Publish

Paul Carleton • Sep 3

Fork of Do it Live! with ObservableHQ

Hago en directo! con ObservableHQ

presentación por Paul Carleton de Stripe para EventLoop en la Ciudad de México, 4/9/2019

English version

Hago en directo con ObservableHQ

Paul Carleton @paulcarletonjr

Ingeniero de Software @ Stripe

Slides: pcarleton.com/eventloop

Meta

SLIDES: **PCARLETON.COM/EVENTLOOP**

Hecho en Observable!

Hay un experimento con la participación más adelante!

Feedback Loops

EDITOR & SCRIPT

1. Abro mi editor de texto
2. Escribo cualquier código (x = 5, y = 6, x + 5 =...)
3. Guardo mi script.
4. Cambio a mi terminal
5. Ejecuto mi script
6. Miro el resultado

Utilizas el REPL?

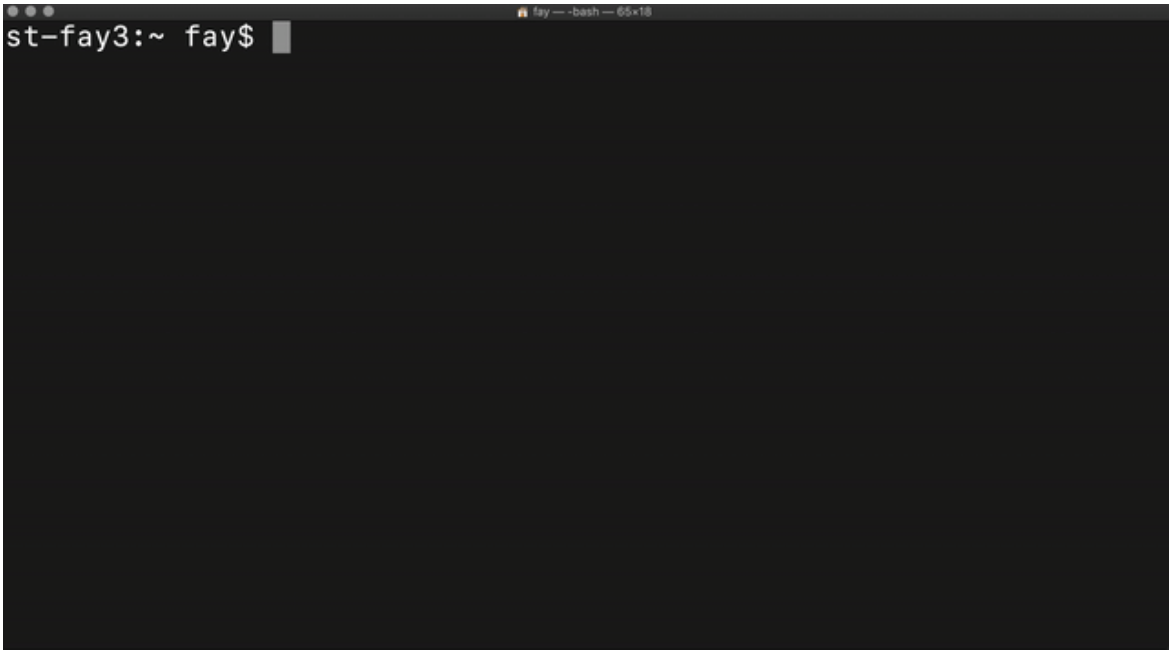
Utilizas el REPL?

◉ Nunca ○ A veces ○ Todos los días!

