TUTORIAL

CREAR EL TEST DE STROOP EN PSYCHOPY

1) ¿Qué es el test de Stroop? 1
2) Abrir Psychopy 2
3) "Builder view" de Psychopy. 3
a) Trial 4
b) Componentes 4
c) Flujo 5
4) Test de Stroop 6
a) "Palabras negras" 7
Estímulo 7
Respuesta 10
Código supletorio 11
Insertar loop 12
b) "Rectángulos" 14
c) "Palabras de colores" 20
d) Instrucciones 26
e) Mensaje de agradecimiento 30
Anexo 1 31
Anexo 2 32

1) ¿Qué es el test de Stroop?

El test de Stroop es un fenómeno psicológico en el que la lectura de una palabra con un color de fuente diferente al contenido de la palabra puede retrasar la capacidad de la persona para nombrar el color. Te explicamos con unos ejemplos; lee en voz alta los siguientes nombres de colores:

ROJO - AZUL - VERDE - NARANJA - ROSA

¿Fácil verdad?

Ahora nombra de qué color son los siguientes rectángulos:

También resulta sencillo, ¿verdad?

Bien, ahora nombra en voz alta **en qué color** están escritos los siguientes nombres de colores:

ROJO - AZUL - VERDE - NARANJA - ROSA

¿Verdad que te ha costado un poco más ahora?

Esto es lo que se conoce como el efecto Stroop y se debe a la interferencia cognitiva que se produce entre la acciones de leer una palabra “rojo” y nombrar el color en la que está escrita “azul”.

Esta tarea puede realizarse de manera computarizada, de manera que un ordenador registre el tiempo qué empleamos en responder. De esta manera, se puede calcular de manera precisa el tiempo requerido para nombrar las palabras escritas en negro, para nombrar el color de los rectángulosy para nombrar el color de las palabras escritas en distintos colores.

En este tutorial, os enseñaremos cómo hacerlo con Psychopy, una herramienta de programación visual que se utiliza en los laboratorios para desarrollar todo tipo de tareas de psicología y neurociencia.

2) Abrir Psychopy

Dependiendo de la instalación, podréis abrir Psychopy de una manera o de otra.

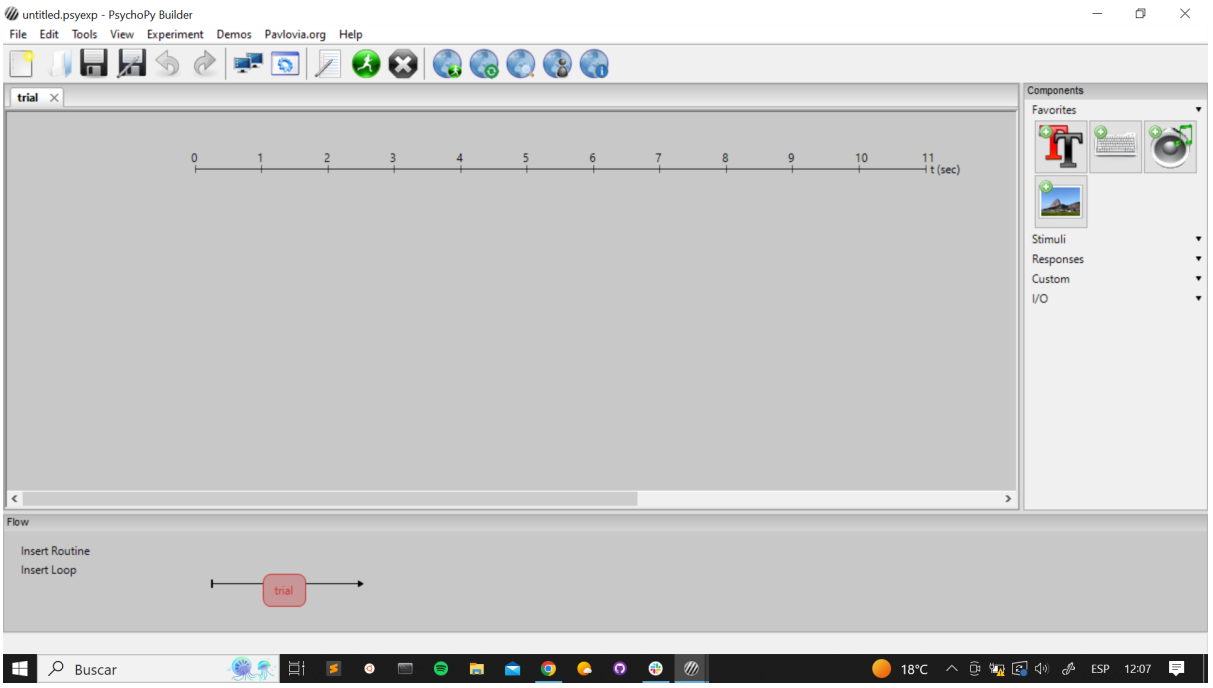
- Si habéis instalado Psychopy utilizando el *"Standalone package"* (recomendado), simplemente tenéis que buscar "Psychopy" en vuestro menú y hacer doble clic en él.

- Si habéis instalado Psychopy a través de *Anaconda*, deberéis:
  - Abrir un terminal de "Anaconda prompt"
  - Entrar en el environment de psychopy escribiendo lo siguiente en el terminal (+ enter)

    conda activate psychopy
  - Abrir psychopy escribiendo lo siguiente en el terminal (+ enter)

    psychopy

Una vez abierto, deberíais ver la siguiente interfaz[1]:



[1] Si en lugar de eso, se os abre vacío, consultad el Anexo 1.

3) “Builder view” de Psychopy.

Nos detendremos un segundo aquí para explicar un poco el funcionamiento de la interfaz. Psychopy se basa en la creación de experimentos con **estructura de rutinas**. Esto significa que, en una rutina, se va repitiendo una misma estructura (trial). En la tarea de Stroop, tendremos tres rutinas.

- Palabras negras
- Rectángulos
- Palabras de colores

Cada una de estas rutinas tendrá a su vez una estructura simple llamada **trial**, que se irá repitiendo una y otra vez. Osea, una rutina consta de “x” trials. Así:

1) En **Palabras negras**, se irá repitiendo la presentación de una palabra, que siempre estará escrita en color negro. En cada trial dentro de la rutina: “Palabras negras”, la palabra escrita en negro será distinta (verde, naranja, lila o azul)

2) En **Rectángulos**, se irá repitiendo la presentación de un rectángulo de un color. En cada trial, el color del rectángulo cambiará (verde, naranja, lila o azul).

3) En **Palabras de colores**, se irá repitiendo la presentación de una palabra. En cada trial, la palabra estará escrita de un color diferente (verde, naranja, lila o azul).

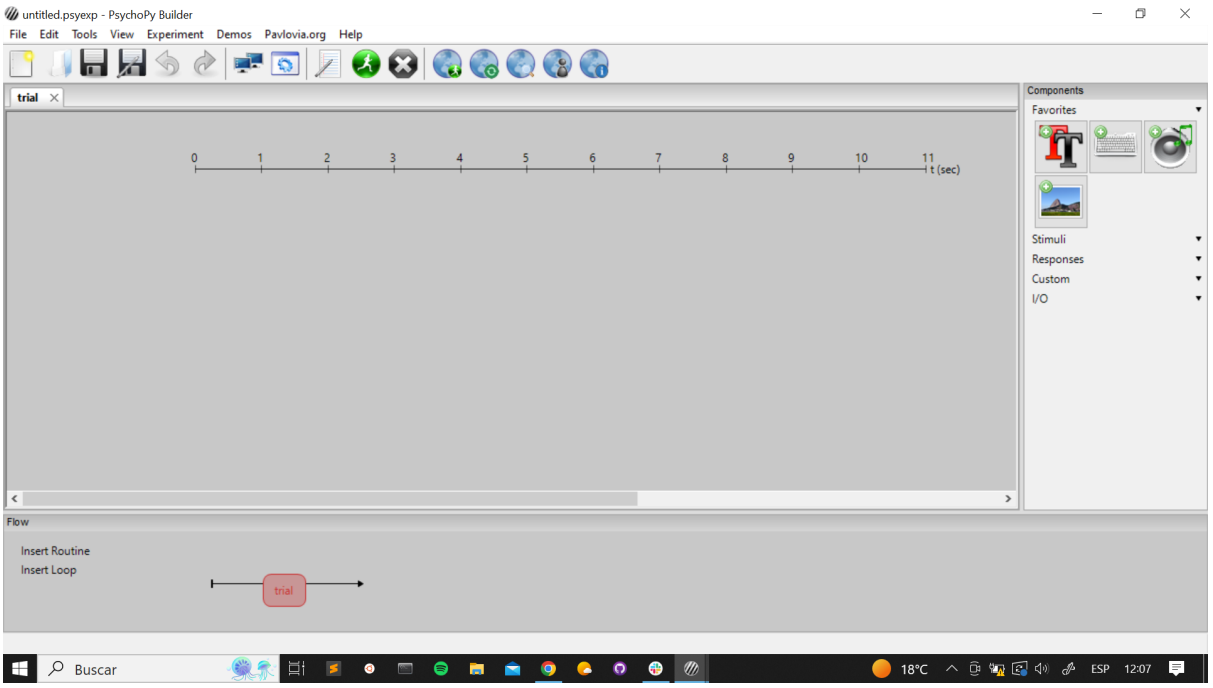
Cada trial de las rutinas explicadas anteriormente terminará cuando la persona responda correctamente a través del teclado, pulsando las teclas “v”, “n”, “l” o “a”. Estas teclas hacen referencia a la primera letra de cada color “v” (verde), “n” (naranja), “l” (lila) y “a” (azul). Se han escogido estos colores porque la posición de los dedos en un teclado QWERTY es cómoda.

Una vez respondan correctamente, se iniciará el siguiente trial y así sucesivamente hasta realizar todos los trials de cada una de las tres rutinas (palabras negras, rectángulos y palabras de colores).

Veamos ahora qué nos encontramos en la pantalla…

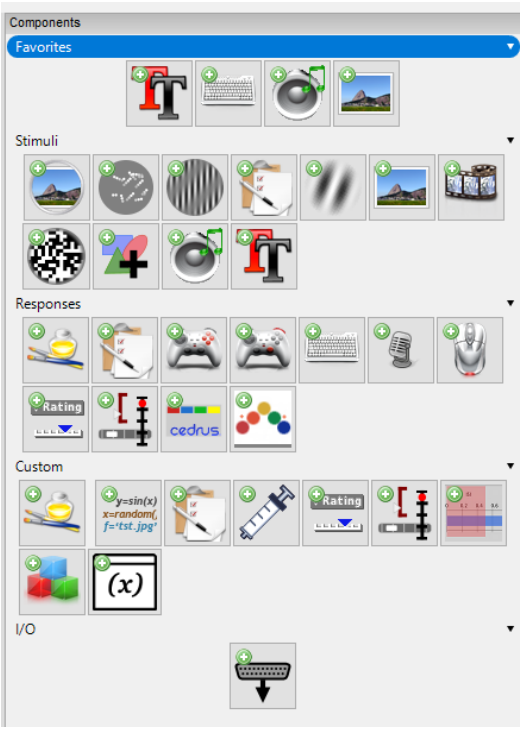
a) Trial

Lo que presenta la pantalla de Psychopy es la línea temporal de un trial.



b) Componentes

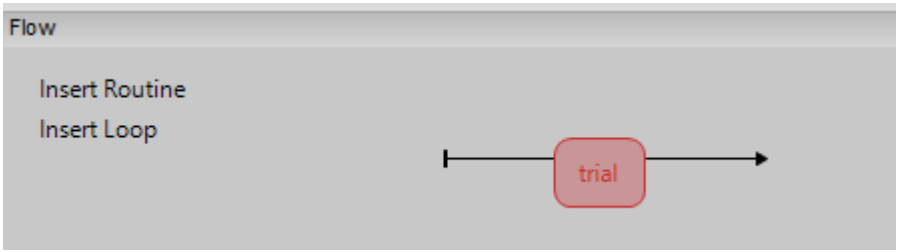
A la derecha, está el apartado de “Componentes”, que hacen referencia a aquellas cosas que pueden introducir en el trial, como por ejemplo imágenes, texto, sonidos, botones de respuesta.



