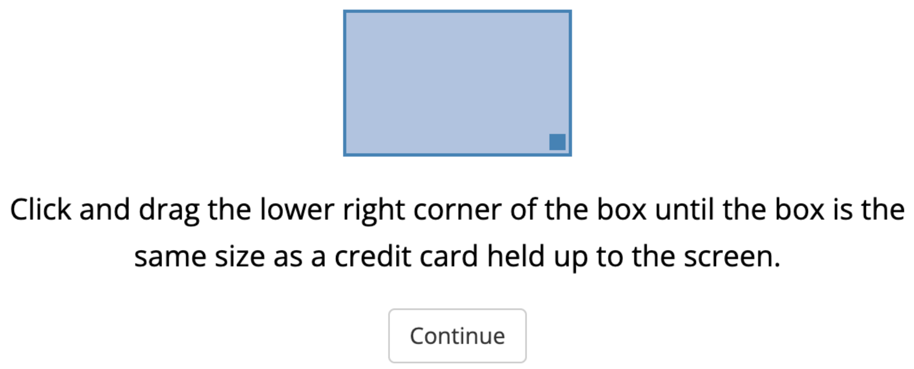
Call function,

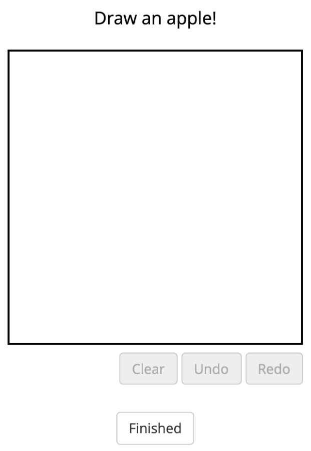
jsPsych

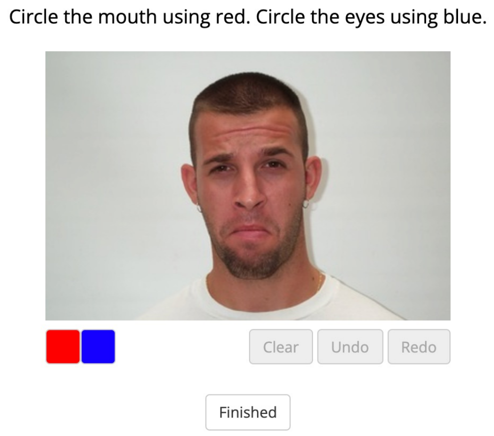
- D’autres plugins:

Initialize-microphone

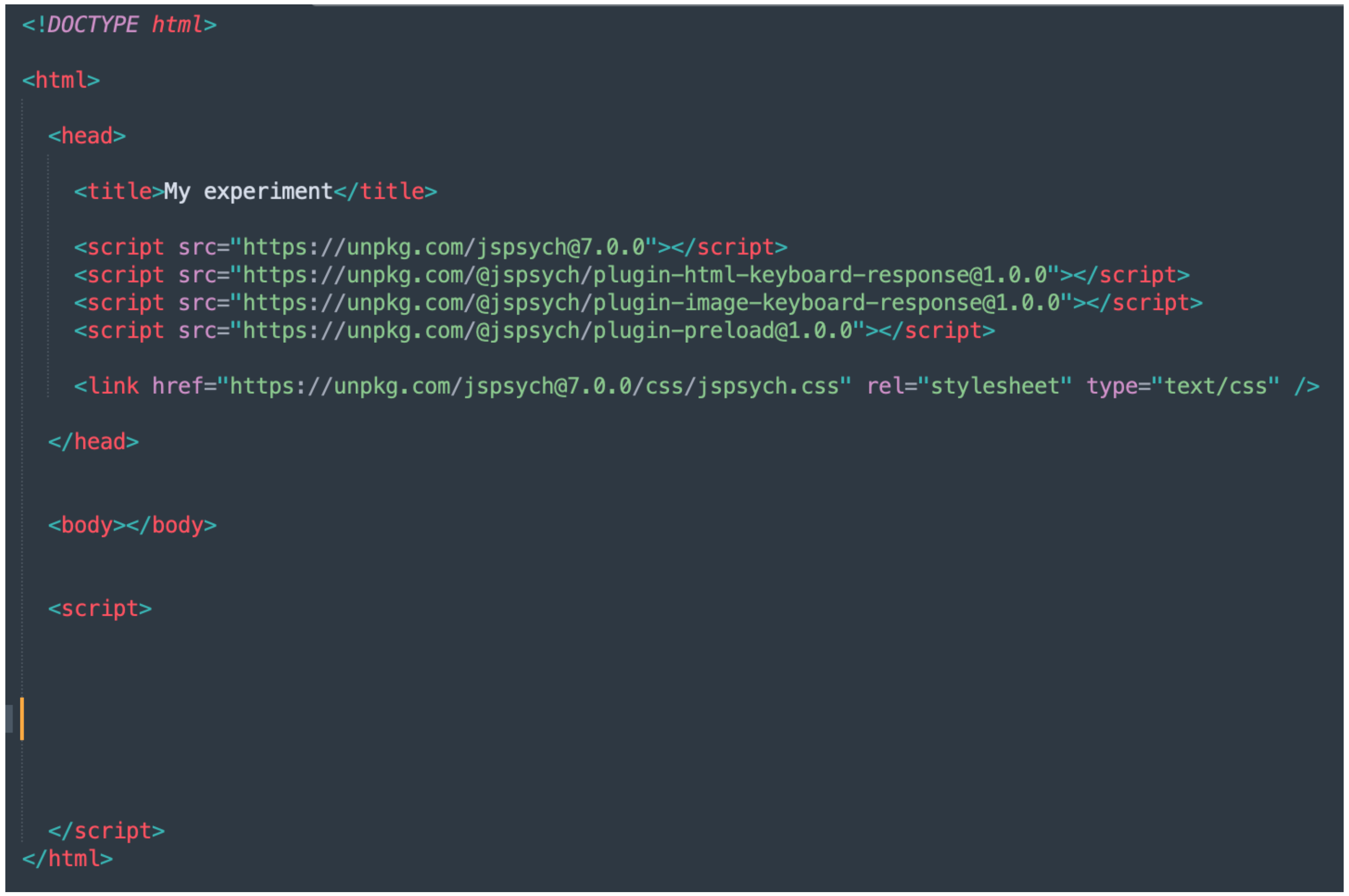
Resize, *Sketchpad,* *…*


Click and drag the lower right corner of the box until the box is the same size as a credit card held up to the screen.


Draw an apple!


Circle the mouth using red. Circle the eyes using blue.

L’architecture web autour



```html
<!DOCTYPE html>

<html>

  <head>

    <title>My experiment</title>

    <script src="https://unpkg.com/jspsych@7.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-image-keyboard-response@1.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-preload@1.0.0"></script>

    <link href="https://unpkg.com/jspsych@7.0.0/css/jspsych.css" rel="stylesheet" type="text/css" />

  </head>


  <body></body>


  <script>







  </script>
</html>
```

