In [2]:
```
%matplotlib inline
```

Presentaciones y encuestas interactivas con jupyter notebooks y RISE

pycon Colombia 2020

Sebastian Flores

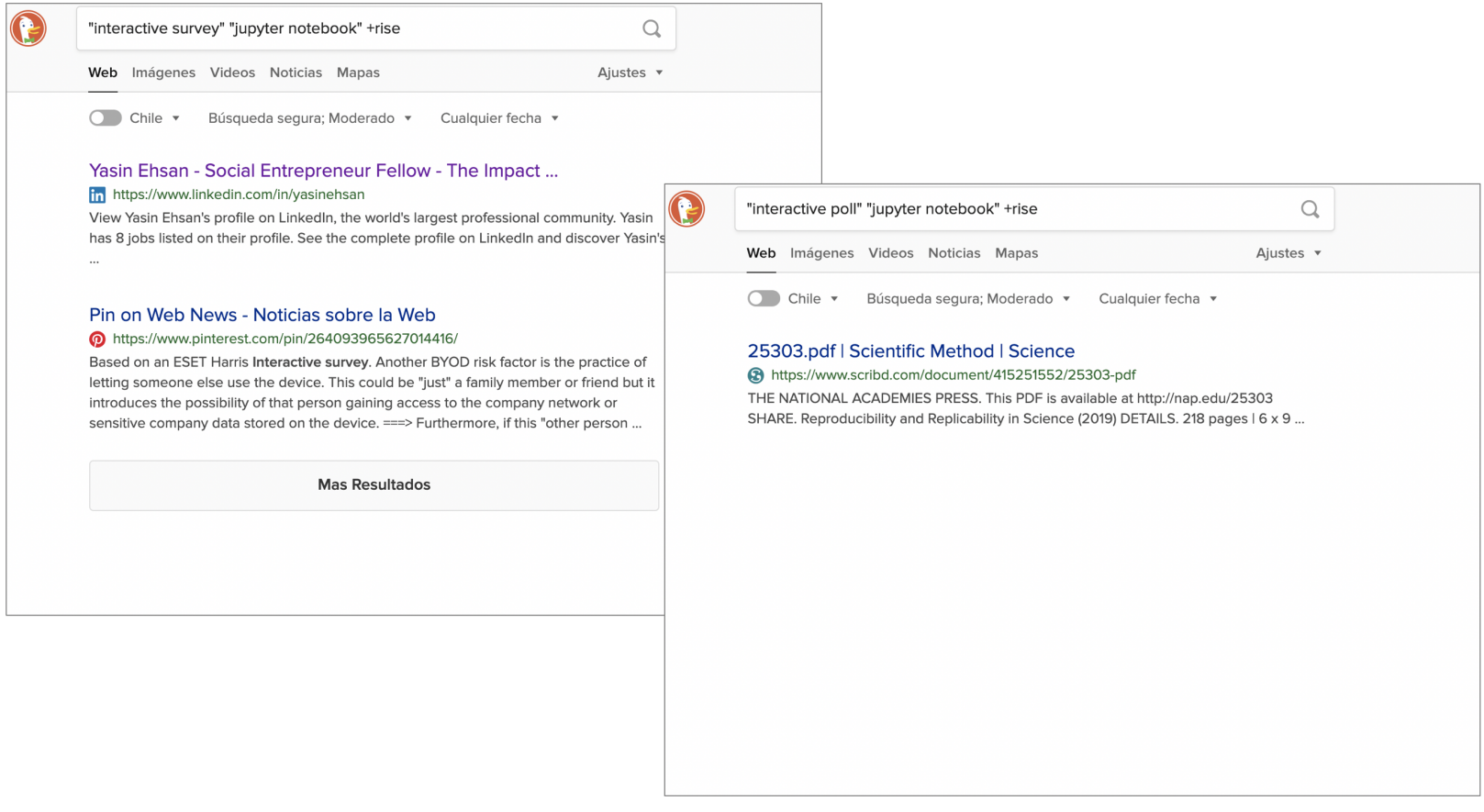
07 Febrero 2020

Observaciones:

- sebastiandres en twitter y github.
- Presentación en github.com/sebastiandres/charlas (github.com/sebastiandres/charlas)
- Opiniones y críticas a título personal.

Introducción

¿Porqué esta presentación?



¿Se pueden hacer encuestas interactivas en presentaciones con jupyter notebooks?

Agenda

1. Sobre hacer presentaciones con RISE.
2. Solución 1: Quick & dirty.
3. Solución 2: ¡Pero quiero los datos!
4. Solución 3: No confío en nadie.