- Hay una sección para los estímulos. En dos de nuestras rutinas (palabras negras y palabras de colores) el estímulo será de **texto**. Mientras que en la rutina de rectángulos, el estímulo será un **polígono**.
- Hay una sección para respuestas. En esta tarea solo utilizaremos las respuestas de teclado para cada rutina.
- Además, hay una sección de personalización. En este experimento añadiremos unas líneas de código adicional, pero ya lo veremos más adelante…

c) Flujo

En la parte inferior, se puede ver el “flujo” (“flow”) de la tarea:



Esta parte es muy importante, pues nos muestra la estructura general del experimento. Con el botón “Insert Rutine” crearemos la estructura básica de un trial en las diferentes rutinas (palabras negras, rectángulos y palabras de colores). Con el botón “Insert Loop” determinaremos cómo se van repitiendo los trials dentro de una rutina.

4) Test de Stroop

El objetivo de este tutorial es tener la siguiente estructura general del experimento:

La estructura será la siguiente:

- **Instrucciones**
  En la imagen tiene el nombre de "*instrucciones*".
  Se explica cómo realizar la tarea de palabras negras.

- **Rutina de palabras negras**
  En la imagen tiene el nombre de "*color_negro*". Se observa el loop, que indica que el trial de esa rutina se va repitiendo.

- **Instrucciones 2**
  En la imagen tiene el nombre de "*instrucciones_2*".
  Se explica cómo realizar la tarea de rectángulos.

- **Rutina de rectángulos**
  En la imagen tiene el nombre de "*color_rectangulo*". Se observa el loop, que indica que el trial de esa rutina se va repitiendo.

- **Instrucciones 3**
  En la imagen tiene el nombre de "*instrucciones_3*".
  Se explica cómo realizar la tarea de palabras de colores.

- **Rutina de palabras de colores.**
  En la imagen tiene el nombre de "*color_variado*". Se observa el loop, que indica que el trial de esa rutina se va repitiendo.

- **Mensaje de agradecimiento**
  En la imagen tiene el nombre de "*gracias*".
  Se le da las gracias al sujeto por participar en el experimento.

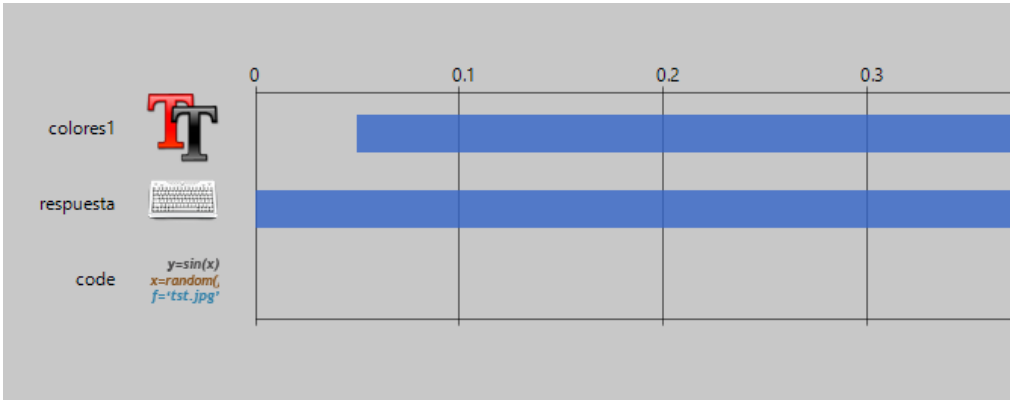
a) ”Palabras negras”

Ahora empezaremos por crear el trial básico de la rutina de “Palabras negras”. Este trail consta de tres componentes:

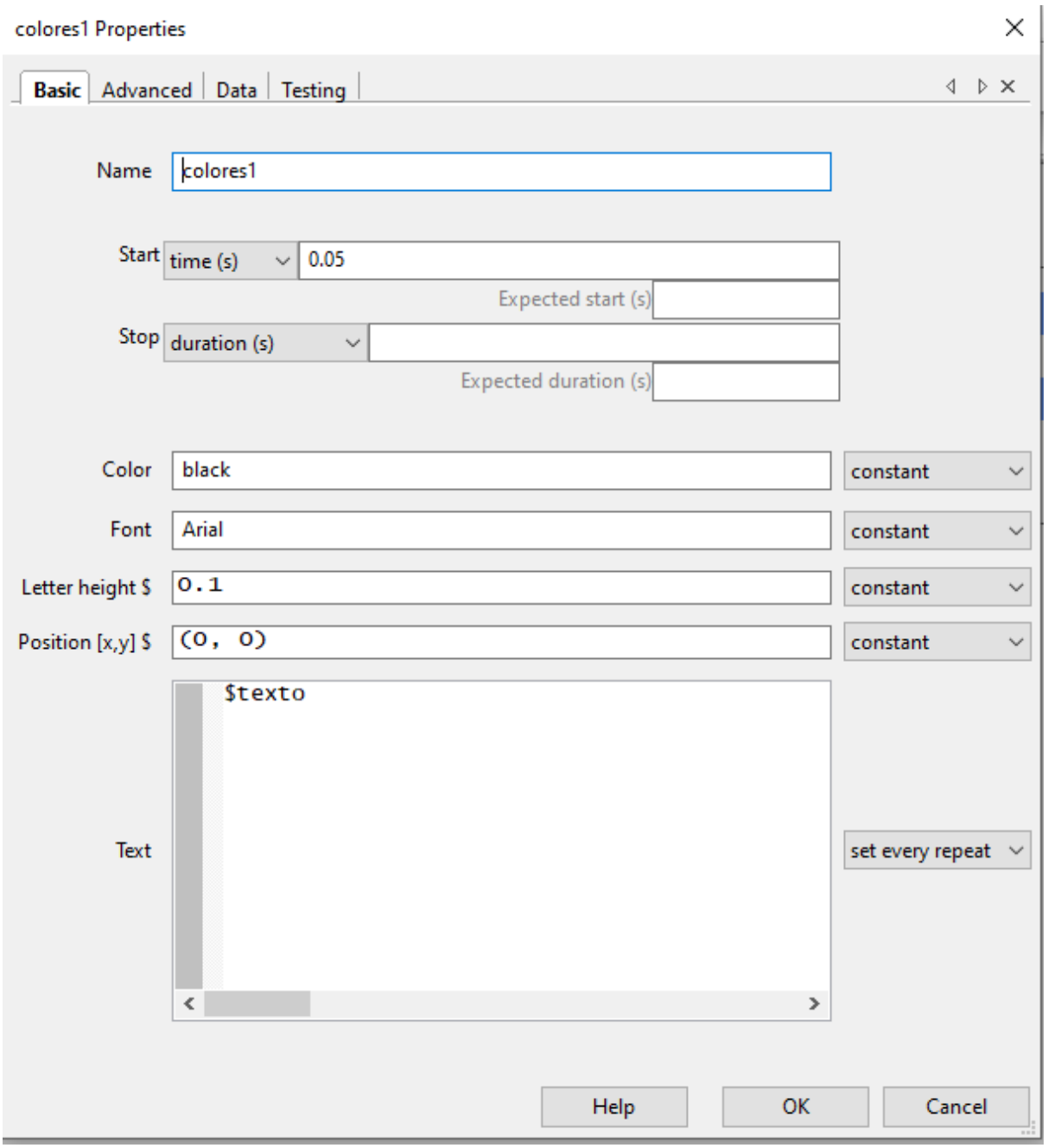
- Estímulo
- Respuesta
- Código supletorio

Estímulo

Con el primero, determinamos lo que los sujetos van a ver aparecer en la pantalla (palabra escrita en negro), mientras que con “Respuesta” y “Código supletorio”, determinaremos cómo se realiza y se guarda la respuesta que den los sujetos.



Para determinar el estímulo, seleccionaremos, de los componentes de Estímulo, el de texto. Aparecerá una ventana que hemos de completar de la siguiente manera:



- En Name, determinamos cómo llamaremos a este componente.
- En Start, determinaremos en qué momento (segundos) aparece. En este caso, no es 0 para permitir un pequeño “blink” entre palabra y palabra cuando incorporamos el Loop a la rutina.
- En Stop, no pondremos nada, pues queremos que la palabra esté presente hasta que el sujeto de una respuesta correcta.
- En Color, ponernos black (la palabra estará escrita en negro)
- En Letter height, que hace referencia al tamaño, ponemos 0.1, y la posición la dejamos en (0,0), que hace referencia al centro de la pantalla.

En el apartado de Text, escribiremos **$texto** y, en la pestaña de la derecha, marcaremos la opción “set every repeat”. Esta parte es fundamental entenderla, porque hace referencia a cómo se van a ir presentando los trials uno detrás de otro.

Lo que hemos escrito en el texto ($texto) combinado con “set every repeat” hace que, en cada trial, el texto escrito vaya cambiando y que en cada trial se presente el nombre de un color distinto.
¿Pero cómo cambian? ¿De dónde sacamos los nombres de las palabras? Pues leyendo un archivo externo (un Excel) que contiene los textos que van a ir apareciendo.

Ahora **crearemos un archivo Excel (.xlsx)** o usaremos el descargado. Este archivo Excel debe tener una una columna llamada texto (fundamental no equivocarse con los nombres, pues el $texto que hemos escrito en el componente de Psychopy lo que hace es leer la columna del archivo excel que lleva ese nombre). Debemos crear también una columna llamada answer.

Completamos el Excel con la información que tenemos en la imagen de la derecha. La columna *texto* determina qué palabra aparece escrita y la columna *answer* hace referencia a qué tecla se debe pulsar para que la respuesta sea correcta.

