



Table of Contents

1. [Prefacio \[Page 1\]](#)
 1. [Cómo este libro está organizado \[Page 2\]](#)
 2. [Para quién es este libro \[Page 2\]](#)
 3. [Convenciones usadas en este libro \[Page 2\]](#)
 4. [Usando los ejemplos de código \[Page 3\]](#)
 5. [Agradecimientos \[Page 3\]](#)
2. [Capítulo 1. Hola \[Page 4\]](#)
 1. [Bosquejo y prototipado \[Page 4\]](#)
 2. [Flexibilidad \[Page 4\]](#)
 3. [Gigantes \[Page 5\]](#)
 4. [Árbol familiar \[Page 5\]](#)
 5. [Únete \[Page 5\]](#)
3. [Capítulo 2. Empezando a programar \[Page 6\]](#)
 1. [Ambiente \[Page 6\]](#)
 2. [Descarga y configuración de archivos \[Page 6\]](#)
 3. [Tu primer programa \[Page 6\]](#)
 4. [Ejemplo 2-1: dibuja una elipse \[Page 7\]](#)
 5. [Ejemplo 2-2: hacer círculos \[Page 8\]](#)
 6. [La consola \[Page 8\]](#)
 7. [Creando un nuevo proyecto \[Page 8\]](#)
 8. [Ejemplos y referencia \[Page 9\]](#)

Prefacio

p5.js está inspirado y guiado por otro proyecto, que empezó hace 15 años. En el año 2001, Casey Reas y Ben Fry empezaron a trabajar en una nueva plataforma para hacer más fácil la programación de gráficos interactiva, la nombraron Processing. Ellos estaban frustrados por lo difícil que era escribir este tipo de software con los lenguajes que normalmente usaban (C++ y Java), y fueron inspirados por lo simple que era escribir programas interesantes con los lenguajes que usaban cuando niños (Logo y BASIC). Su mayor influencia fue Design by Numbers (DBN), un lenguaje que ellos estaban trabajando en mantenimiento y enseñando en ese tiempo (y que fue creado por su tutor de investigación, John Maeda). Con Processing, Ben y Casey estaban buscando una mejor manera de probar sus ideas en código, en vez de solo conversarlas o pasar demasiado tiempo programándolas en C++. Su otro objetivo era construir un lenguaje para enseñar cómo programar a estudiantes de diseño y de arte y también brindarles una manera más fácil de trabajar con gráficos a estudiantes más avanzados. Esta combinación es una desviación positiva de la manera en que comúnmente se enseña programación. Los nuevos usuarios empiezan concentrándose en gráficos e interacción en vez de estructuras de datos y resultados en forma de texto en la consola. A través de los años, Processing se ha transformado en una gran comunidad. Es usado en salas de clases en todo el mundo, en planes de estudios de artes, humanidades y ciencias de la computación, además de profesionales. Hace dos años, Ben y Casey se me acercaron con una pregunta: ¿cómo se vería Processing si funcionara en la web? p5.js empieza con el

objetivo original de Processing, hacer que programar sea accesible para artistas, diseñadores, educadores y principiantes, y luego lo reinterpreta para la web actual usando Javascript y HTML. El desarrollo de p5.js ha sido como acercar mundos distintos. Para facilitar la transición a la Web de los usuarios de la existente comunidad de Processing, nos adherimos a la sintaxis y a las convenciones de Processing tanto como fuera posible. Sin embargo, p5.js está construido con Javascript, mientras que Processing está construido con un lenguaje llamado Java. Estos dos lenguajes tienen distintos patrones y funciones, así que a en ocasiones nos tuvimos que desviar de la sintaxis de Processing. También fue importante que p5.js fuera integrado sin problemas a las existentes características, herramientas y marcos de la web, para atraer usuarios familiarizados con la web pero novatos en programación creativa. Sintetizar todos estos factores fue un gran desafío, pero el objetivo de unir estos marcos proporcionó un camino claro a seguir en el desarrollo de p5.js. Una primera versión beta fue lanzada en agosto del 2014. Desde ese entonces, ha sido usado e integrado a programas de estudios en todo el mundo. Existe un editor oficial de p5.js que está actualmente en desarrollo, y ya se ha avanzado en muchas nuevas características y librerías.

p5.js es un esfuerzo comunitario - cientos de personas han contribuido funciones esenciales, soluciones a errores, ejemplos, diseño, reflexiones y discusión. Pretendemos continuar la visión y el espíritu de la comunidad de Processing mientras la abrimos aún más en la Web.