0.1 Sobre hacer presentaciones con jupyter y RISE

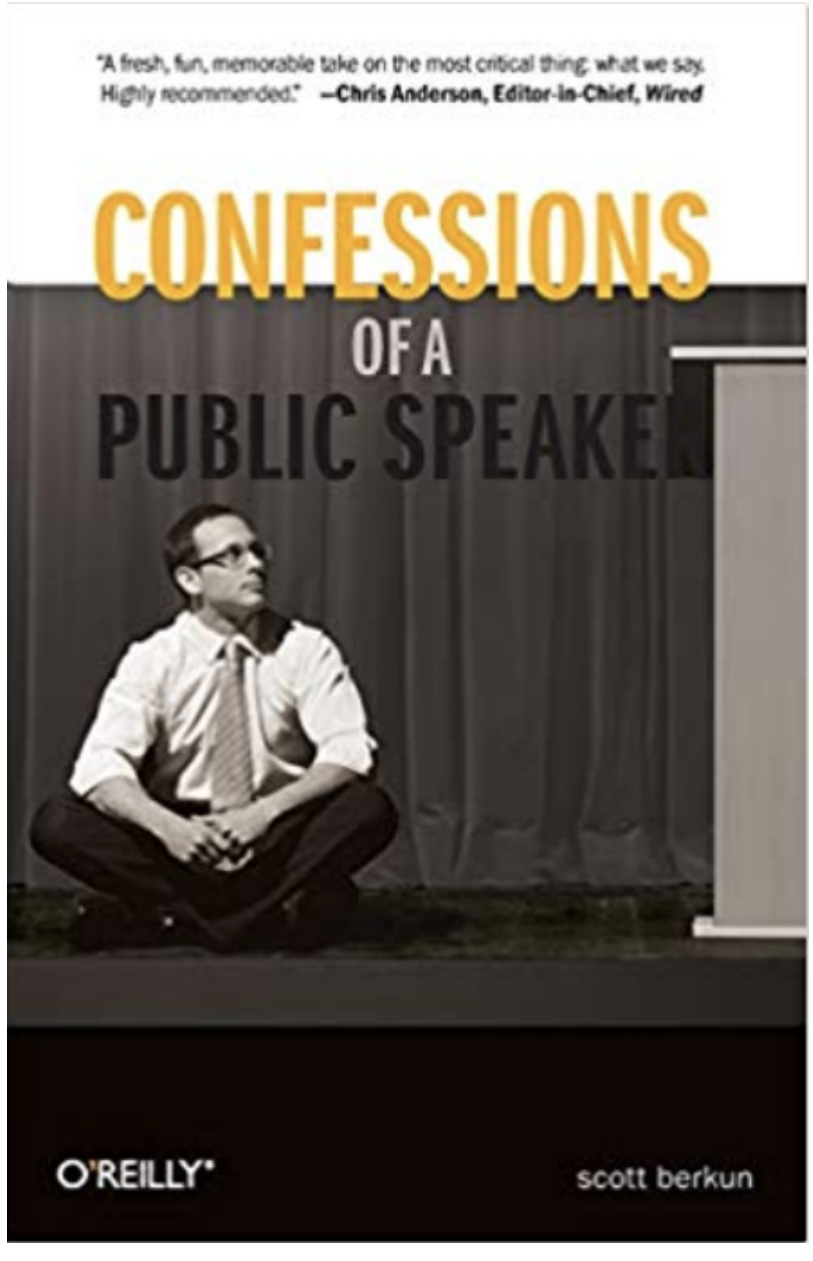
¿Consejos sobre hacer presentaciones en general?

- Contenido pensado en tu audiencia.
- Ensayar, ensayar, ensayar.
- Incrementar el desafío.