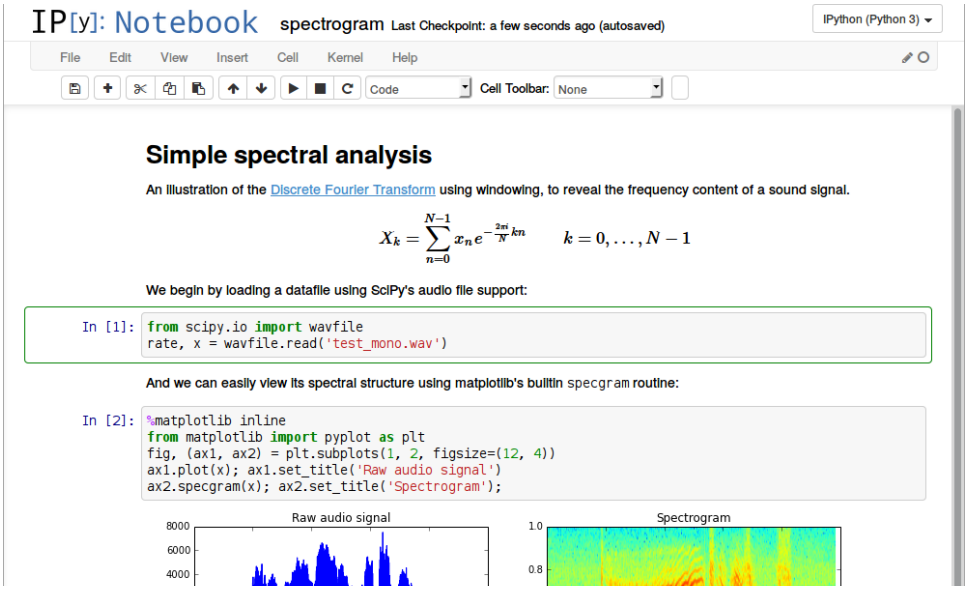
Selecciona tu respuesta

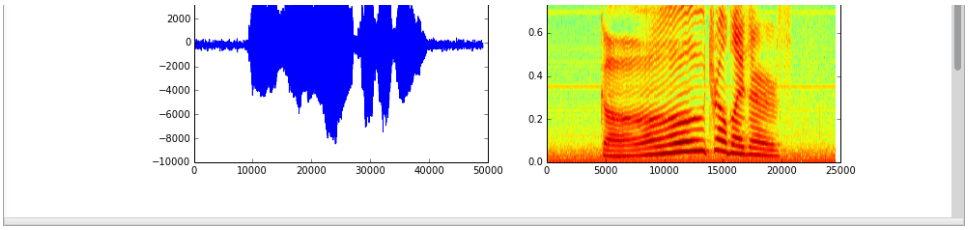
▸ `Response {}`

REPL: Read Eval Print Loop



Utilizas Jupyter?





Utilizas Jupyter?

Utilizas Jupiter?

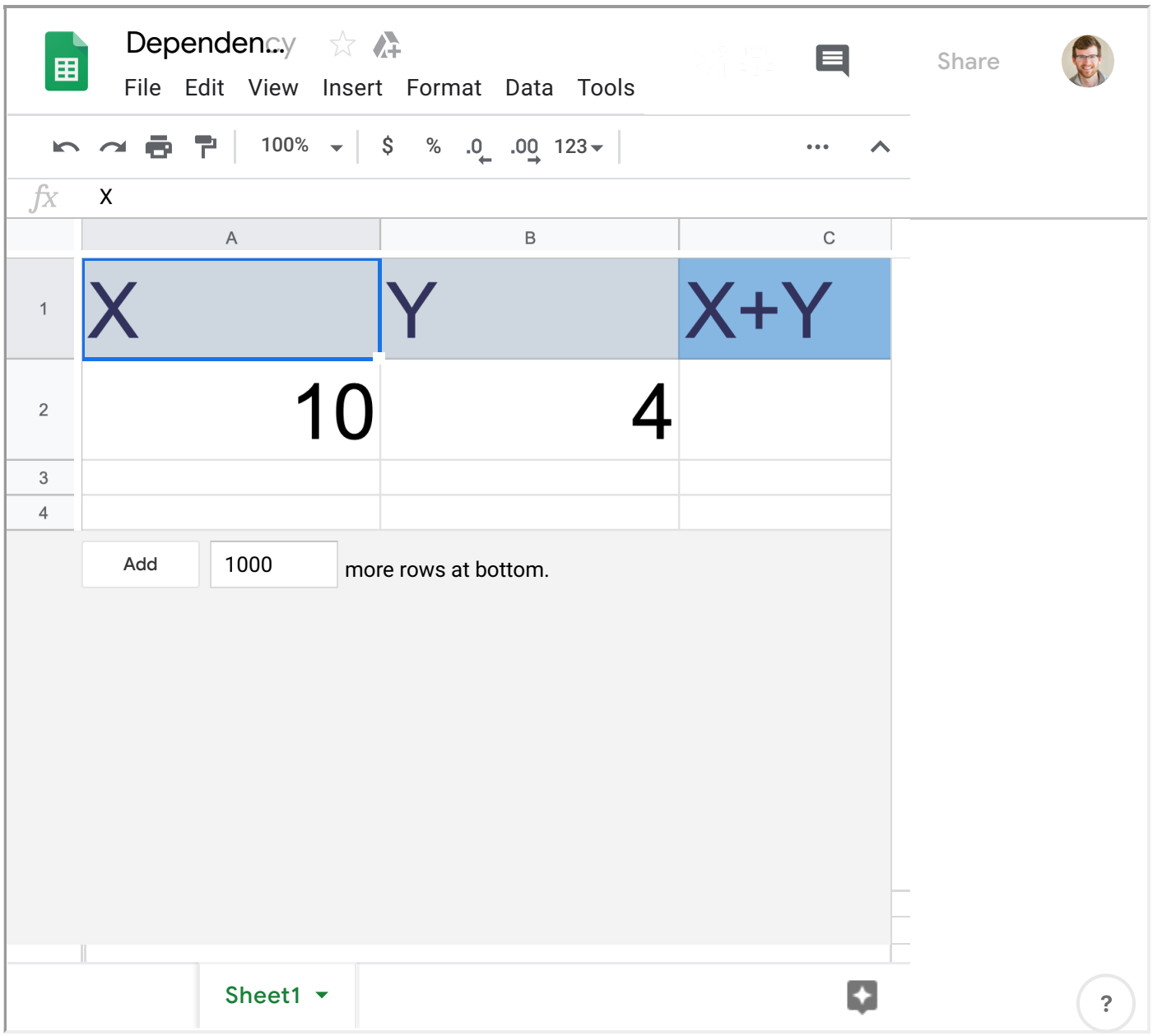
◉ Nunca ○ A veces ○ Todos los días!