Cómo este libro está organizado

Los capítulos de este libro están organizados de la siguiente manera:

Para quién es este libro

Este libro fue escrito para personas que quieren crear imágenes y programas interactivos simples a través de una casual y concisa introducción a la programación de computadores. Es para personas que quieren una ayuda para entender los miles de ejemplos de código en p5.js y los manuales de referencia disponibles en la web de manera gratuita. Introducción a p5.js no es un libro de referencia sobre programación. Como el título sugiere, te hará una introducción. Es para adolescentes, entusiastas, abuelos, y cualquier persona entremedio. Este libro es apropiado también para personas con experiencia en programación que quieren aprender los conceptos básicos sobre gráficas de computador interactivas. Introducción a p5.js contiene técnicas que pueden ser aplicadas a crear juegos, animaciones e interfaces.

Convenciones usadas en este libro

Las siguientes convenciones tipográficas son usadas en este libro:

Usando los ejemplos de código

El material complementario (ejemplos de código, ejercicios, etc.) está disponible para descarga.

Agradecimientos

Le agradecemos a Brian Jepson y Anna Kaziunas France por su gran energía, apoyo y visión.

No nos imaginamos este libro sin el ejemplo de Introducción a Arduino de Massimo Banzi. Este excelente libro de Massimo es el prototipo.

Capítulo 1. Hola

p5.js sirve para escribir software que produce imágenes, animaciones e interacciones. La intención es escribir una línea de código y que un círculo aparezca en la pantalla. Añade unas pocas líneas de código, y ahora el círculo sigue al ratón. Otra línea de código, y el círculo cambia de color cuando presionas el ratón. Le llamamos a esto bosquejar con código. Tú escribes una línea, luego añades otra, luego otra, y así. El resultado es el programa creado una pieza a la vez. Los cursos de programación típicamente se enfocan primero en estructura y teoría. Cualquier aspecto visual - una interfaz, una animación - es considerado un postre que solo puede ser disfrutado después de que terminas de comer tus vegetales, equivalente a varias semanas de estudiar algoritmos y métodos. A través de los años, hemos visto a muchos amigos tratar de tomar estos cursos, para luego abandonarlos después de la primera sesión o después de una muy larga y frustrante noche previa a la entrega de la primera tarea. Toda curiosidad inicial que tenían sobre hacer que el computador trabaje para ellos es perdida porque no pueden ver un camino claro entre lo que tienen que aprender al principio y lo que quieren crear.

p5.js ofrece una manera de programar a través de la creación de gráficas interactivas. Hay muchas maneras posibles de enseñar código, pero los estudiantes usualmente encuentran apoyo y motivación en retroalimentación visual inmediata. p5.js provee esta retroalimentación, y su énfasis en imágenes, prototipado y comunidad es discutido en las siguientes páginas.

Bosquejo y prototipado

Bosquejar es una manera de pensar, es jugueteo y rápido. El objetivo básico es explorar muchas ideas en un corto periodo de tiempo. En nuestro propio trabajo, usualmente empezamos bosquejando en papel y luego trasladando nuestros resultados a código. Las ideas para animación e interacción son usualmente bosquejadas como un guión gráfico con anotaciones. Después de hacer algunos bosquejos en software, las mejores ideas son seleccionadas y combinadas en prototipos. Es un proceso cíclico de hacer, probar y mejorar que va y viene entre papel y pantalla.