0.2 Sobre hacer presentaciones con jupyter y RISE

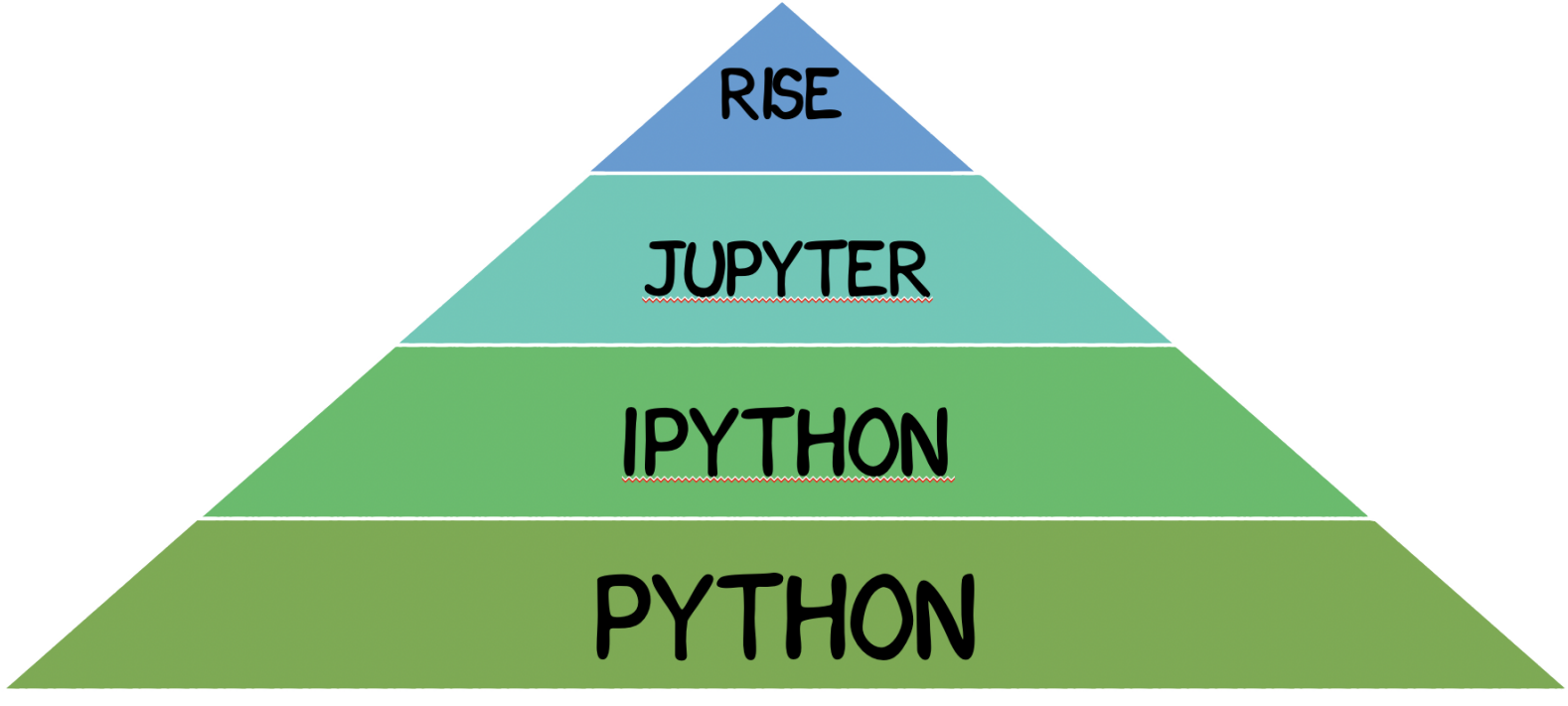
Si forma parte importante de tu día a día, investiga y aprende a hacer mejores presentaciones.

Libro que considero muy bueno: Confessions of a public speaker - Scott Berkum.


"A fresh, fun, memorable take on the most critical thing: what we say. Highly recommended." —Chris Anderson, Editor-in-Chief, Wired
CONFESSIONS
OF A
PUBLIC SPEAKER
O'REILLY
scott berkun

0.3 Sobre hacer presentaciones con jupyter y RISE

¿jupyter notebook? ¿RISE?



0.3 Sobre hacer presentaciones con jupyter y RISE

¿Qué es la extensión RISE?

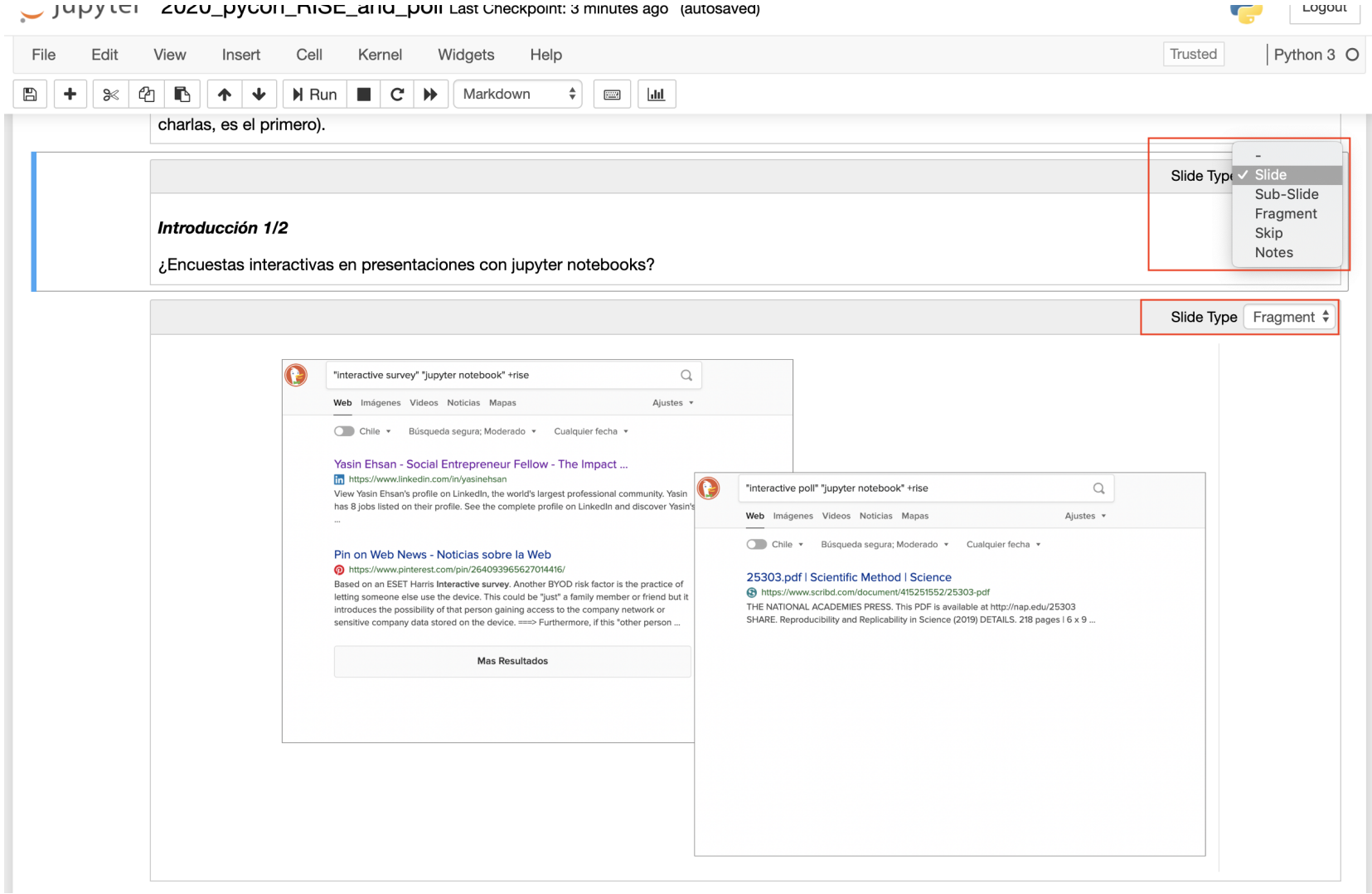
Una celda de jupyter notebook se puede clasificar (adicionalmente al code/markdown) como:

- *Slide*: Diapositiva normal.
- *Fragment*: Fragmento de diapositiva.
- *Sub-slide*: Opcional.
- *Skip*: No mostrar.
- *Notes*: Notas.

Toda la info en https://rise.readthedocs.io/ (https://rise.readthedocs.io/)

0.3 Sobre hacer presentaciones con jupyter y RISE

¿Qué es la extensión RISE?



0.4 Sobre hacer presentaciones con jupyter y RISE

Ok, pero... ¿porqué? ¿Porqué? ¿PORQUÉ?

- Tener una presentación autocontenida: los ejemplos (código) queda en jupyter notebook.
- Corregir directamente cuando sea necesario.
- Hacer participar a la audiencia.

0.4 Sobre hacer presentaciones con jupyter y RISE

Sí, incluso ejecutar código en python

In [4]:
```
x = 200
y = 200
x is y
```

Out[4]:
```
True
```

In [6]:
```
a = 300
b = 300
a is b
```

Out[6]:
```
False
```

In [7]:
```
def cheeseburger(n):
    return ["bread"] + ["meat", "cheese"]*n + ["bread"]

cheeseburger(3)
```

Out[7]:
```
['bread', 'meat', 'cheese', 'meat', 'cheese', 'meat', 'cheese', 'bread']
```

0.5 Sobre hacer presentaciones con jupyter y RISE

¿Que he aprendido?

- Usa "Ctrl +" y "Ctrl -" para controlar aspecto.
- Entrega parte del control de contenido al público. Usa un teclado+mouse inalámbrico.
- Plantea desafíos de código incompleto pero entrega propuestas de solución.
- `git` es tu amigo.
- Improvisa. Nunca será como planificaste.

0.6 Descripción del Problema

¿Qué queremos?

Realizar encuesta interactiva sin salir del modo presentación (en RISE).

¿Porqué?

Hacer encuestas para obtener retroalimentación de la audiencia cuando importa, sin cambiar pantalla o sistema.

1.7 Descripción del Problema

¿Se puede?

Sí, y de varias maneras distintas.

¿Cómo?

IFrames.

0.9 Descripción del Problema

Nuestro héroe se llama IFrame.

In [7]:
```
from IPython.display import IFrame
IFrame("https://es.wikipedia.org/wiki/Iframe", width=1000, height=500) #https://es.wikipedia.org/wiki/Iframe
```

Out[7]:

WIKIPEDIA

iframe

Iframe (por *inline frame* o *marco incorporado* en inglés) es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal. Fue introducido en el navegador Microsoft Internet Explorer en 1997 y durante mucho tiempo solo fue soportado en este navegador, la etiqueta *Iframe* actualmente es ya aceptada por la W3 como un elemento estándar y es ampliamente soportado por gran variedad de navegadores.

```
<html>
  <head>
    <title>IFrames</title>
  </head>

  <body>
    <iframe src="http://www.example.com/" name="SubHtml"
      width="400" height="500" scrolling="auto" frameborder="1">
      <p>Texto alternativo para navegadores que no aceptan iframes.</p>
    </iframe>
  </body>
</html>
```

Los iFrames admiten diversos atributos como "AllowTransparency", que deberemos establecer a "true",

```
<iframe src="paginaACargar.html"....AllowTransparency="true"....>,
```

Otros:

0.10 Descripción del Problema

Más ejemplos ...