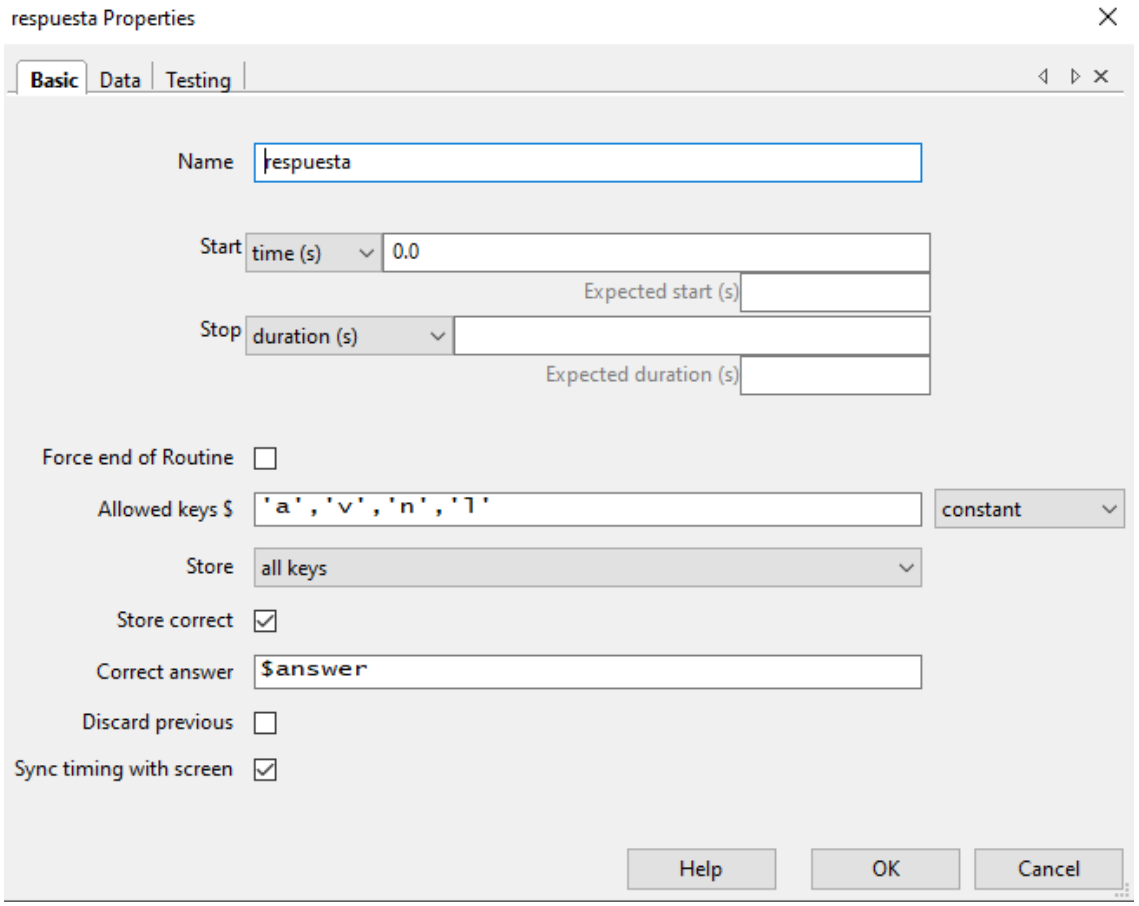
Guardaremos este Excel en la misma carpeta en la que se encuentra la tarea de psychopy con el nombre “colores1_negro.xlsx” (puedes aprovechar y guardar la tarea que estamos creando en *File* → *Save as*…).

Volvamos a Psychopy, porque ahora vamos a crear el componente encargado de recoger las respuestas de los sujetos.

texto	answer
azul	a
azul	a
azul	a
azul	a
azul	a
azul	a
azul	a
azul	a
azul	a
naranja	n
naranja	n
naranja	n
naranja	n
naranja	n
naranja	n
naranja	n
naranja	n
naranja	n
verde	v
verde	v
verde	v
verde	v
verde	v
verde	v
verde	v
verde	v
verde	v
lila	l
lila	l
lila	l
lila	l
lila	l
lila	l
lila	l
lila	l
lila	l

Respuesta

Para la respuesta, seleccionaremos el componente de Psychopy “*keyboard*”, pues la respuesta que deberán dar los participantes será la de pulsar la tecla correspondiente a la primera letra del nombre de la palabra (por ejemplo, si es azul, deberán pulsar la tecla “a” y si es naranja, la letra “n”).
El componente de la respuesta deberá completarse de la siguiente manera:



- En este caso, llamaremos al componente “respuesta” y haremos que empiece a 0.0 y que no termine (no ponemos nada en Stop). La opción “*Force end of Routine*”, que aparecerá marcada, la desmarcamos.
- “*Allowed keys*” hace referencia a las teclas que se aceptan como respuesta. Para este experimento, introducimos la “a” de azul, la “v” de verde, la “n” de naranja y la “l” de lila.
- En la opción *Store* ponemos “*all keys*”. Marcamos “*Store correct*”. En correct answer tenemos que escribir $answer. Este apartado sigue la misma lógica que los textos en el componente de texto. Para que el programa sepa cual es la respuesta correcta, mira en el Excel que hemos creado, esta vez en la columna que hemos llamado “answer”.

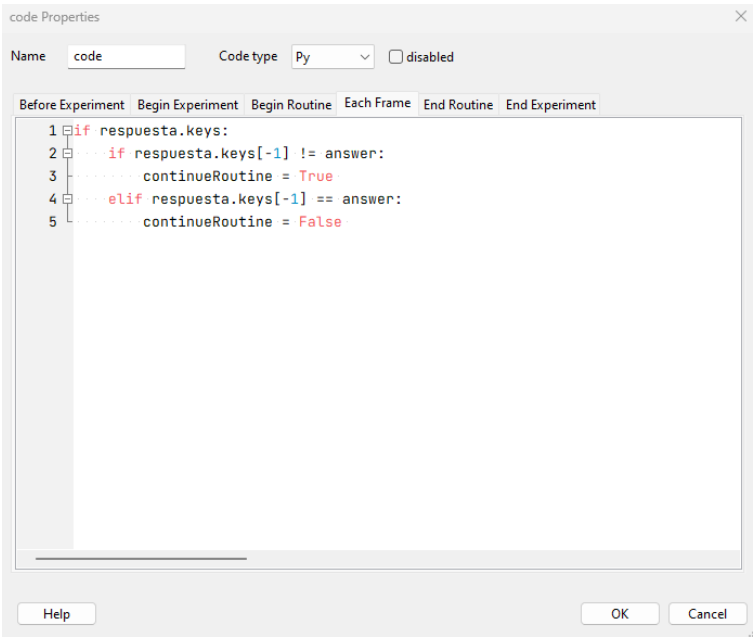
Código supletorio

A veces ocurre que las funciones que nos aporta el "*Builder View*" con su interfaz interactiva son limitadas, y queremos añadir alguna cosa más al experimento. Tal como está el trial que hemos creado hasta ahora, siempre que el participante pulse una de las teclas habilitadas ("a", "v", "n", o "l") pasará al siguiente trial ***independientemente de si la respuesta es correcta o no***. Esto último no nos interesa, pues queremos que el participante **solo pase al siguiente trial tras haber contestado correctamente**. Para habilitar esta función, deberemos añadir unas líneas de código python.

Para hacerlo, añadiremos un componente de "Code".

Iremos a la pestaña que pone "Each Frame" y copiaremos lo siguiente[2]:

```
if respuesta.keys:
    if respuesta.keys[-1] != answer:
        continueRoutine = True
    elif respuesta.keys[-1] == answer:
        continueRoutine = False
```



De esta manera, no se avanzará hasta el siguiente trial hasta que la respuesta sea correcta.

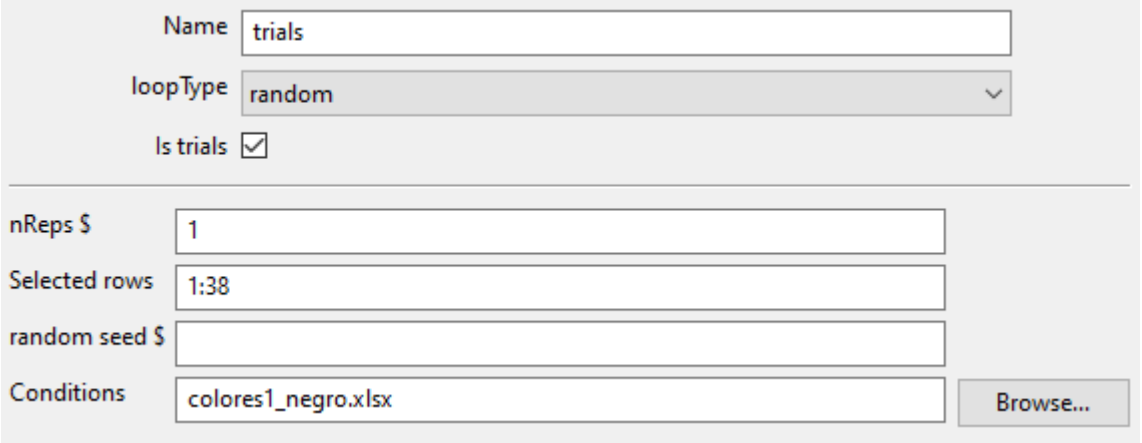
[2] En programación es muy importante ser consistente con la tabulación. Procurad que quede exactamente igual que en la imagen.

Insertar loop

¡Ya tenemos todos los componentes necesarios creados!, ahora falta darle la estructura de rutina y programar la repetición de los trials.

A esto se le llama “Insertar un Loop”, que significa determinar las normas para la repetición del trial que hemos creado. Para hacerlo, **vamos al apartado de flujo** y le damos al botón “*Insert Loop*”. Al hacer eso, nos aparecerá un círculo negro para determinar donde se inicia el loop . Presionaremos antes del cuadrado que hace referencia a la estructura del trial de la rutina que acabamos de crear. Una vez hayamos hecho click, aparecerá una ventana de “*Loop properties*”, que nos servirá para determinar las propiedades de la repetición de los trials.

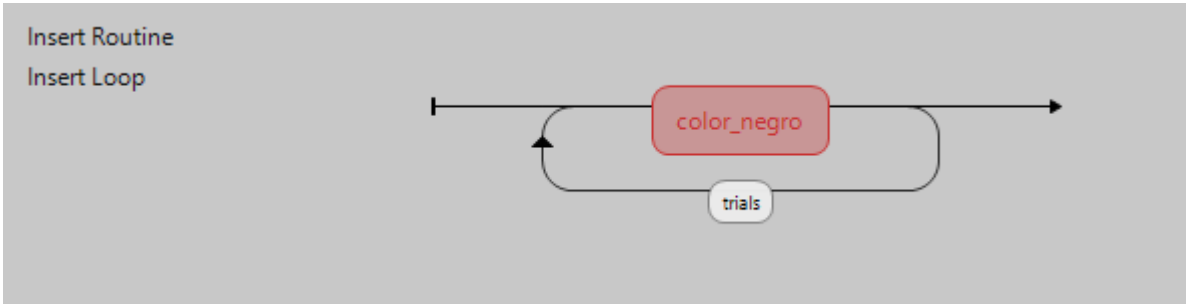
Esta ventana ha de rellenarse con la siguiente información:



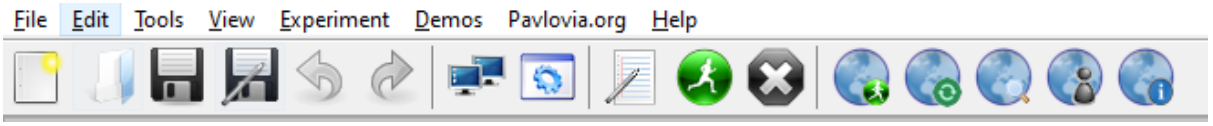
- Le llamaremos trials (Name) y en “*looptype*” seleccionaremos la opción “random”. Esto significa que no va a seguir el orden tal cual está en documento excel, sinó que irá eligiendo los trials al azar hasta hacerlos todos (sin repetirse).
- En “*nReps”*, pondremos 1, pues queremos que solo haga las repeticiones que ponemos en el documento excel, y en “*Selected rows*”, especificaremos las filas del excel que tienen los datos que nos interesan. Como se puede ver en la imagen, hay información hasta la fila 37, por lo que pondremos de 1:38 (si ponemos 1:37, la 37 no la cogemos).
- Finalmente, en “*Conditions”*, especificamos el nombre el archivo que estamos utilizando, que es el que hemos creado antes con el nombre “colores1_negro.xlsx”. En este caso, como hemos guardado el .xlsx en el mismo lugar en el que hemos creado la tarea, no hay que hacer nada[3].