Utiliser cognition.run

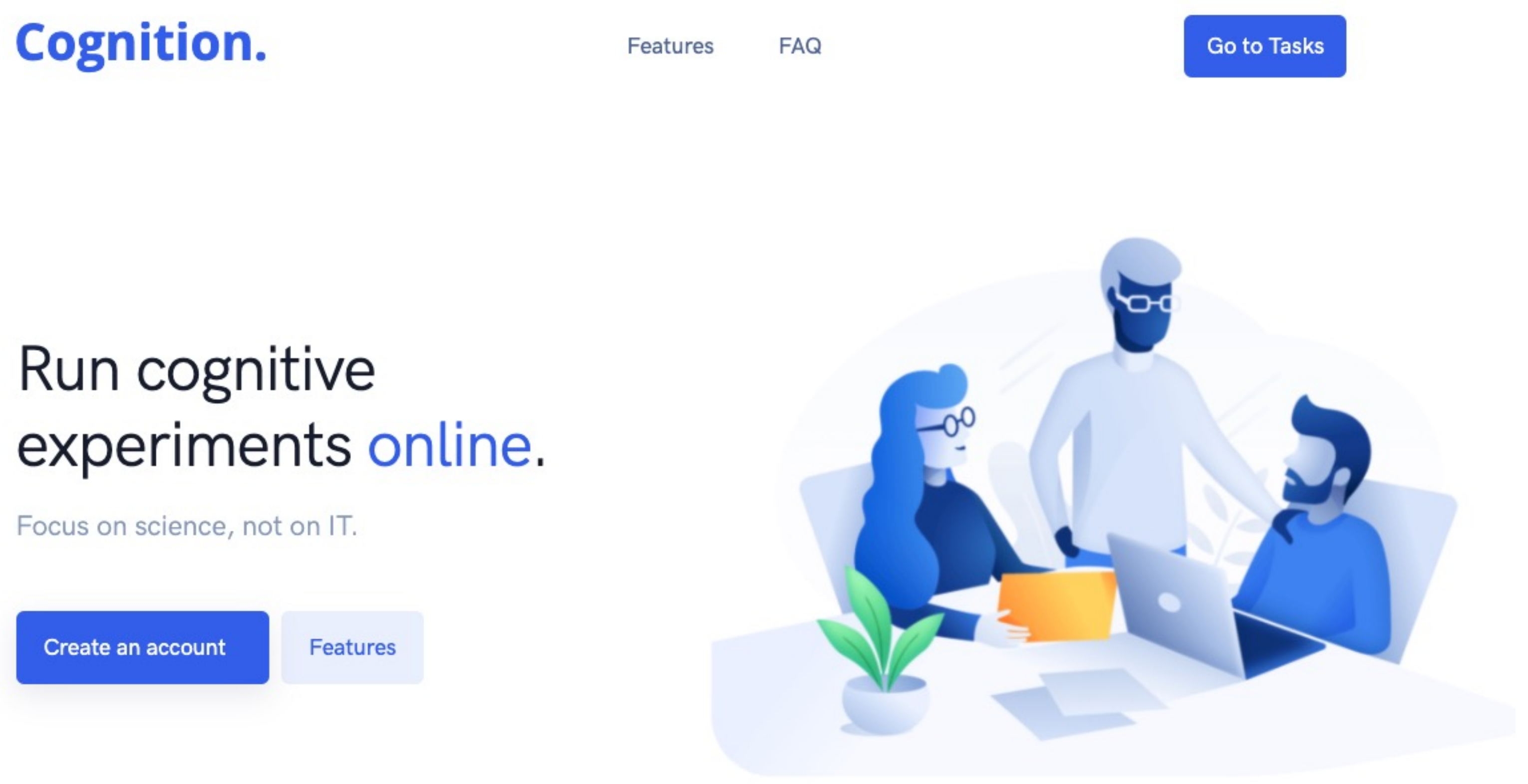
- Cognition permet dans son outil de prévisualisation de ne se soucier que de cette partie <script>.

```
<script>



</script>
```

Cognition.run



Cognition.