In [8]:
```
from IPython.display import IFrame
IFrame("https://www.python.org/", width=800, height=300)
```

Out[8]:

No siempre funciona:

- https://www.python.org/ (https://www.python.org/)
- google, facebook, twitter, ...

En general a los sitios (sobre todo los que requieren autentificación y desean hacer tracking de usuarios) bloquean los IFrames.

1.1 Problema: Realizar encuestas interactivas.

Solucion nivel 1:

KISS: Keep IT Simple, Stupid.

Podemos usar IFrame con los servicios de alguna compañía especializada en encuestas interactivas:

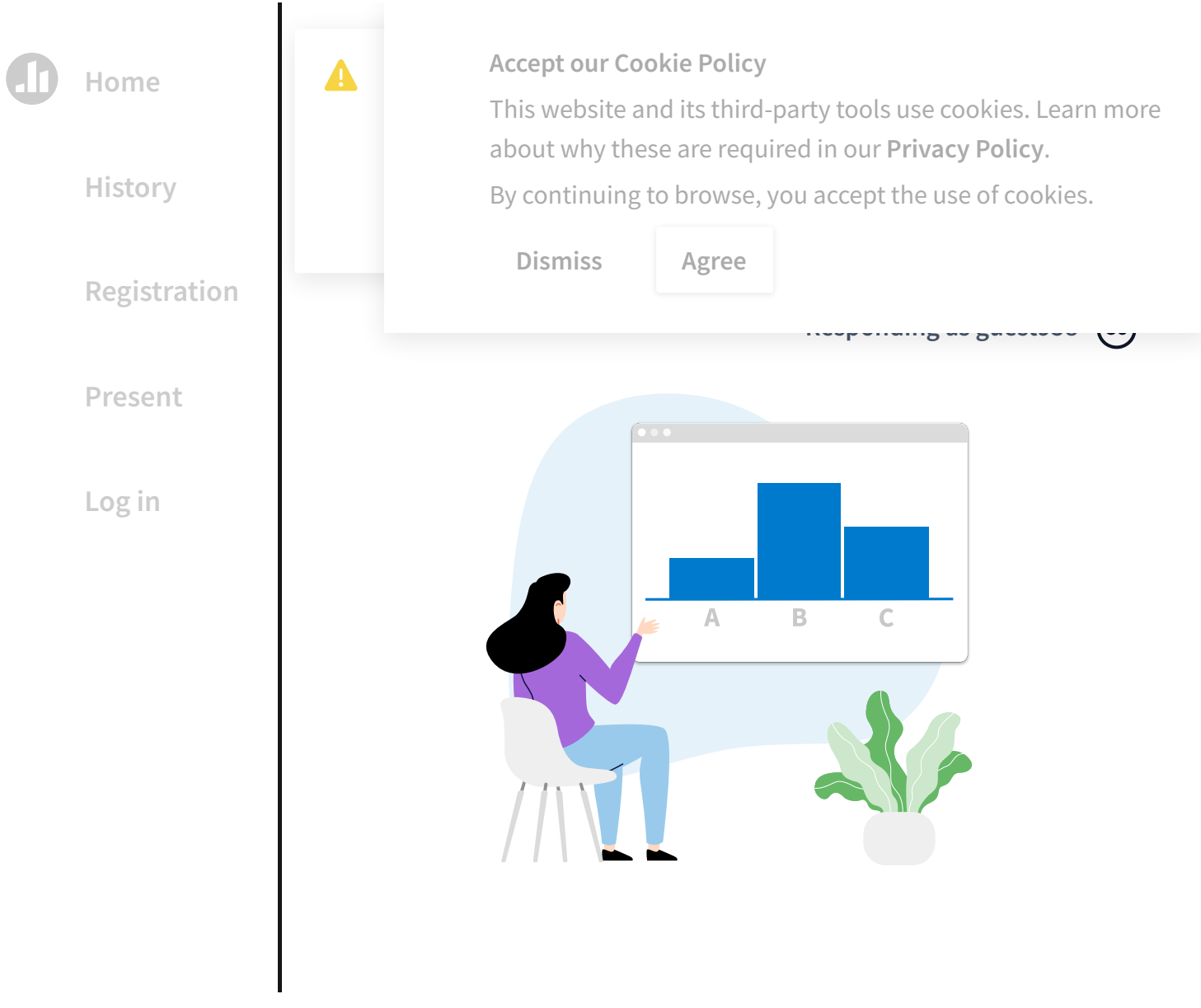
- Polleverywhere
- Mentimeter
- Muchos, muchos otros...

1.2 Solución nivel 1: Encuesta

Probemos este método con una encuesta simple:

In [1]:
```
from IPython.display import IFrame
IFrame("https://pollev.com/sebastianflo711", width=800, height=600)
```

Out[1]:



1.3 Solución nivel 1:

Link encuesta:

https://pollev.com/sebastianflo711

¿Qué debe cumplir un sitio de encuestas interactivas?