Flexibilidad

Tal como un cinturón de herramientas para software, p5.js consiste de muchas herramientas que funcionan juntas en diversas combinaciones. Como resultado, puede ser usado para exploraciones rápidas o para investigación en profundidad. Porque un programa hecho con p5.js puede ser tan corto como unas pocas líneas de código o tan largo como miles, existe espacio para crecimiento y variación. Las librerías de p5.js lo extienden a otros dominios incluyendo trabajar con sonido y la adición de botones, barras deslizadoras, cajas de entrada y captura de cámara con HTML.

Gigantes

Las personas han estado haciendo imágenes con computadores desde los años 1960s, y hay mucho que podemos aprender de esta historia. Por ejemplo, antes de que los computadores pudieran proyectar a pantallas CRT o LCD, se usaban grandes máquinas trazadoras para dibujar las imágenes. En la vida, todos nos paramos sobre hombros de gigantes, y los titanes para p5.js incluyen pensadores del diseño, gráfica computacional, arte, arquitectura, estadística y disciplinas intermedias. Dale un vistazo a Sketchpad (1963) por Ivan Sutherland, Dynabook (1968) por Alan Kay y otros artistas destacados en Artist and Computer (Harmony Books, 1976) por Ruth Leavitt. El ACM SIGGRAPH y Ars Electronica brindan atisbos fascinantes en la historia de la gráfica y el software.

Árbol familiar

Como los lenguajes humanos, los lenguajes de programación pertenecen a familias de lenguajes relacionados. p5.js es un dialecto de un lenguaje de programación llamado Javascript. La sintaxis del lenguaje es casi idéntica, pero p5.js añade características personalizadas relacionadas a gráficas e interacción y provee un acceso simple a características de HTML5 nativas que ya están soportadas por los navegadores web. Por estas características compartidas, aprender p5.js es un paso útil para aprender a programar en otros lenguajes y usar otras herramientas computacionales.

Únete

Miles de personas usan p5.js cada día. Como ellos, tú puedes descargar p5.js gratuitamente. Incluso tienes la opción de modificar el código de p5.js para que se adapte a tus necesidades. p5.js es un proyecto FLOSS (esto es, free/libre/open source software) y en el espíritu de esta comunidad, te alentamos a participar y compartir tus proyectos y tu conocimiento en línea en <http://p5js.org>.

Capítulo 2. Empezando a programar

Para sacar el máximo provecho de este libro, no basta con solo leerlo. Necesitas experimentar y practica. No puedes aprender a programar solamente leyendo - debes hacerlo. Para empezar, descarga p5.js y haz tu primer bosquejo.

Ambiente

Primero, necesitarás un editor de código. Un editor de código es similar a un editor de texto (como Bloc de notas), excepto que tiene una funcionalidad especial para editar código en vez de texto plano. Puedes usar cualquier editor que quieras, te recomendamos Atom y Brackets, ambos disponibles para descarga. También existe un editor oficial de p5.js en desarrollo. Si lo quieres usar, lo puedes descargar visitando <http://p5js.org/download> y seleccionando el botón que dice "Editor". Si estás usando el editor de p5.js, puedes saltar a la sección "Tu primer programa".

Descarga y configuración de archivos

Empieza por visitar <http://p5js.org/download> y selecciona "p5.js complete". Tras la descarga, haz doble click en el archivo .zip y arrastra el directorio a alguna ubicación en tu disco duro. Puede ser Archivos de programa o Documentos o simplemente tu Escritorio, pero lo importante es que el directorio p5 sea extraído de este archivo .zip. El directorio p5 contiene un proyecto de ejemplo con el que puedes empezar a trabajar. Abre tu editor de código. Luego abre el directorio llamado "empty-example" en tu editor de código. En la mayoría de los editores de código, puedes hacer esto seleccionando en el menú Archivo la opción Abrir, y luego seleccionando "empty-example". Ahora estás listo para empezar tu primer programa!