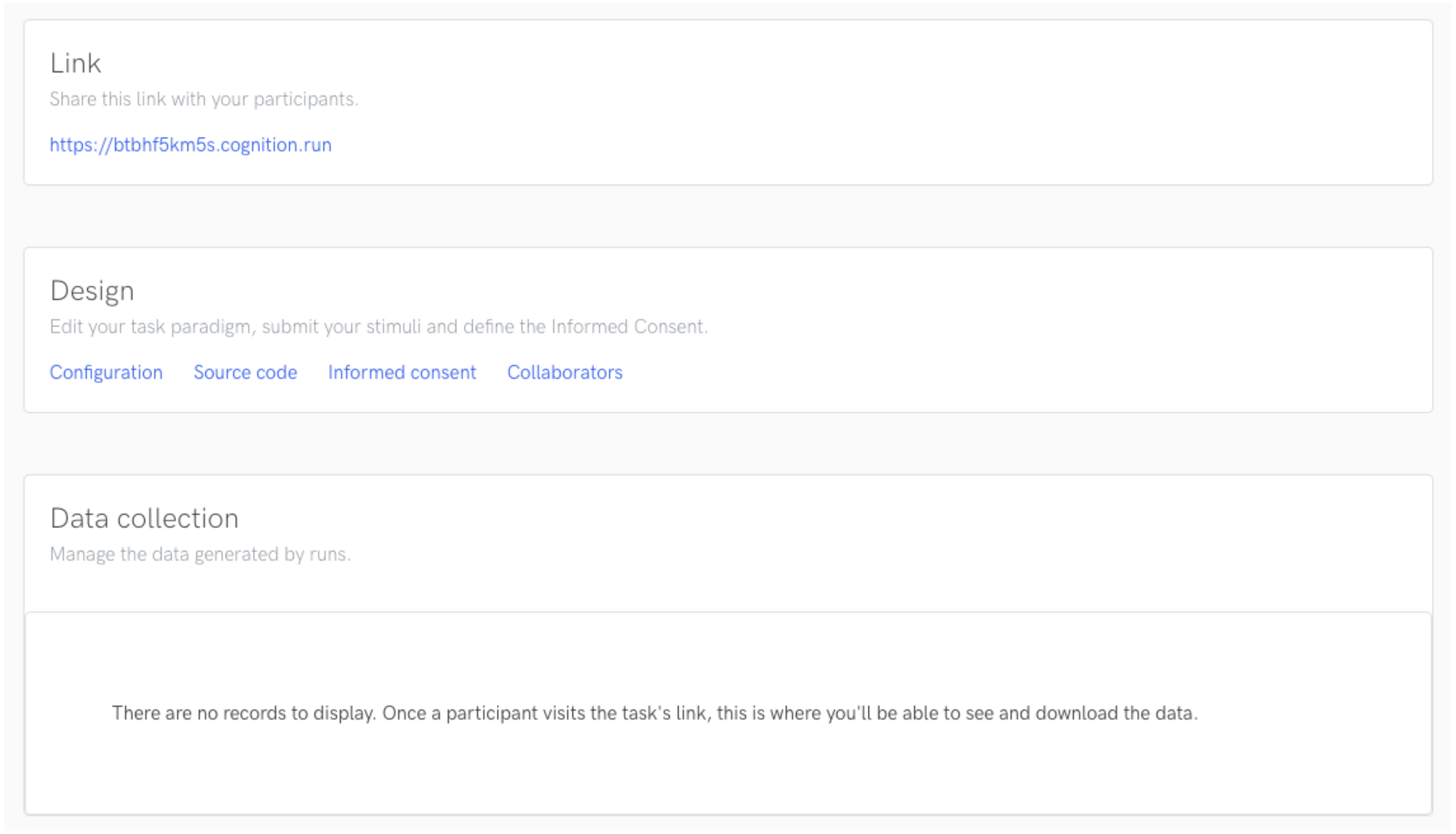
Features FAQ Go to Tasks

Run cognitive experiments online.

Focus on science, not on IT.

Create an account Features

Cognition.run


Link

Share this link with your participants.

https://btbhf5km5s.cognition.run

Design

Edit your task paradigm, submit your stimuli and define the Informed Consent.