[3] Si el archivo “colores1_negro.xlsx” se encontrase en otra ubicación, habría que poner el path completo de donde se encuentra el archivo. Para hacerlo, se le da a “Browse…” y se selecciona el .xlsx en la ubicación deseada.

Cuando hayamos completado la ventana de "*Loop properties*", deberemos hacer click después de esa rutina para cerrar el loop. Al hacerlo, habremos creado un loop que se inicia antes del trial que hemos creado y se cierra después, como en la siguiente imagen[4]:



¡Ya tenemos un tercio del experimento creado! ¡Es un buen momento para guardar y probar el experimento! Para guardar, hay que presionar en el icono del disco y, para probar el experimento, darle al **círculo verde de "Run experiment"**.



Verás que el experimento empieza con una ventana que te pide un nombre. Si algo no funcionase bien, y el experimento no se desarrollase con normalidad, te aparecerá una ventana con el error encontrado. En el Anexo 2 te mostramos un ejemplo.

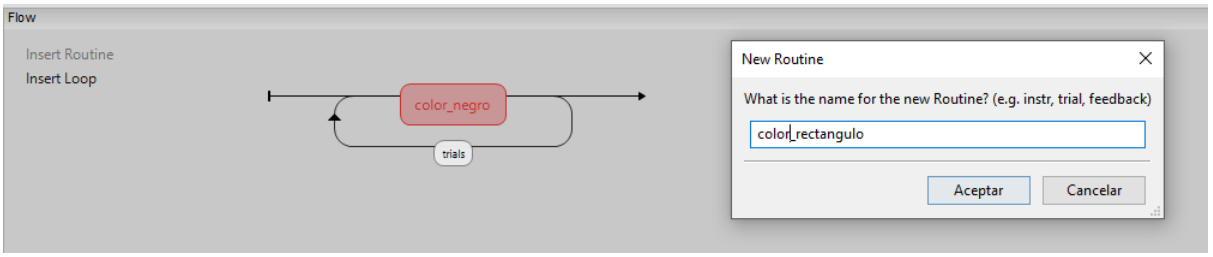
Ahora, crearemos las rutinas que nos faltan: "Rectángulos" y "Palabras de colores".

CONSEJO DEL PROFESIONAL
A medida que vayas creando y añadiendo cosas, **procura guardar y comprobar que el experimento funciona**.

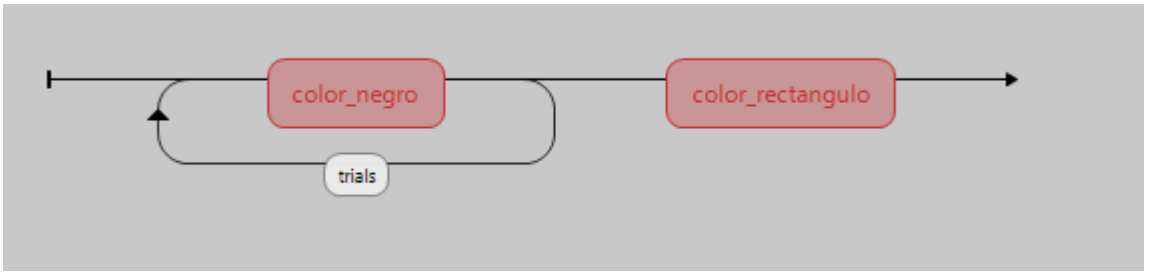
[4] En la imagen anterior, le hemos cambiado el nombre a la rutina a "color negro" (el nombre por defecto es trial). Para cambiar el nombre, simplemente hay que presionar encima del rectángulo rojo y darle a "rename".

b) “Rectángulos”

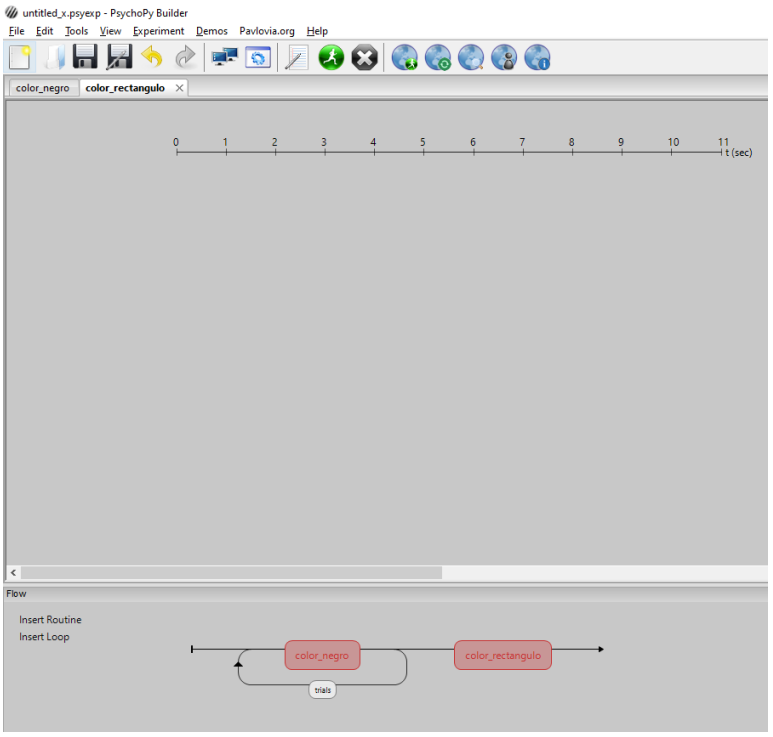
Para crear una nueva rutina, vamos al apartado de flow y le damos a “*Insert Routine*”, y le damos a (new) - ya que es una rutina nueva-. Nos pedirá un nombre y, a continuación, que determinemos una posición. Elegiremos el nombre “color_rectangulo” y la pondremos después de la de “color_negro”



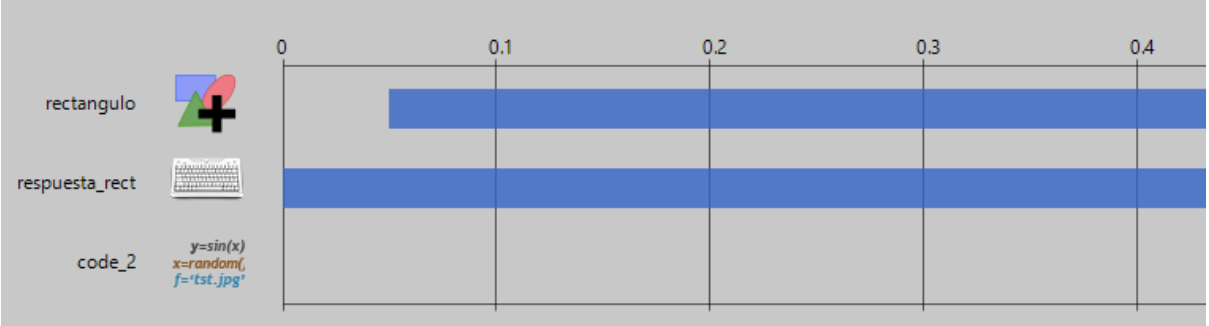
Al hacerlo, veremos que, al lado de la de color_negro, hemos creado una rutina nueva:



Al hacer esto, habremos creado una nueva estructura vacía de trail que podemos llenar con diferentes componentes:

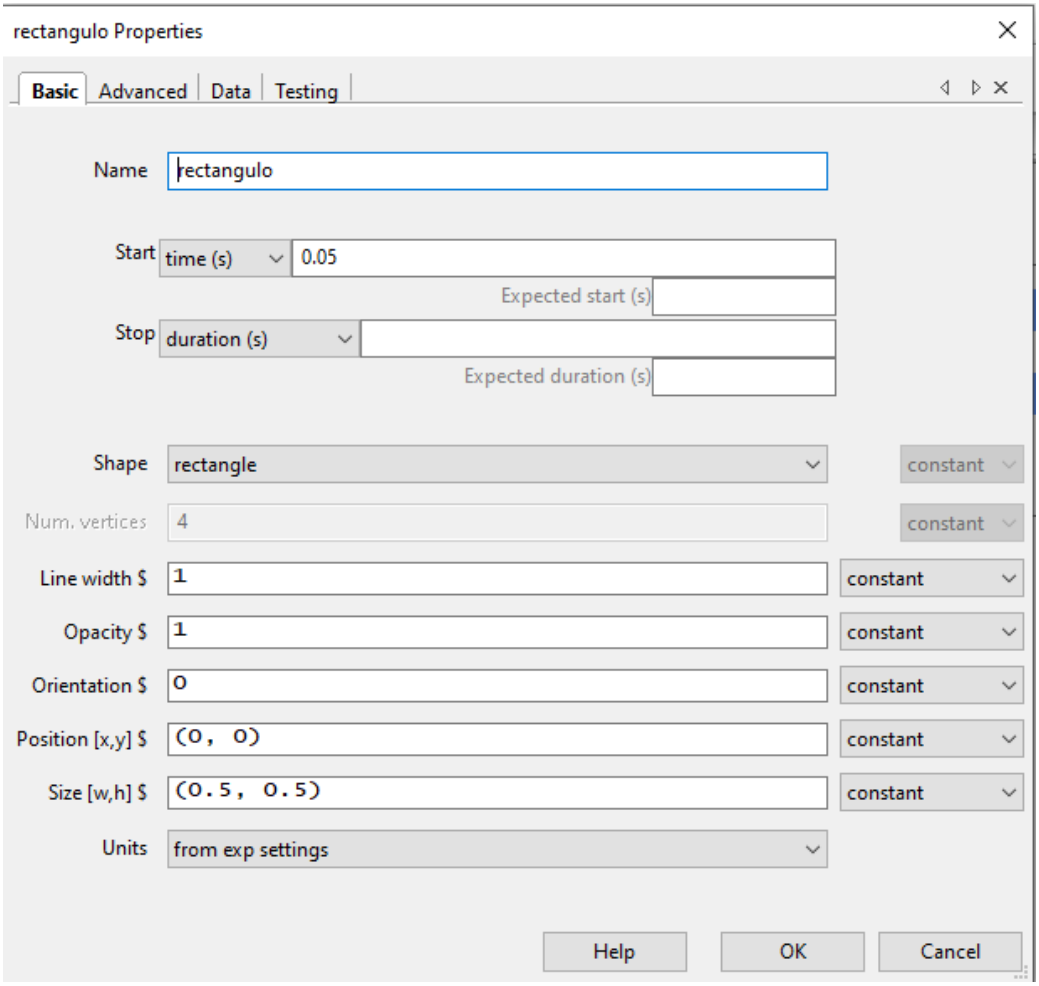


Ahora, la lógica es la misma que en el anterior ejemplo. En este caso, los componentes del trial son los siguientes:

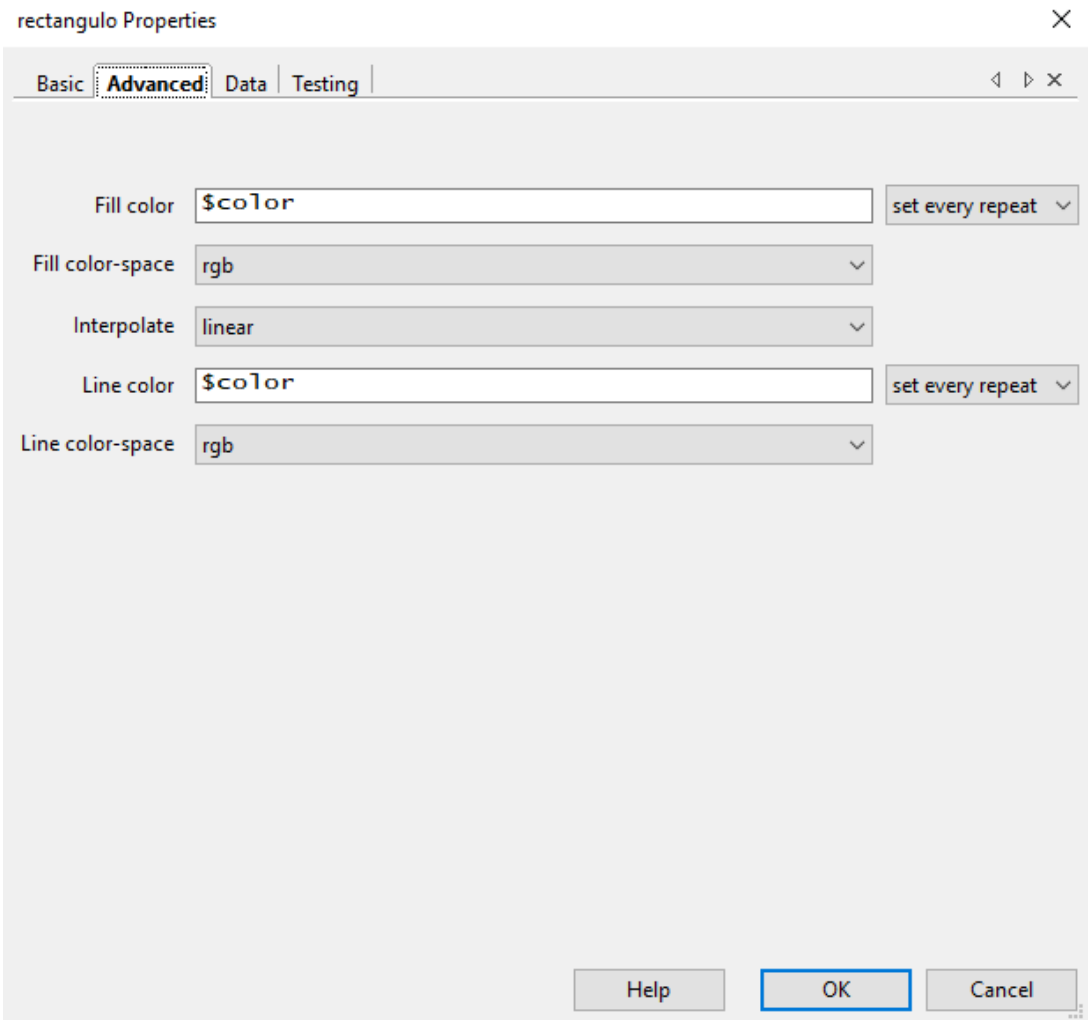


En lugar de tener un texto, el estímulo esta vez será un **polígono** (en componentes, seleccionar "*Polygon*". Después tenemos un componente de respuesta y un código suplementario, que serán similares a los de la rutina anterior.

El polígono deberá completarse de la siguiente manera:

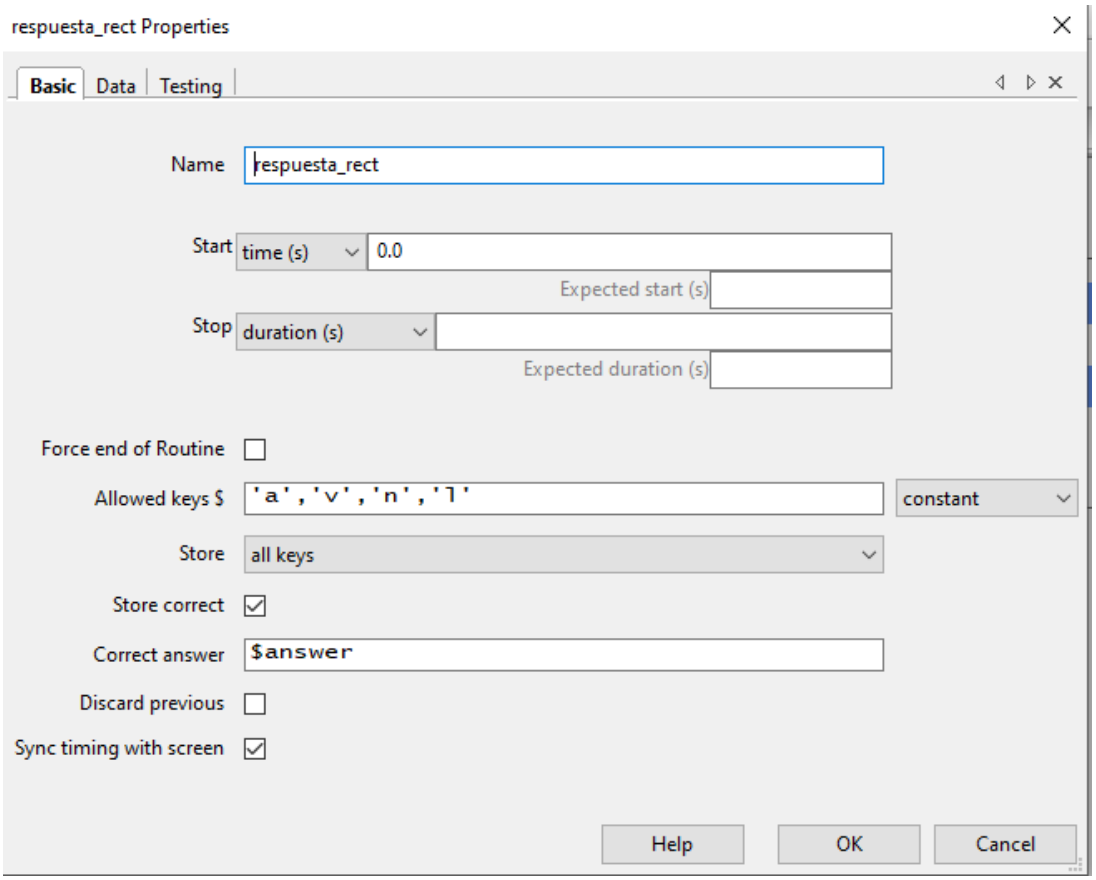


De esta manera, se especifican la forma del polígono, la opacidad, posición, dimensiones… Para especificar que el color del rectángulo debe cambiar en cada trial, nos moveremos de la pestaña de "*Basic*" (imagen anterior) a la pestaña de "*Advanced*", y la completamos de la siguiente manera:



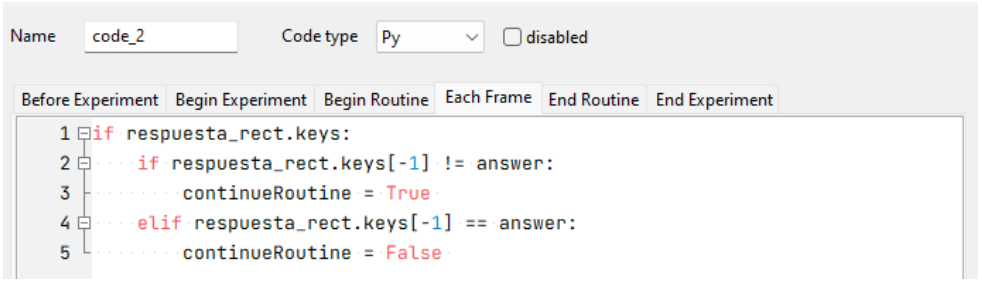
- Siguiendo la misma lógica que en la rutina anterior, en "*Fill color*" (determinamos de qué color será el interior del polígono) pondremos $color
- En *"Line Color"* (determinamos de qué color será el contorno del polígono) pondremos lo mismo ($color). En este caso, ya podemos intuir que al crear el excel que especifique cómo serán los diferentes trials, una de las columnas deberá llamarse "color".

El componente de la respuesta lo crearemos de la misma manera que en la rutina de “Palabras negras”, simplemente le cambiaremos el nombre para evitar problemas. En este caso, se rellenará así:



Finalmente, el componente de código extra para que solo avance al siguiente trial cuando la respuesta sea correcta, tendrá un aspecto parecido al que usamos en la rutina de palabras negras. Lo único distinto es que en alude al componente de respuesta del rectángulo, al que hemos llamado respuesta_rect:

```
if respuesta_rect.keys:
    if respuesta_rect.keys[-1] != answer:
        continueRoutine = True
    elif respuesta_rect.keys[-1] == answer:
        continueRoutine = False
```



Una vez creados los distintos componentes del trial de “Rectángulos”, crearemos el Excel. En este caso, en vez de tener una columna de texto, tendremos una columna de color, como se muestra en la imagen de la derecha.

Los colores están en la nomenclatura rgb dónde:

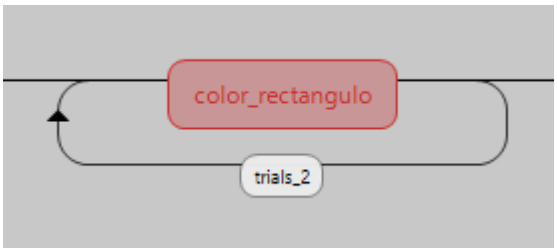
- azul = [-1, -1,1]
- naranja = [1, 0.1, -1]
- verde = [-1, 0.04, -1]
- lila = [0.004, -1, 0.004]

Al igual que antes, tenemos una columna con las respuestas correctas, pues el componente de respuesta era prácticamente el mismo que el anterior.

Este .xlsx también lo guardaremos en la misma ubicación donde se encuentran la tarea que estamos creando y el colores1_negro.xlsx. Este documento excel, tendrá el nombre “colores2_rectangulos.xlsx”.

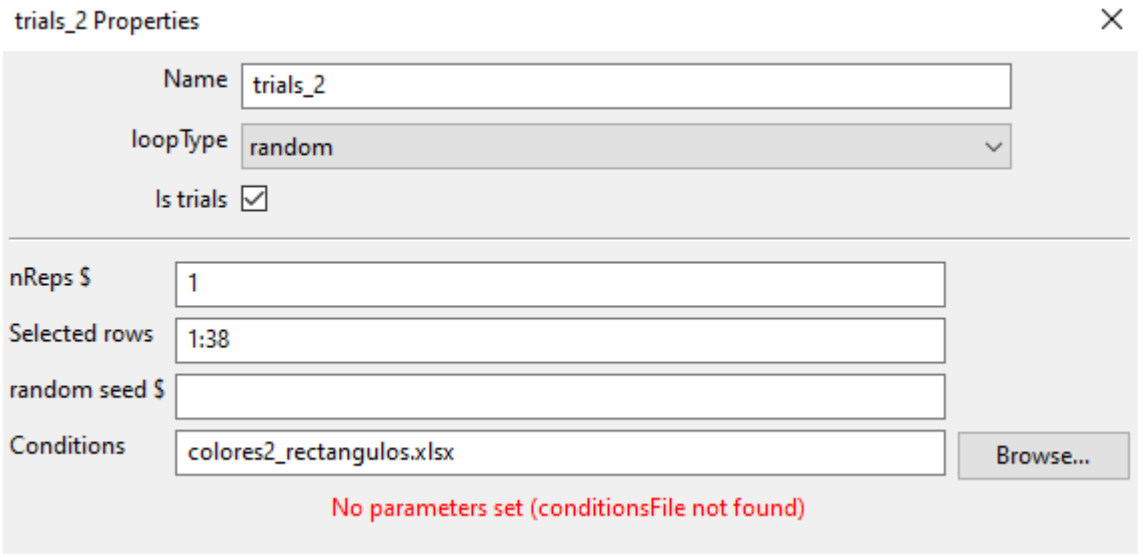
color	answer
[-1,1,1]	a
[-1,1,1]	a
[-1,1,1]	a
[-1,1,1]	a
[-1,1,1]	a
[-1,1,1]	a
[-1,1,1]	a
[-1,1,1]	a
[-1,1,1]	a
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.1,-1]	n
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[-1,0.04,-1]	v
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l
[0.004,-1,0.004]	l

Para terminar con esta rutina, hemos de crear un Loop que empiece antes de la rutina de “Rectángulos” y termine después:



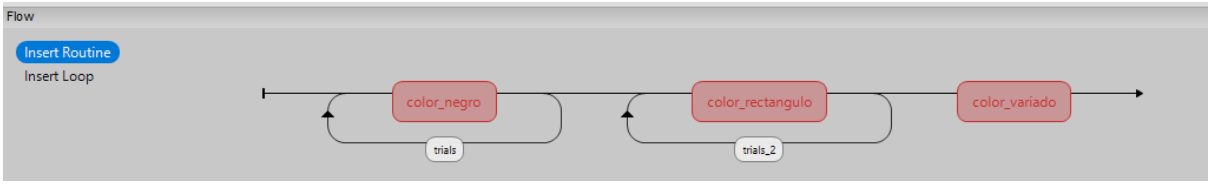
Fijémonos ahora en cómo rellenar las propiedades de este Loop.