Selecciona tu respuesta

▸ `Response {}`

Jupyter Notebook

Click para notebook de Python nuevo



Observable

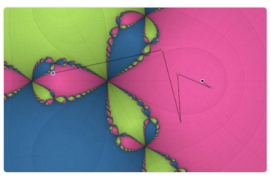
NOTEBOOK + RASTREAR DEPENDENCIAS

Para que lo usamos?	Para que no lo usamos?
Explorar data y compartir resultados	Componente de servidor
Redacciones interactivos	Correr en producción
Hacer un prototipo de cliente	Usar con cosas confidenciales*
Construir visualizaciones	

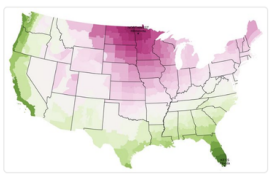
Ejemplos padres

Arte | Ciencia | Sonidos | Astronomía

POPULAR THE WEEK OF AUG 18 – AUG 25



01 **Finding Roots in the Complex Plane**
Ricky Reusser Aug 27

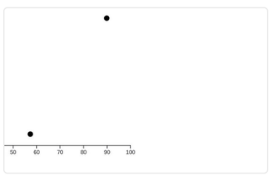


02 **Where The Seasons Are**
Mike Bostock Aug 25

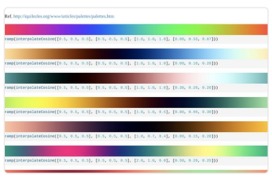
03 **Turbo**
Mike Bostock Aug 21

04 **Pieces of a Map**
Jeshurun Hembd Aug 31

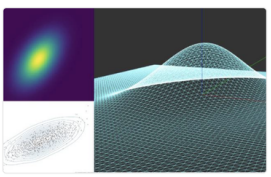
05 **Luv and HCL**
Mike Bostock Aug 20



06 **Marks and Channels in D3**
John Alexis Guerra Gómez Aug 21



07 **Cosine Color Schemes**
Mike Bostock Aug 21



08 **Multivariate Normal Distribution**
sw1227 Aug 25

Dibujar un Rectángulo

Empecemos con algo simple: hacer un rectángulo. Vamos a usar el Canvas (mira Mozilla para mas detalles).

Las Dependencias

Observable está rastreando las dependencias. Eso significa que podemos poner el ancho en su propio valor en otro cell. Intente cambiar el valor y mira que el rectángulo cambia inmediatamente.

```
rectWidth = 100
```

Deslizadores

Un valor en su propio cell es padre, pero podemos mejorar con un deslizador!