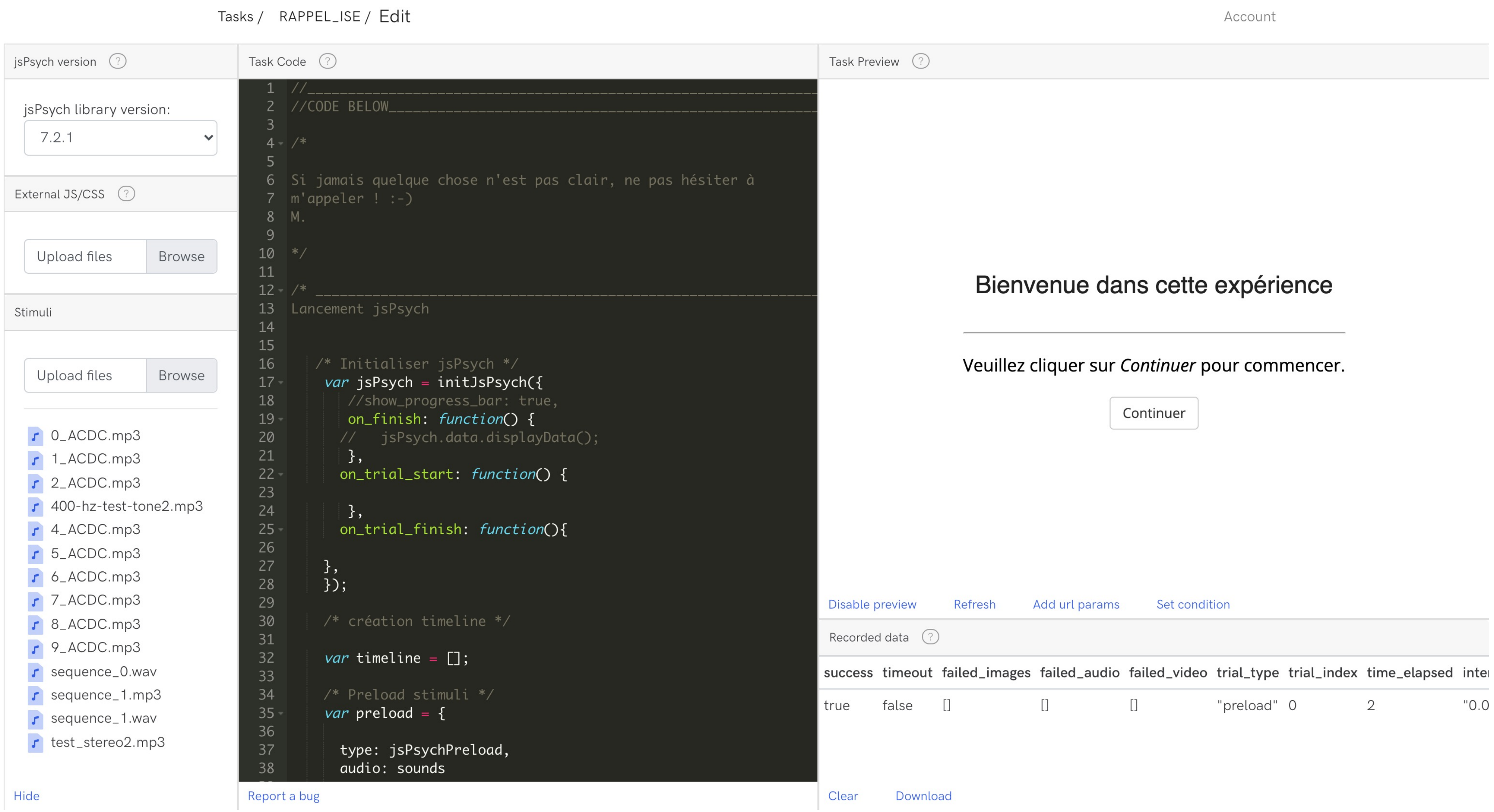
Configuration Source code Informed consent Collaborators

Data collection

Manage the data generated by runs.

There are no records to display. Once a participant visits the task's link, this is where you'll be able to see and download the data.