- Encuesta acccesible por url pública, incrustable por iframe.
- Gráfico dinámico accesible por url pública, incrustable por iframe.

1.3 Solución nivel 1:

Veamos los resultados de la encuesta anterior.

In [12]:
```
from IPython.display import IFrame
IFrame("https://www.polleverywhere.com/multiple_choice_polls/7Xwy472JbCrk3fITm12XM", width=800, height=600)
# Observación: ¡requiere estar autentificado como dueño de la encuesta!
```

Out[12]:

1.4 Solución nivel 1: Resumen

¿Cómo se debe usar?

1. Seleccionar algún proveedor.
2. Crear y configurar la encuesta previamente.
3. Obtener el enlace de las preguntas, e incrustarla con un iframe.
4. Obtener el gráfico de las respuestas, e incrustarla con un iframe.
5. Probar que funcione correctamente. Borrar respuestas de prueba.

1.5 Pausa

¿Que viene antes, el huevo o la gallina?

In [13]:
```
# Solución al problema
# python3 permite usar unicode
sorted(['\N{EGG}', '\N{CHICKEN}'])
```

Out[13]:
```
['🐔 ', '⚪ ']
```

2.1 Problema: Realizar encuestas interactivas.

Solución Nivel 2

Una segunda opción que requiere un poco más de configuración pero entrega más control, es usar un servicio de encuestas tradicional:

- Microsoft Forms
- Google Forms
- Survey Monkey
- Typeform
- Muchas otras...

Se diferencia que nosotros podemos hacer los gráficos porque podemos descargar los datos.

2.2 Solución Nivel 2: Encuesta + Datos

Probemos este método con otra encuesta simple:

In [14]:
```
long_url = "https://forms.office.com/Pages/ResponsePage.aspx?id=zu7OdUTRPU-clJ5rQCX8_4qs5cX1Y7dFhVdiCz848sBUNFU3UzU3OTNHVDhWNURSMEs2WDBUMDdCTi4u"
short_url = "https://bit.ly/2uYDdhm"
iframe_options = {"width":800, "height":600}
IFrame(short_url, **iframe_options)
```

Out[14]:

2.3 Solución Nivel 2:

Link encuesta:

https://bit.ly/2uYDdhm

¿Qué debe cumplir un sitio de encuestas nivel 2?

- Encuesta accesible al público mediante un enlace abierto. Puede que no sea incrustable con un iframe.
- Opcionalmente, que cuente con gráficos en línea.
- Datos disponibles y puedan descargarse.
- Crear una función (en Python) para descargar e interpretar los datos.
- Probar que la encuesta funcione. Borrar las respuestas de prueba.

2.4 Solución Nivel 2: Respuestas

In [16]:

```
long_url = "https://forms.office.com/Pages/AnalysisPage.aspx?id=zu7OdUTRPU-clJ5rQCX8_4qs5cX1Y7dFhVdiCz848sBUNFU3UzU3OTNHVDhWNURSMEs2WDBUMDdCTi4u&AnalyzerToken=ND54U8Erls95gHxjbyWus4LzH6FPyZ35"
short_url = "https://bit.ly/37We2dE"
iframe_options = {"width":800, "height":600}
IFrame(short_url, **iframe_options)
```

Out[16]:

2.4 Solución Nivel 2: Respuestas (versión 2)

- Descargamos la planilla de datos con el resultado desde algún enlace (https://uplanner my.sharepoint.com/:x:/r/personal/sebastian_flores_u-planner_com/_layouts/15/Doc.a sourcedoc=%7B7D9BF52E-2931-4702-86ED-00792E617ECF%7D&file=2020_02_09_encuesta2_pycon.xlsx&action=default&mobi

In [11]:
```
# Alternative to ls data/*.xlsx
import glob
glob.glob("data/*.xlsx")
```

Out[11]:
```
['data/2020_02_08_encuesta2_pycon.xlsx']
```

In [12]:
```
import pandas as pd
df_dict = pd.read_excel("data/2020_02_08_encuesta2_pycon.xlsx", sheet_name=None)
df_dict.keys()
```

