# Stroop实验设计教程

## 1.引入必要的库和资源

-我们需要引入 jsPsych 库以及一些常用的插件,如 html-keyboard-response、image-keyboard-response 和 preload。

-同时需要引入 CSS 文件来设置实验的样式。

\*代码示例：\*

```

<head>

<title>My experiment</title>

<script src="https://unpkg.com/jspsych@7.3.4"></script>

<script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.1.3"></script>

<script src="https://unpkg.com/@jspsych/plugin-image-keyboard-response@1.1.3"></script>

<script src="https://unpkg.com/@jspsych/plugin-preload@1.1.3"></script>

<link href="https://unpkg.com/jspsych@7.3.4/css/jspsych.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" href="layout.css">

</head>

```

## 2.创建实验说明页

-在实验开始前,我们需要向参与者介绍实验的目的和操作流程。

-使用 jsPsychHtmlKeyboardResponse 插件来显示实验说明页,并让参与者按任意键开始实验。

\*代码示例：\*

```

var instructions = {

type: jsPsychHtmlKeyboardResponse,

stimulus: document.querySelector('.instructions-container').outerHTML,

post\_trial\_gap: 1000

};

timeline.push(instructions);

```

## 3.预加载图像

-在实验过程中,我们需要展示一些图像,因此需要先预先加载这些图像。

-使用 jsPsychPreload 插件来预加载所有需要用到的图像。

\*代码示例：\*

```

var preload = {

type: jsPsychPreload,

images: ['img/Red-R.jpg', 'img/Red-G.jpg', 'img/Green-G.jpg', 'img/Green-R.jpg']

};

timeline.push(preload);

```

图片刺激：

![Red-R](https://github.com/user-attachments/assets/d64559f9-ec14-4c66-909c-7b54315f34b2) ![Red-G](https://github.com/user-attachments/assets/0e21678a-5708-4bc9-9aac-5658addd8461) ![Green-R](https://github.com/user-attachments/assets/9190c567-be40-4016-a92d-65c764324e83) ![Green-G](https://github.com/user-attachments/assets/24d74f66-77a8-4df7-ae99-889e2fff4b92)

## 4.定义实验刺激

-实验中要展示的刺激,包括正确反应键和图像路径等,都需要事先定义好。

-我们可以把这些信息组织成一个数组,方便后续使用。

\*代码示例：\*

```

var test\_stimuli = [

{ stimulus: 'img/Red-R.jpg', correct\_response: 'f'},

{ stimulus: 'img/Green-G.jpg', correct\_response: 'j'},

{ stimulus: 'img/Green-R.jpg', correct\_response: 'j'},

{ stimulus: 'img/Red-G.jpg', correct\_response: 'f'},

];

```

## 5.定义注视点和测试试次

-注视点用于吸引参与者的注意力,在每个测试试次之前会显示。

-测试试次使用 jsPsychImageKeyboardResponse 插件来展示图像并记录参与者的反应。

\*代码示例：\*

```

var fixation = {

type: jsPsychHtmlKeyboardResponse,

stimulus: '<div style="font-size:30px; display:flex; justify-content:center; align-items:center; ">+</div>',

choices: "NO\_KEYS",

trial\_duration: function(){

return jsPsych.randomization.sampleWithoutReplacement([250, 500, 750, 1000, 1250, 1500, 1750, 2000], 1)[0];

},

data: {

task: 'fixation'

}

};

var test = {

type: jsPsychImageKeyboardResponse,

stimulus: jsPsych.timelineVariable('stimulus'),

choices: ['f', 'j'],

css\_classes: 'center-image',

data: {

task: 'response',

correct\_response: jsPsych.timelineVariable('correct\_response')

},

on\_finish: function(data){

data.correct = jsPsych.pluginAPI.compareKeys(data.response, data.correct\_response);

}

};

```

## 6.定义测试流程

-将注视点和测试试次组合成一个测试流程,并设置重复次数和随机化顺序。

\*代码示例：\*

```

var test\_procedure = {

timeline: [fixation, test],

timeline\_variables: test\_stimuli,

repetitions: 10,

randomize\_order: true

};

timeline.push(test\_procedure);

```

## 7.定义实验结束页

-在实验结束后,我们需要向参与者反馈他们的表现情况,如正确率和平均反应时。

-使用 jsPsychHtmlKeyboardResponse 插件来显示实验结果。

\*代码示例：\*

```

var debrief\_block = {

type: jsPsychHtmlKeyboardResponse,

stimulus: function() {

var trials = jsPsych.data.get().filter({task: 'response'});

var correct\_trials = trials.filter({correct: true});

var accuracy = Math.round(correct\_trials.count() / trials.count() \* 100);

var rt = Math.round(correct\_trials.select('rt').mean());

return `<p>你的正确率为 ${accuracy}% </p>

<p>你的平均反应时为 ${rt} ms</p>

<p>按任意键结束实验,感谢参与!</p>`;

}

};

timeline.push(debrief\_block);

```

## 8.运行实验

-最后,我们调用 jsPsych.run(timeline) 来运行整个实验流程。

\*代码示例：\*

```

jsPsych.run(timeline);

```

[最终代码](experiment.html)