Cognition.run



Tasks / RAPPEL_ISE / Edit

Account

jsPsych version

jsPsych library version: 7.2.1

External JS/CSS

Upload files | Browse

Stimuli

Upload files | Browse

- 0_ACDC.mp3
- 1_ACDC.mp3
- 2_ACDC.mp3
- 400-hz-test-tone2.mp3
- 4_ACDC.mp3
- 5_ACDC.mp3
- 6_ACDC.mp3
- 7_ACDC.mp3
- 8_ACDC.mp3
- 9_ACDC.mp3
- sequence_0.wav
- sequence_1.mp3
- sequence_1.wav
- test_stereo2.mp3

Hide

Task Code

```
//________________________________________________________________
//CODE BELOW______________________________________________________

/*

Si jamais quelque chose n'est pas clair, ne pas hésiter à
m'appeler ! :-)
M.

*/

/* ______________________________________________________________
Lancement jsPsych


   /* Initialiser jsPsych */
    var jsPsych = initJsPsych({
       //show_progress_bar: true,
       on_finish: function() {
      //   jsPsych.data.displayData();
       },
      on_trial_start: function() {

       },
      on_trial_finish: function(){

    },
    });

    /* création timeline */

    var timeline = [];

    /* Preload stimuli */
    var preload = {

      type: jsPsychPreload,
      audio: sounds
```