- En este caso, el nombre será trials_2. Seguirá siendo random y comprenderá, de nuevo, de las filas 1 a la 38 (igual que antes). En este caso, el archivo de Excel que leerá para determinar cómo cambian los trials es el que acabamos de crear con el nombre “colores2_rectangulos.xlsx”.



c) “Palabras de colores”

Crearemos ahora la última rutina que nos falta. Para ello, seguiremos los pasos descritos al inicio del punto b) “Rectángulos”, pero utilizando el nombre de “color variado”. Situaremos esta rutina al final.

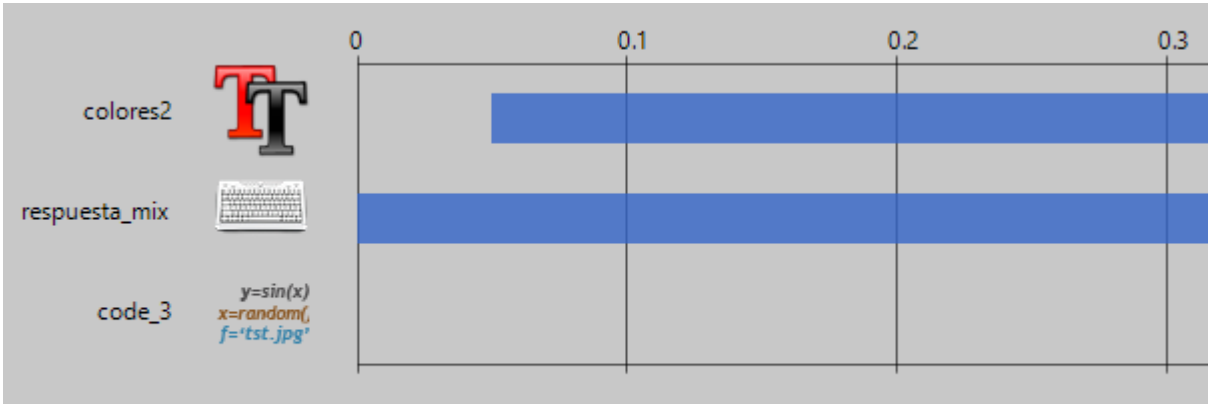


Igual que antes, se nos creará una pestaña de trial vacía en la que se pueden añadir componentes.

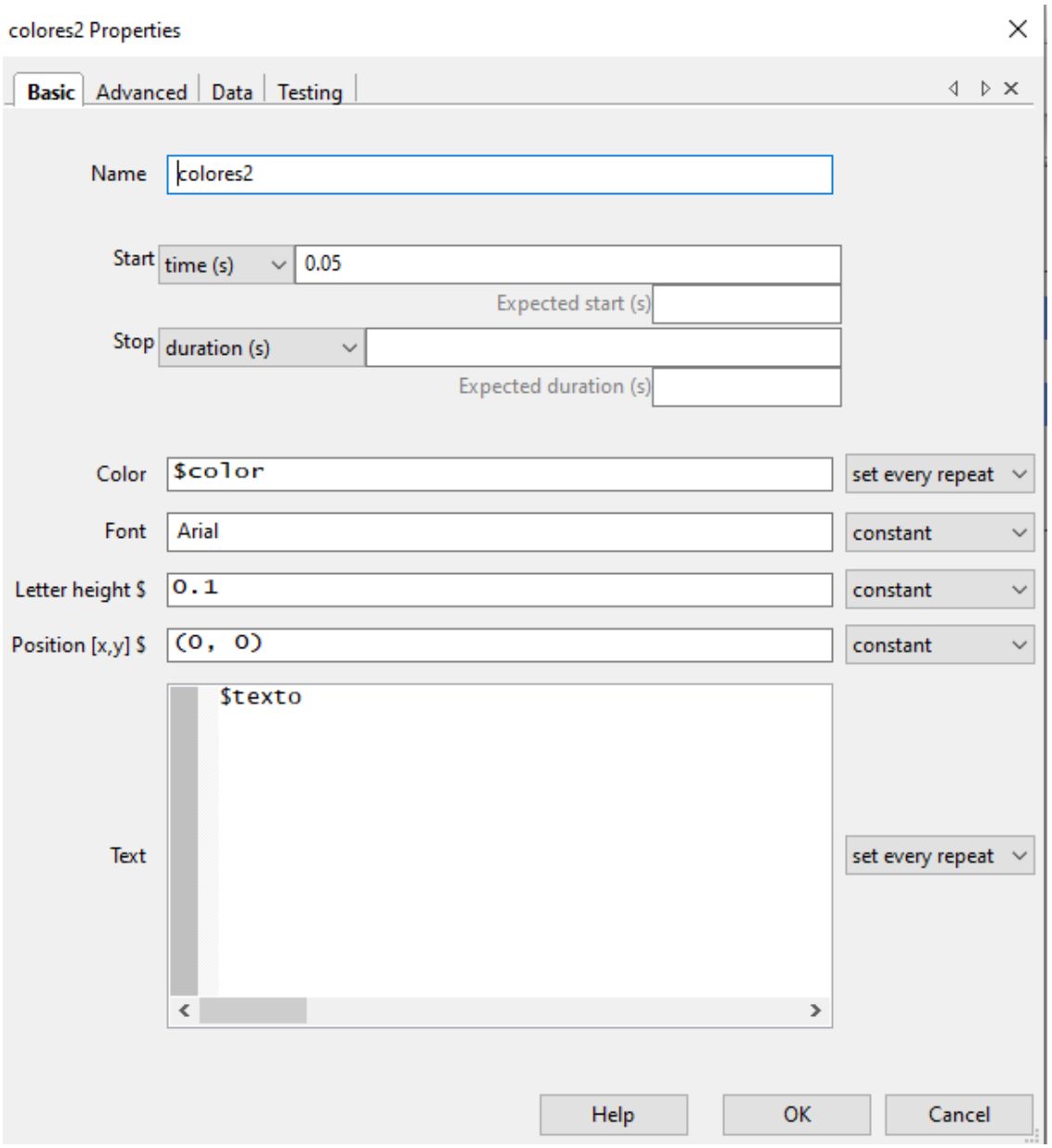
Esta rutina es una especie de combinación de las anteriores, pues tendremos un texto en el que va cambiando tanto la palabra que se presenta como el color del texto. Osea, es como en la rutina de “Palabras negras” pero ahora las palabras están escritas en diferentes colores.

ROJO - **AZUL** - **VERDE** - **NARANJA** - **ROSA**

Los componentes del trial de esta rutina vuelven a ser los que nos encontramos en “Palabras negras”:

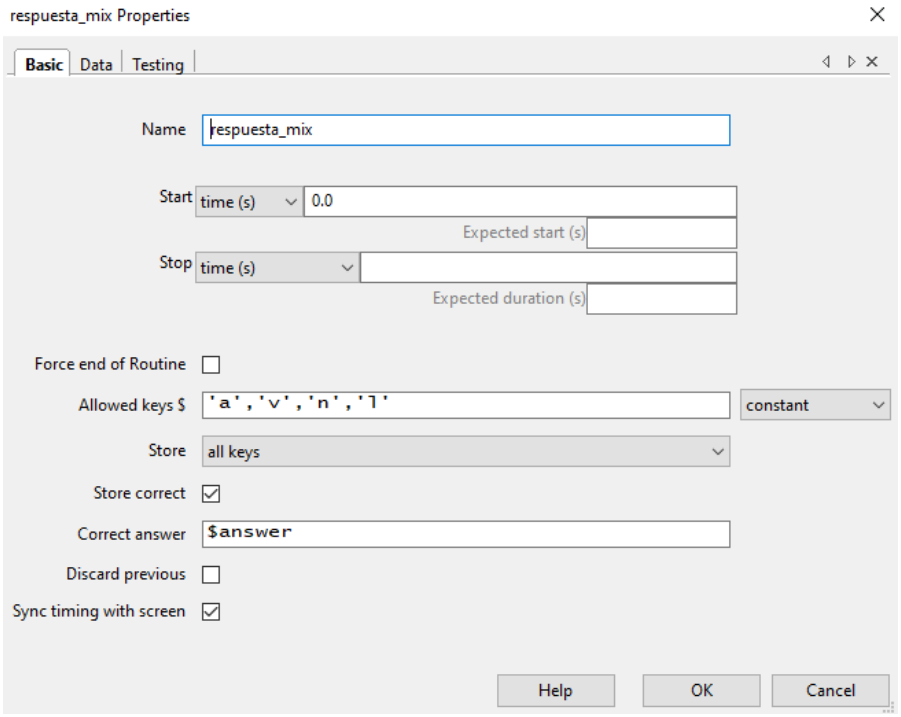


Fijémonos ahora en el componente de texto:



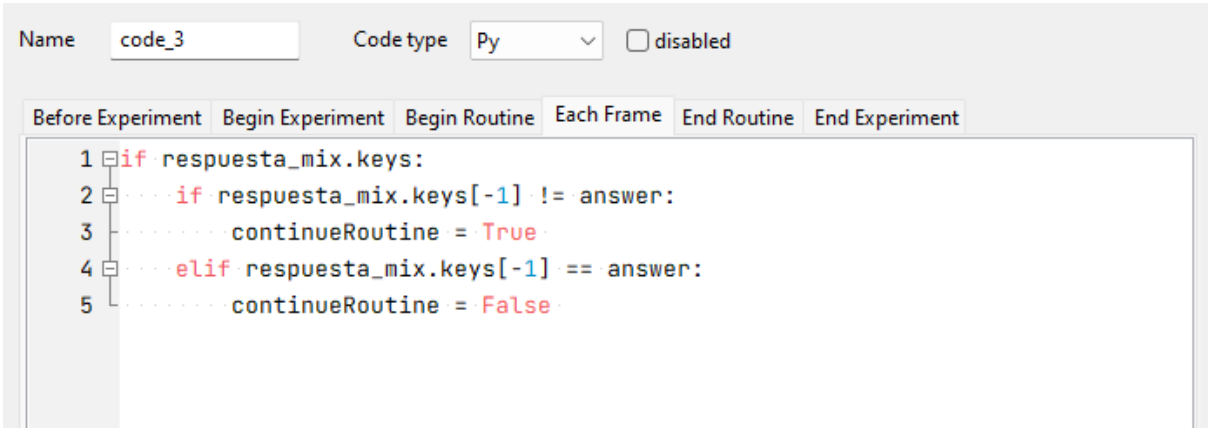
- Denominaremos a este componente “colores2”.
- A diferencia del componente de “palabras negras”, donde en color pusimos *black*, en este escribiremos $color y pondremos “*set every repeat*” en las opciones de la derecha (para que cambie de color en cada trial).
- Asimismo, en *Text* escribiremos **$texto** y también elegiremos la opción de “*set every repeat*” (para que cambie la palabra en cada trial).
    - Los cambios que hemos realizado en este componente de texto hacen que el archivo Excel que debemos crear tenga tres columnas en lugar de dos.