Tu primer programa

Cuando abras el directorio "empty-example", lo más probable es que veas una barra lateral con el nombre del directorio en la parte superior y una lista con los archivos contenidos en este directorio. Si haces click en alguno de estos archivos, verás los contenidos del archivo aparecer en el área principal.

Un bosquejo en p5.js está compuesto de unos cuantos lenguajes distintos usados en conjunto. HTML (HyperText Markup language) brinda la columna vertebral, enlazando todos los otros elementos en la página. Javascript (y la librería p5.js) te permiten crear gráficas interactivas que puedes mostrar en tu página HTML. A veces CSS (Cascading Style Sheets) es usado para definir elementos de estilo en la página HTML, pero no cubriremos esta materia en este libro.

Si revisas el archivo `index.html`, te darás cuenta que contiene un poco de código HTML. Este archivo brinda la estructura a tu proyecto, uniendo la librería `p5.js` y otro archivo llamado `sketch.js`, donde tú escribiras tu propio programa. El código que crea estos enlaces tiene esta apariencia:

No necesitas hacer nada en el código HTML en este momento - ya está configurado para ti. Luego, haz click en `sketch.js` y revisa el código:

El código plantilla contiene dos bloques, o funciones, `setup()` y `draw()`. Puedes poner tu código en cualquiera de los dos lugares, y cada uno tiene un propósito específico.

Cualquier código que esté involucrado en la definición del estado inicial de tu programa corresponde al bloque `setup()`. Por ahora, lo dejaremos vacío, pero más adelante en el libro, añadirás código aquí para especificar el tamaño de tu lienzo para tus gráficas, el peso de tu trazado o la velocidad de tu programa.

Cualquier código involucrado en realmente dibujar contenido a la pantalla (definir el color de fondo, dibujar figuras, texto o imágenes) será colocado en el bloque `draw()`. Es aquí donde empezarás a escribir tus primeras líneas de código.

Ejemplo 2-1: dibuja una elipse

Entre las llaves del bloque `draw()`, borra el texto `//put drawing code here` y reemplázalo con el siguiente:

```
background(204; ellipse(50, 50, 80, 80);
```

Tu programa entero deberá verse así:

Esta nueva línea de código significa "dibuja una elipse, con su centro 50 pixeles a la derecha desde el extremo izquierdo y 50 pixeles hacia abajo desde el extremo superior, con una altura y un ancho de 80 pixeles". Graba el código presionando Command-S, o desde el menú con File-Save.

Para ver el código corriendo, puedes abrir `index.html` en cualquier navegador web (como Chrome, Firefox o Safari). Navega al directorio "empty-example" en tu explorador de archivos y haz doble click en `index.html` para abrirlo. Otra alternativa es desde el navegador web, escoger Archivo-Abrir y seleccionar el archivo `index.html`.

Si has escrito todo correctamente, deberías ver un círculo en tu navegador. Si no lo ves, asegúrate de haber copiado correctamente el código de ejemplo. Los números tienen que estar entre paréntesis y tener comas entre ellos. La línea debe terminar con un punto coma.

Una de las cosas más difíciles sobre empezar a programar es que tienes que ser muy específico con la sintaxis. El software `p5.js` no es siempre suficientemente inteligente como

para entender lo que quieres decir, y puede ser muy exigente con la puntuación. Te acostumbrarás a esto con un poco de práctica.

A continuación, avanzaremos para hacer esto un poco más emocionante.

Ejemplo 2-2: hacer círculos

Borra el texto del ejemplo anterior, y prueba este. Graba tu código, y refresca (Command-R) index.html en tu navegador para verlo actualizado.

Este programa crea un lienzo para gráficas que tiene un ancho 480 pixeles y una altura de 120 pixeles, y luego empieza a dibujar círculos blancos en la posición de tu ratón. Cuando presionas un botón del ratón, el color del círculo cambia a negro. Explicaremos después y en detalle más de los elementos de este programa. Por ahora, corre el código, mueve el ratón y haz click para experimentarlo.