Report a bug

Task Preview

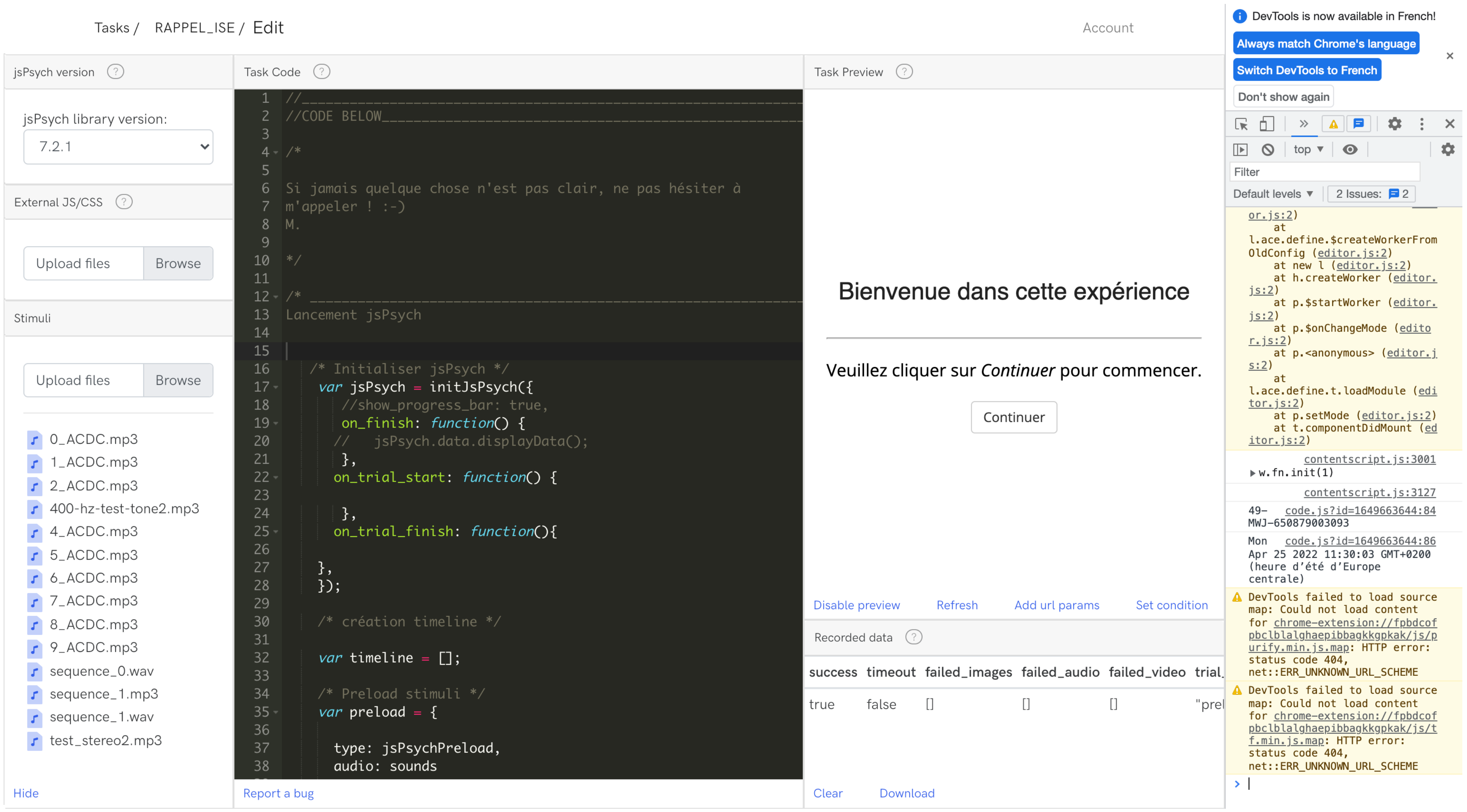
Bienvenue dans cette expérience

Veuillez cliquer sur *Continuer* pour commencer.

Recorded data

success	timeout	failed_images	failed_audio	failed_video	trial_type	trial_index	time_elapsed	inte
true	false	[]	[]	[]	"preload"	0	2	"0.0

Cognition.run



Tasks / RAPPEL_ISE / Edit

Account

jsPsych version

jsPsych library version: 7.2.1

External JS/CSS

Upload files | Browse

Stimuli

Upload files | Browse

- 0_ACDC.mp3
- 1_ACDC.mp3
- 2_ACDC.mp3
- 400-hz-test-tone2.mp3
- 4_ACDC.mp3
- 5_ACDC.mp3
- 6_ACDC.mp3
- 7_ACDC.mp3
- 8_ACDC.mp3
- 9_ACDC.mp3
- sequence_0.wav
- sequence_1.mp3
- sequence_1.wav
- test_stereo2.mp3

Hide

Task Code

```
//______________________________________________________________
//CODE BELOW____________________________________________________

/*

Si jamais quelque chose n'est pas clair, ne pas hésiter à
m'appeler ! :-)
M.

*/

/* ____________________________________________________________
Lancement jsPsych


  /* Initialiser jsPsych */
   var jsPsych = initJsPsych({
     //show_progress_bar: true,
      on_finish: function() {
    //   jsPsych.data.displayData();
      },
     on_trial_start: function() {

      },
     on_trial_finish: function(){

   },
   });

   /* création timeline */

   var timeline = [];

   /* Preload stimuli */
   var preload = {

     type: jsPsychPreload,
     audio: sounds
```

Report a bug

Task Preview

Bienvenue dans cette expérience

Veuillez cliquer sur *Continuer* pour commencer.

Continuer

Disable preview | Refresh | Add url params | Set condition

Recorded data

success	timeout	failed_images	failed_audio	failed_video	trial_
true	false	[]	[]	[]	"prel

Clear | Download

DevTools is now available in French!

Always match Chrome's language | Switch DevTools to French | Don't show again

top

Filter

Default levels ▼ | 2 Issues: 2

```
or.js:2)
    at
l.ace.define.$createWorkerFrom
OldConfig (editor.js:2)
    at new l (editor.js:2)
    at h.createWorker (editor.
js:2)
    at p.$startWorker (editor.
js:2)
    at p.$onChangeMode (edito
r.js:2)
    at p.<anonymous> (editor.j
s:2)
    at
l.ace.define.t.loadModule (edi
tor.js:2)
    at p.setMode (editor.js:2)
    at t.componentDidMount (ed
itor.js:2)
          contentscript.js:3001
 ▶w.fn.init(1)
          contentscript.js:3127
49-    code.js?id=1649663644:84
MWJ-650879003093
Mon    code.js?id=1649663644:86
Apr 25 2022 11:30:03 GMT+0200
(heure d'été d'Europe
centrale)
⚠ DevTools failed to load source
map: Could not load content
for chrome-extension://fpbdcof
pbclblalghaepibbagkkgpkak/js/p
urify.min.js.map: HTTP error:
status code 404,
net::ERR_UNKNOWN_URL_SCHEME
⚠ DevTools failed to load source
map: Could not load content
for chrome-extension://fpbdcof
pbclblalghaepibbagkkgpkak/js/t
f.min.js.map: HTTP error:
status code 404,
net::ERR_UNKNOWN_URL_SCHEME
>
```