El componente de respuesta y el de código deben completarse de manera muy similar a los anteriores. El de respuesta (respuesta_mix):



y el código supletorio (en vez de respuesta o respuesta_rect, ahora será respuesta_mix):

```
if respuesta_mix.keys:
    if respuesta_mix.keys[-1] != answer:
        continueRoutine = True
    elif respuesta_mix.keys[-1] == answer:
        continueRoutine = False
```



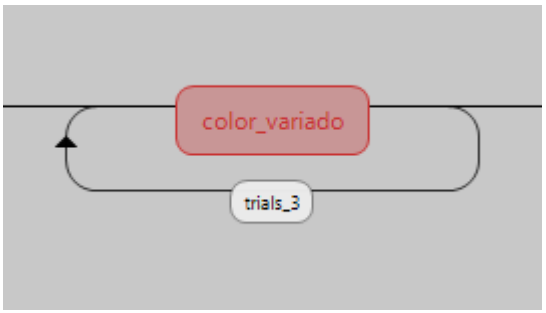
Veamos ahora cómo debe ser el archivo de excel que hemos de crear para la estructura de trials. En este caso debemos tener tres columnas, una para el texto, una para el color y otra para la respuesta correcta (que hace referencia al color en el que está escrito el texto).

Este excel está creado de manera que cada combinación posible se pregunta 3 veces (4 colores elegidos de dos en dos dan lugar a 12 combinaciones y 12x3=36. Por eso siempre hacemos 36 trials en cada rutina).

Guardaremos este excel en la misma carpeta en la que se encuentra la tarea de psychopy con el nombre “colores3_mix.xlsx”

color	texto	answer
[-1,1,1]	verde	a
[-1,1,1]	verde	a
[-1,1,1]	verde	a
[-1,1,1]	naranja	a
[-1,1,1]	naranja	a
[-1,1,1]	naranja	a
[-1,1,1]	lila	a
[-1,1,1]	lila	a
[-1,1,1]	lila	a
[-1,0.1,-1]	azul	n
[-1,0.1,-1]	azul	n
[-1,0.1,-1]	azul	n
[-1,0.1,-1]	verde	n
[-1,0.1,-1]	verde	n
[-1,0.1,-1]	verde	n
[-1,0.1,-1]	lila	n
[-1,0.1,-1]	lila	n
[-1,0.1,-1]	lila	n
[-1,0.04,-1]	azul	v
[-1,0.04,-1]	azul	v
[-1,0.04,-1]	azul	v
[-1,0.04,-1]	naranja	v
[-1,0.04,-1]	naranja	v
[-1,0.04,-1]	naranja	v
[-1,0.04,-1]	lila	v
[-1,0.04,-1]	lila	v
[-1,0.04,-1]	lila	v
[0.004,-1,0.004]	azul	l
[0.004,-1,0.004]	azul	l
[0.004,-1,0.004]	azul	l
[0.004,-1,0.004]	verde	l
[0.004,-1,0.004]	verde	l
[0.004,-1,0.004]	verde	l
[0.004,-1,0.004]	naranja	l
[0.004,-1,0.004]	naranja	l
[0.004,-1,0.004]	naranja	l

Finalmente, crearemos el Loop que debe empezar antes y terminar después de la rutina:



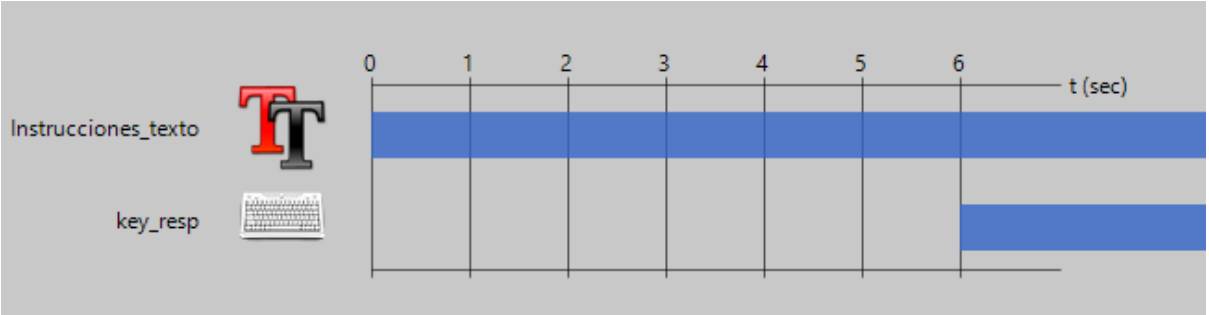
Fijémonos ahora en las propiedades de este último loop:

Tendrá un nombre distinto y mantendrá las mismas propiedades de random, repeticiones y filas seleccionadas que antes. El archivo excel que leerá, en este caso, será el “colores3_mix.xlsx”.

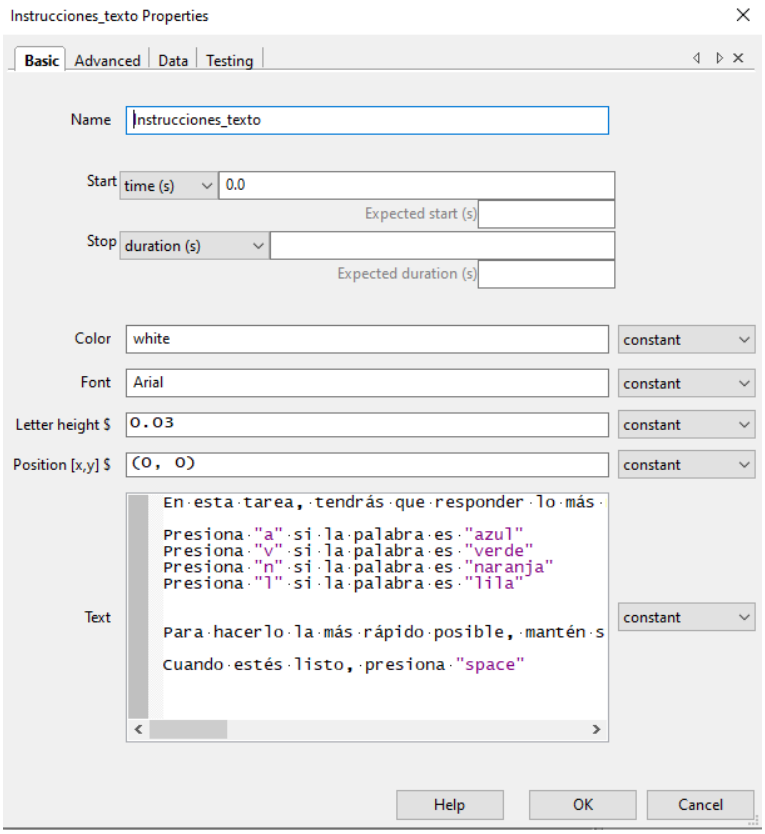
d) Instrucciones

Ya tenemos nuestro experimento diseñado. Para terminarlo, simplemente crearemos tres rutinas de instrucciones (llamadas: **instrucciones**, **instrucciones_2** e **instrucciones_3**) en medio de las rutinas con los loops.

Las rutinas de instrucciones tendrán estos componentes:

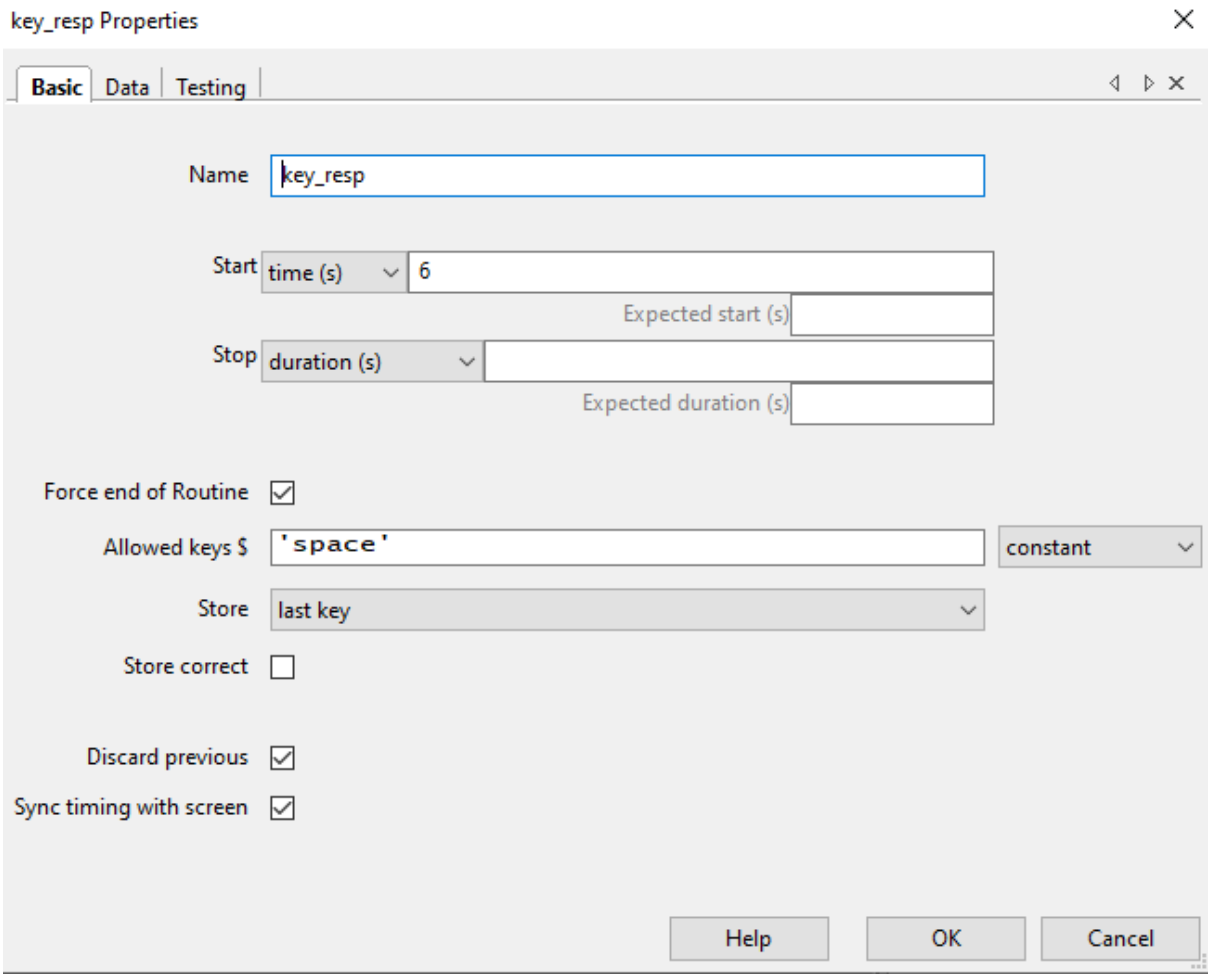


El componente de texto será así:



Las instrucciones explican al participante que ha de hacer en cada tarea. Al final de este apartado, compartimos los textos exactos a incorporar en el apartado Text del componente de texto en cada una de las rutinas de instrucciones.