205

```
{
  const context = DOM.context2d(
    width /* special ObservableHQ value for width of page */,
    150 /* height of canvas */
  );

  const rectWidth = 150;
  const rectHeight = 100;

  context.fillStyle = "#424770"
  context.fillRect(
    10 /* top left x */ ,
    10 /* top left y */,
    rectWidth,
    rectHeight
  );

  return context.canvas
}
```

4

cupTopRadius

4

cupBottomRadius

```
rectWidth = 100 /* Cambiáme! */
```

Conclusión

- Observable = Notebook interactivo con los poderes de spreadsheet
- Es fantástico para explorar, visualizar, y aprender

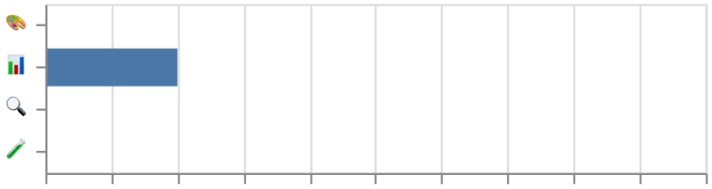
?

```
viewof rectWidth2 = html`<input type=range min=10 max=400>`
```

```
rectWidth2 /* Este cell reporta el valor */
```

PREGUNTAS?

Y QUE VAS A CONSTRUIR?



Utilizas el REPL?

○ Arte ◉ Gráficos ○ Investigaciones ○ Prototipos

Selecciona tu respuesta

```
▸ Response {}
```

```
drawBubbleTea(new BubbleTea({
    cup: new Cup({
      topRadius: cupTopRadius,
      bottomRadius: cupBottomRadius
    })})).node();
```

Appendix

```
import {slider} from @jashkenas/inputs
```

```
<style>
```

```
fetch(`https://bolder-condorraptor.glitch.me/vote/3/${r3}/${my_id}`)
```

```
spacer = ƒ(height)
```

```
$ = ƒ(selector, context)
```

```
slide = ƒ()
```

```
spacer(800)
```

```
spacer(800)
```

```
slide_style = html`<style>
.slide {
  width: calc(100% + 28px);
  margin: 0 -14px;
  padding: 10%;
  box-sizing: border-box;
  background: #424770;
  color: #CED7DF;
  min-height: 65vw;
  font-size: 3vw;
  font-family: "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue",
Helvetica, Arial, "Lucida Grande", sans-serif;
   font-weight: 300;
  line-height: 1.15;
  display: flex;
  align-items: center;
}

.slide a {
  color: #68D4F8;
}

.slide p,
.slide pre,
.slide img {
  max-width: 100%;
  color: #CED7DF;
  font-size: 2.8vw;
}

.slide h1 {
  color: #FFFFFF;
}

.slide h2 {
  color: #FFFFFF;
  text-transform: uppercase;
  font-size: 3vw;
}
```

```
}

.slide h3 {
  color: #FFFFFF;
  text-transform: uppercase;
  font-size: 2.5vw;
}

.slide strong {
  color: #FCD669;
}

.slide em {
  color: #FFC7EE;
}

.slide--img {
  max-width: none;
  padding: 0;
}

.slide ul {
  font-size: 3vw;
}

.slide table {
  font-size: 3vw;
}

.slide > * {
  width: 100%;
}

</style>`
```