La consola

El navegador web tiene incluida una consola que puede ser muy útil para depurar programas. Cada navegador tiene una manera diferente de abrir la consola. Aquí están las instrucciones sobre cómo hacerlo con los navegadores más típicos:

Para abrir la consola en Chrome, desde el menú superior escoge View-Developer-Javascript Console

Con Firefox, desde el menú superior escoge Tools-Web-Developer-Web Console.

Usando Safari, necesitarás habilitar la funcionalidad antes de que puedas usarla. Desde el menú superior, selecciona Preferencias y luego haz click en la pestaña Avanzado y activa la casilla "Show develop menu in menu bar". Tras hacer esto, serás capaz de seleccionar Develop-Show Error Console.

En Internet Explorer, abre F12 Developer Tools, luego selecciona Console Tool.

Deberías ahora ver un recuadro en la parte inferior o lateral de tu pantalla. Si hay un error de digitación, aparecerá texto rojo explicando qué error es. Este texto puede a veces ser críptico, pero si revisas al lado derecho de la línea, estará el nombre del archivo y el número de la línea de código donde fue detectado el error. Ese es un lugar adecuado donde empezar a buscar errores en tu programa.

Creando un nuevo proyecto

Haz creado un bosquejo desde un ejemplo vacío, ¿pero cómo creas un nuevo proyecto? La manera más fácil de hacerlo es ubicando el directorio "empty-example" en tu explorador de

archivos y luego copiar y pegarlo para crear un segundo "empty-example". Puedes renombrar la carpeta a lo que quieras - por ejemplo, Proyecto 2.

Ahora puedes abrir este directorio en tu editor de código y empezar a hacer un nuevo bosquejo. Cuando quieras verlo en el navegador, abre el archivo `index.html` dentro de tu nuevo directorio Proyecto 2.

Siempre es una buena idea grabar tus bosquejos frecuentemente. Mientras estás probando cosas nuevas, graba tu bosquejo con diferentes nombres (Archivo-Guardar como), para que así siempre seas capaz de volver a versiones anteriores. Esto es especialmente útil si - o cuando - algo se rompe.

Nota

Un error común es estar editando un proyecto pero estar viendo otro en el navegador web, haciendo que no puedas ver los cambios que has hecho. Si te das cuenta que tu programa se ve igual a pesar de haber hecho cambios a tu código, revisa que estás viendo el archivo `index.html` correcto.

Ejemplos y referencia

Aprender cómo programar con p5.js involucra explorar mucho código: correr, alterar, romper y mejorarlo hasta que lo hayas reformulado en algo nuevo. Con esto en mente, el sitio web de p5.js tiene docenas de ejemplos que demuestran diferentes características de la librería. Visita la página de Ejemplos para verlos. Puedes jugar con ellos editando el código en la página y luego haciendo click en "Run". Los ejemplos están agrupados en distintas categorías según su función, como Forma, color, e imagen. Encuentra un tema que te interese en la lista y prueba un ejemplo.

Si ves una parte del programa con la que no estás familiarizado o sobre la que quieres aprender su funcionalidad, visita la referencia de p5.js.

La referencia de p5.js explica cada elemento de código con una descripción y ejemplos. Los programas en la Referencia son mucho más cortos (usualmente cuatro o cinco líneas) y más fáciles de seguir que los ejemplos de la página Learn. Ten en cuenta que estos ejemplos usualmente omiten `setup()` y `draw()` por simplicidad, pero estas líneas de código que ves deberán ser puestas dentro de uno de estos bloques para poder ser ejecutadas. Recomendamos mantener la página de Referencia abierta mientras estás leyendo este libro y mientras estás programando. Puede ser navegada por tema o usando la barra de búsqueda en la parte superior de la página.

La Referencia fue escrita con el principiante en mente, esperamos que sea clara y entendible. Estamos muy agradecidos de las personas que han visto errores y los han señalado. Si crees que puedes mejorar una entrada en la referencia o que has encontrado algún error, por favor haznos saber esto haciendo click en el link en la parte inferior de la página de referencia.