El componente de respuesta será así:



En este caso, lo que se pretende es que el participante lea con atención las instrucciones, por lo que la tecla “space” no funciona hasta pasados 6 segundos y, cuando se pulsa, se termina la rutina (la función “Force end of Routine” está habilitada). En esta rutina no hemos de añadir ningún loop, ya que se trata de instrucciones sueltas.

A continuación, te presentamos los textos a incorporar en las diferentes instrucciones:

Instrucciones:

> En esta tarea, tendrás que responder lo más rápido que puedas.
>
> Presiona "a" si la palabra es "azul"
> Presiona "v" si la palabra es "verde"
> Presiona "n" si la palabra es "naranja"
> Presiona "l" si la palabra es "lila"
>
> Para hacerlo lo más rápido posible, mantén siempre un dedo encima de cada letra.
>
> Cuando estés listo, presiona "space"

Instrucciones_2:

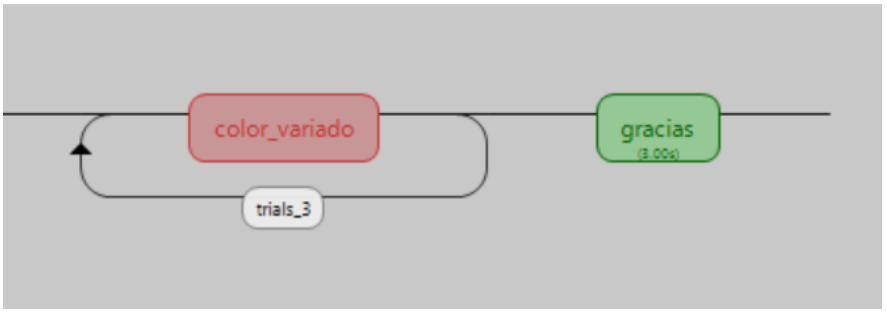
> Ahora tendrás que indicar el color del cuadrado.
>
> Presiona "a" si es de color "azul"
> Presiona "v" si es de color "verde"
> Presiona "n" si es de color "naranja"
> Presiona "l" si es de color "lila"
>
> Para hacerlo lo más rápido posible, mantén siempre un dedo encima de cada letra.
>
> Cuando estés listo, presiona "space"

Instrucciones_3:

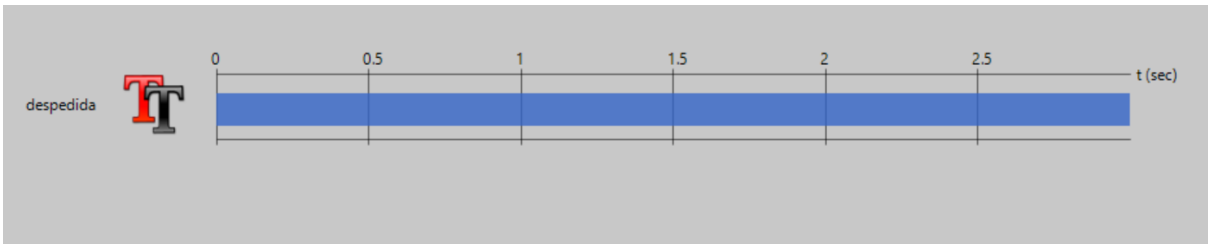
> Finalmente, tendrás que indicar de qué color está escrita la palabra.
>
> Presiona "a" si la palabra está escrita en color "azul"
> Presiona "v" si la palabra está escrita en color "verde"
> Presiona "n" si la palabra está escrita en color "naranja"
> Presiona "l" si la palabra está escrita en color "lila"
>
> Para hacerlo lo más rápido posible, mantén siempre un dedo encima de cada letra.
>
> Cuando estés listo, presiona "space"

e) Mensaje de agradecimiento

¡Ya casi lo tienes! Siempre es de buena educación despedirse dando las gracias por participar en el experimento. Para ellos, insertamos una rutina al final de todo:



Esta rutina consta de un único componente de texto al que llamaremos “despedida”

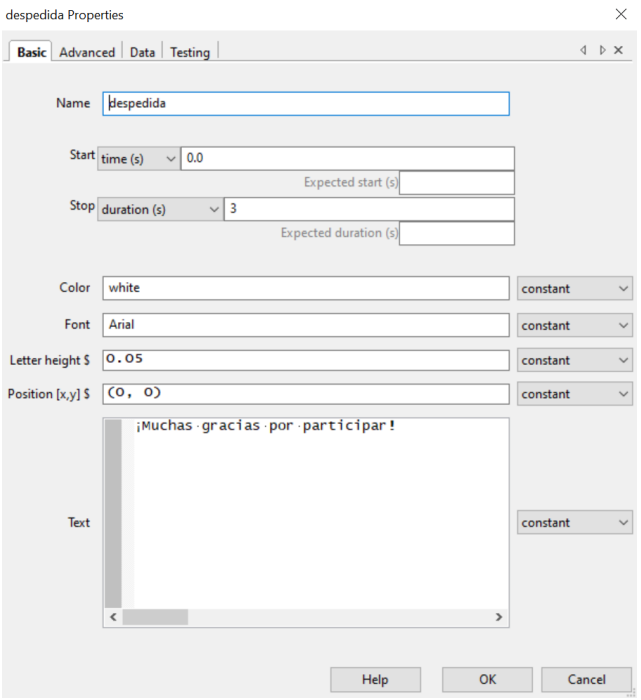


Crearemos este componente de texto como se muestra en la imagen siguiente.

Simplemente les daremos las gracias por participar y fijamos una duración de 3 segundos.

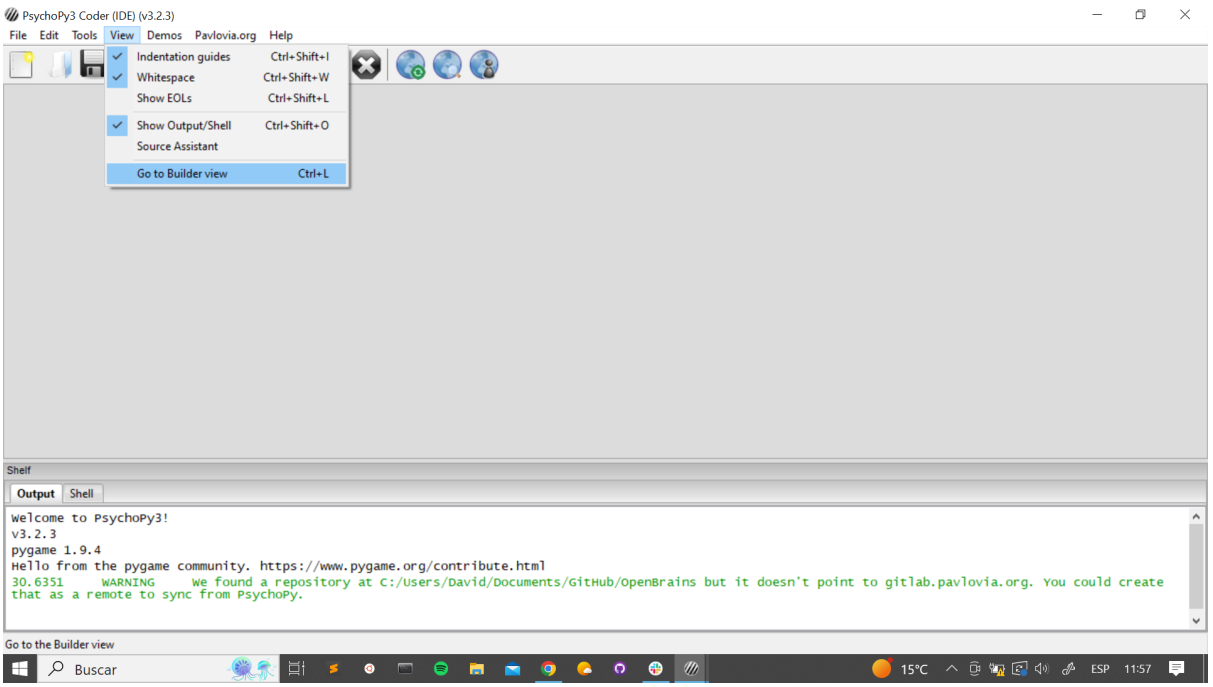
Pasados estos segundos, el experimento terminará.

¡Ya tienes tu experimento de Stroop creado y listo para funcionar! ¡Enhorabuena!



Anexo 1

Si se os abre algo parecido a esto (probable si abrís Psychopy desde *Anaconda*):



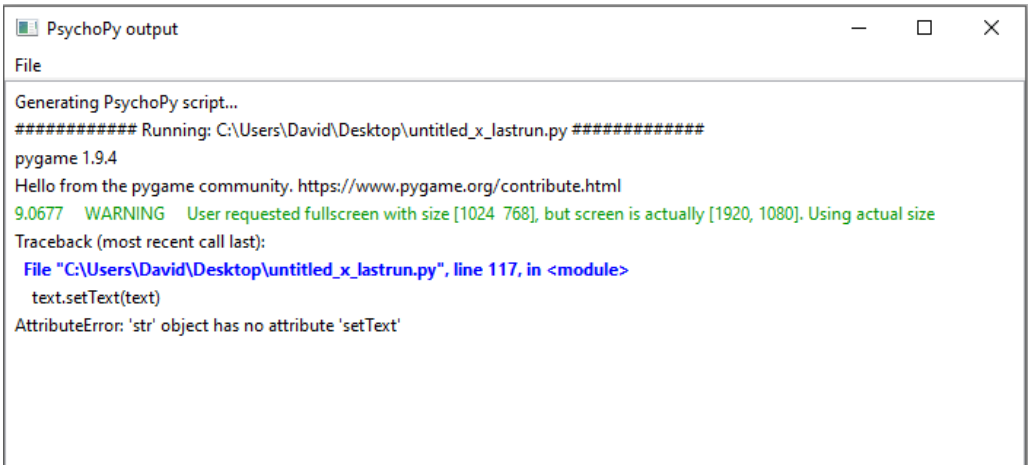
No os tenéis que preocupar.

Es simplemente que Psychopy tiene dos opciones “coder” y “builder”. Se os ha abierto la versión “coder”, que se basa en utilizar el lenguaje de programación de Python para diseñar toda la tarea. La que nos interesa es la segunda (“builder”), que es la una interfaz visual (de “drag and drop”) mucho más intuitiva que utilizaremos. Para abrirla, ir a “View” → “Go to Builder view” o pulsar Ctrl+L

Una vez hecho esto, deberíamos estar en la interfaz de la imagen anterior.

Vuelve a Abrir Psychopy.

Anexo 2



Este error me sugiere que pasa algo en el componente de especificar el texto (text.setText). En este ejemplo de error, es que he utilizado el nombre “text” para dos cosas distintas (nombre del componente y texto a introducir), por lo que cambiando eso ya funciona con normalidad. En caso de errores desconocidos, recomendamos copiar el error y realizar una búsqueda en Google, pues Psychopy cuenta con muy buena documentación para solventar errores (¡es parte de la aventura! Nunca se programa todo a la perfección la primera vez :))

Vuelve a Insertar loop.