- Analyse data for science, journalism, education, etc

```
spacer = function(height) {
  const context = this ? this.getContext("2d") : DOM.context2d(
    width /* special ObservableHQ value for width of page */,
    height /* height of canvas */
  );

  return context.canvas
}
```

```
slide = {
  function slide() {
    const container = document.createElement("div");
    container.className = "slide";
    container.appendChild(md.apply(this, arguments));
    return container;
  }
  slide.img = function(strings) {
    const img = new Image;
    let string = strings[0] + "", i = 0, n = arguments.length;
    while (++i < n) string += arguments[i] + "" + strings[i];
    img.src = string.trim();
    img.className = "slide slide--img";
    return img;
  };
  slide.cell = function() {
    const container = document.createElement("div");
    container.className = "slide2";
    container.style.paddingTop = "20px";
    container.appendChild(md.apply(this, arguments));
    return container;
  }
  return slide;
}
```

```
{
  let slide_cells = $(".slide, .slide2")
  // let slide_cells = $(".observablehq")

  //let all_cells = slide_cells.concat(document.getElementsByClassName("slide2"));
  //document.cells = all_cells;
  return document.cells = slide_cells;
}
```

```
{
document.current=0;
document.onkeyup = function(e) {
  if (e.which === 39 || e.which === 37) {
    if (e.which === 39){
      document.current+=1;
```

```
      document.current+=1;
      if (document.current < 0) { document.current = 0; }
    }
    if (e.which === 37){
        document.current -= 1;
            if (document.current >= document.cells.length) { document.current =
document.cells.length - 1; }
    }
    var elmnt =   document.cells[document.current]
    elmnt.scrollIntoView();
  }
}
}
```

```
questionGraph = f(qlabel)
```

```
vegalite = f(spec)
```

```
choices = ▶ Object {1: Array(3), 2: Array(3), 3: Array(4)}
```

```
big_o = html`<svg viewBox="0 0 25 28" width="250" height="300" aria-
label="Observable" fill="currentColor" ><path d="M12.5 22.6667C11.3458 22.6667
10.3458 22.4153 9.5 21.9127C8.65721 21.412 7.98339 20.7027 7.55521 19.8654C7.09997
18.9942 6.76672 18.0729 6.56354 17.1239C6.34796 16.0947 6.24294 15.0483 6.25
14C6.25 13.1699 6.30417 12.3764 6.41354 11.6176C6.52188 10.8598 6.72292 10.0894
7.01563 9.30748C7.30833 8.52555 7.68542 7.84763 8.14479 7.27274C8.62304 6.68378
9.24141 6.20438 9.95208 5.87163C10.6979 5.51244 11.5458 5.33333 12.5
5.33333C13.6542 5.33333 14.6542 5.58467 15.5 6.08733C16.3428 6.588 17.0166 7.29733
```

```
17.4448 8.13459C17.8969 8.99644 18.2271 9.9103 18.4365 10.8761C18.6448 11.841
18.75 12.883 18.75 14C18.75 14.8301 18.6958 15.6236 18.5865 16.3824C18.4699
17.1702 18.2639 17.9446 17.9719 18.6925C17.6698 19.4744 17.2948 20.1524 16.8427
20.7273C16.3906 21.3021 15.7927 21.7692 15.0479 22.1284C14.3031 22.4876 13.4542
22.6667 12.5 22.6667ZM14.7063 16.2945C15.304 15.6944 15.6365 14.864 15.625
14C15.625 13.1073 15.326 12.3425 14.7292 11.7055C14.1313 11.0685 13.3885 10.75
12.5 10.75C11.6115 10.75 10.8688 11.0685 10.2708 11.7055C9.68532 12.3123 9.36198
13.1405 9.375 14C9.375 14.8927 9.67396 15.6575 10.2708 16.2945C10.8688 16.9315
11.6115 17.25 12.5 17.25C13.3885 17.25 14.124 16.9315 14.7063 16.2945ZM12.5
27C19.4031 27 25 21.1792 25 14C25 6.82075 19.4031 1 12.5 1C5.59687 1 0 6.82075 0
14C0 21.1792 5.59687 27 12.5 27Z" fill="currentColor"></path></svg>`
```

```
io = require('https://bundle.run/socket.io-client@2.2.0')
```

```
msgs = Generators.observe(notify => {
  const msgs = [];

  let socket = io('ws://bolder-condorraptor.glitch.me/');
  socket.on('chat message', function(msg){
    console.log(msg);
    msgs.push(msg);
    notify(msgs);
  });
  return () => socket.close();
})
```

```
questionGraph = function(qlabel) {

const deduped = new Map();
  msgs.filter((msg) => (msg.q == qlabel)).map((msg) => {
    deduped.set(msg.id, {choice: choices[qlabel][parseInt(msg.c)], ...msg});
  });

  const vals = Array.from(deduped.values());

  return vegalite({
```

```
  "width": (width* 0.6),
    "config": {"axis":{"labelColor": "white", "titleColor": "white"}},
   "mark": "bar",
    "encoding":
    {
      "x": {
      "aggregate": "distinct",
        "field": "id",
        "type": "quantitative",
        "axis": {title: "personas"},
        "scale": {"domain":[0, Math.max(10, vals.length*1.2)]}
      },
      "y": {"field": "choice", "type": "nominal",
            "axis": {title: "selección"},
            "scale": {"domain": choices[qlabel]}
           }
    },
  // "height": 100,
  // "padding": {"top": 30, "left": 20, "right": 20, "bottom": 50},
  // "autosize": "none",
  "data": {"values": vals},

})
}
```

```
vegalite = require("@observablehq/vega-lite@0.1")
```

```
choices = {
  return {
    "1": ["Nunca", "A veces", "Todos los días!"],
    "2": ["Nunca", "A veces", "Todos los días!"],
    "3": ["🎨 ", "📊 ", "🔍 ", "🧪 "],
  }
};
```