Coder une expérience

Exemple de création de code

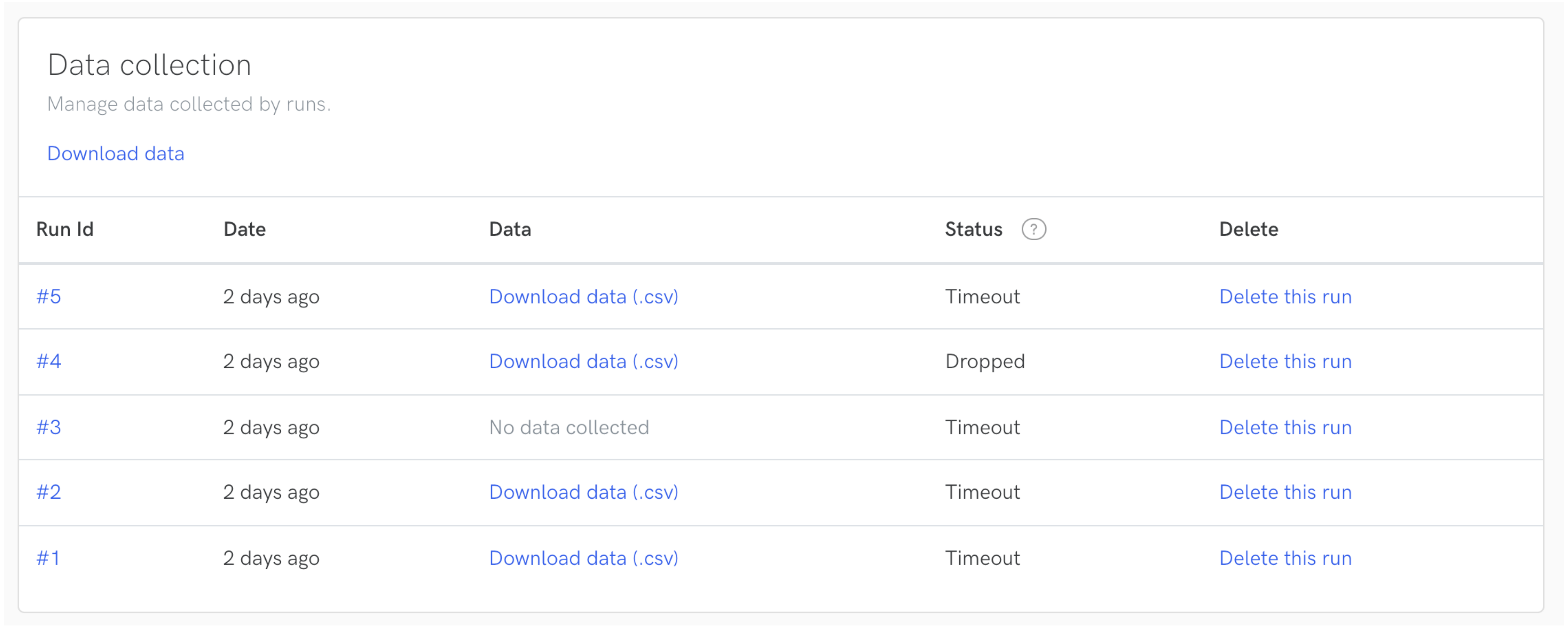
Récupérer ses données

- Deux possibilités :

Récupérer via cognition.run,

Configurer via un serveur.

Récupérer ses données



Data collection

Manage data collected by runs.

Download data

Run Id	Date	Data	Status	Delete
#5	2 days ago	Download data (.csv)	Timeout	Delete this run
#4	2 days ago	Download data (.csv)	Dropped	Delete this run
#3	2 days ago	No data collected	Timeout	Delete this run
#2	2 days ago	Download data (.csv)	Timeout	Delete this run
#1	2 days ago	Download data (.csv)	Timeout	Delete this run