Out[12]:
```
odict_keys(['Hoja1', '_56F9DC9755BA473782653E2940F9', 'Form1'])
```

In [13]:
```
df = df_dict["Form1"]
question = df.columns[-1]
print(question)
```

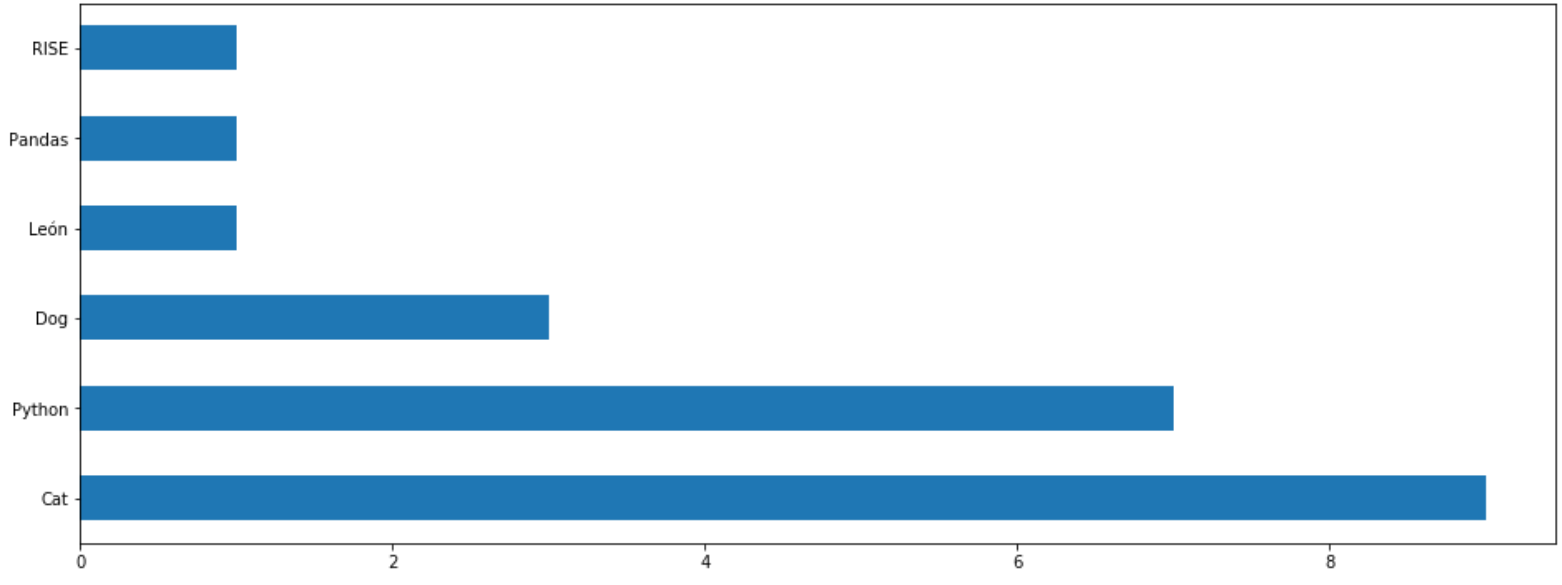
```
Which one is the best pet?
```

2.5 Solución Nivel 2: Respuestas

In [15]:
```
question_df = df[question]
answer_counts = question_df.value_counts()
answer_counts
```

Out[15]:
```
Cat       9
Python    7
Dog       3
León      1
Pandas    1
RISE      1
Name: Which one is the best pet?, dtype: int64
```

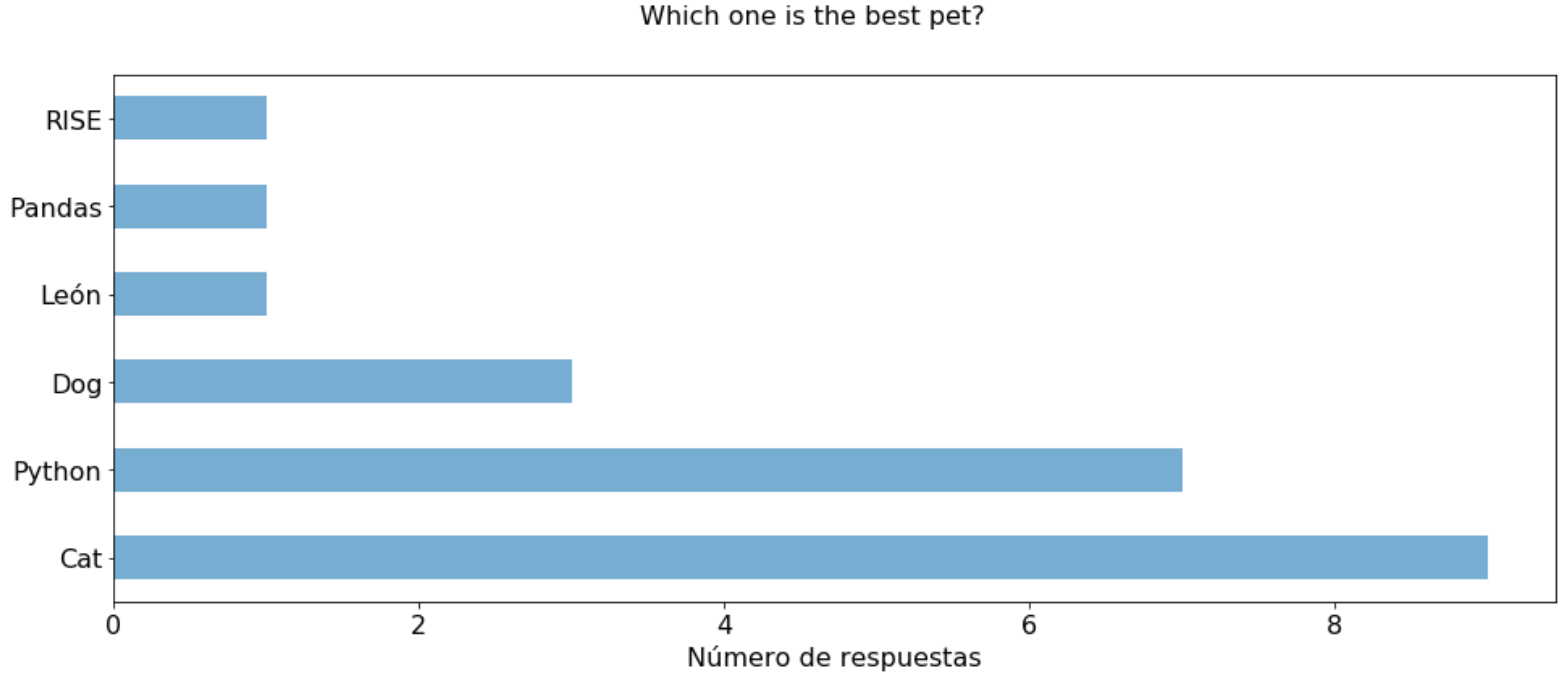
In [16]:
```
from matplotlib import pyplot as plt
answer_counts.plot.barh(figsize=(16,6))
plt.show()
```



2.6 Solución Nivel 2: Respuestas

Hagamos un gráfico un poco mejor.

In [17]:
```
from matplotlib import pyplot as plt
answer_counts.plot.barh(figsize=(16,6), fontsize=16, alpha=0.60)
plt.suptitle(question, fontsize=16)
plt.xlabel("Número de respuestas", fontsize=16)
plt.ylabel("");
```



2.7 Solución nivel 2: Resumen

1. Elegir algún proveedor.
2. Crear la encuesta.
3. Obtener la url pública de las preguntas, e incrustarla en un iframe.
4. Obtener la url de los datos de las respuestas.
5. Usar una api o descargar los datos.
6. Graficar desde python para interpretar los datos.

2.1 Problema: Realizar encuestas interactivas.

Solución Nivel 3

La solución más compleja es por supuesto hacer el sistema completo:

- **Base de datos**: donde almacenar la encuesta y las respuestas.
- **Frontend**: para realizar la encuesta (mostrar preguntas, recolectar respuestas, mostrar gráficos).
- **Backend**: interactuar con la base de datos.

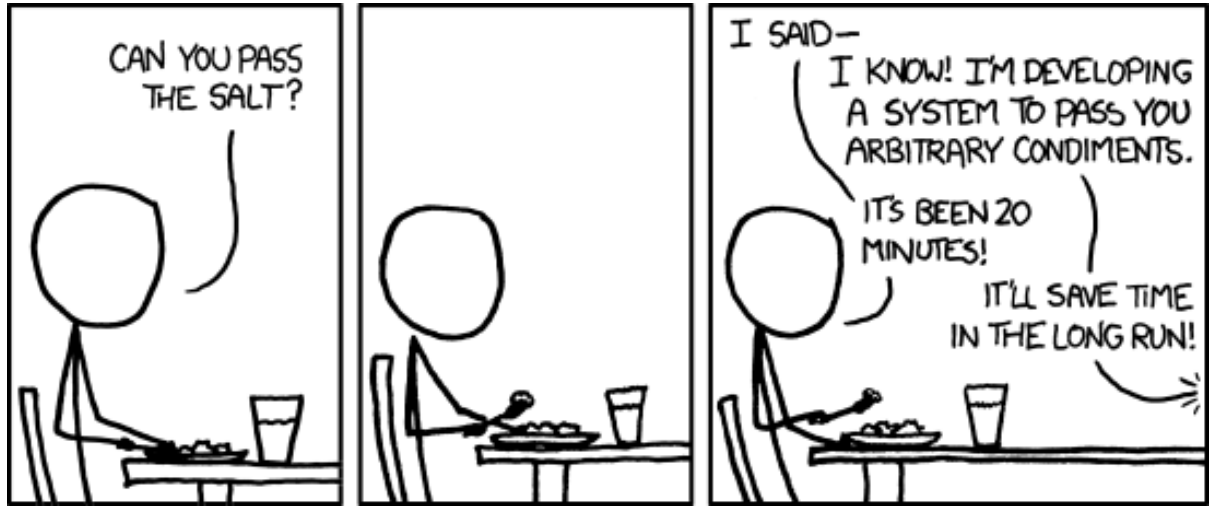
En este caso: mysql, flask, flask-mysql y pandas, con algo de html y css, y una librería de javascript para graficar.

Solución Nivel 3

¿Porqué alguien voluntariamente se sometería a este tormento?

Filosofía xkcd:

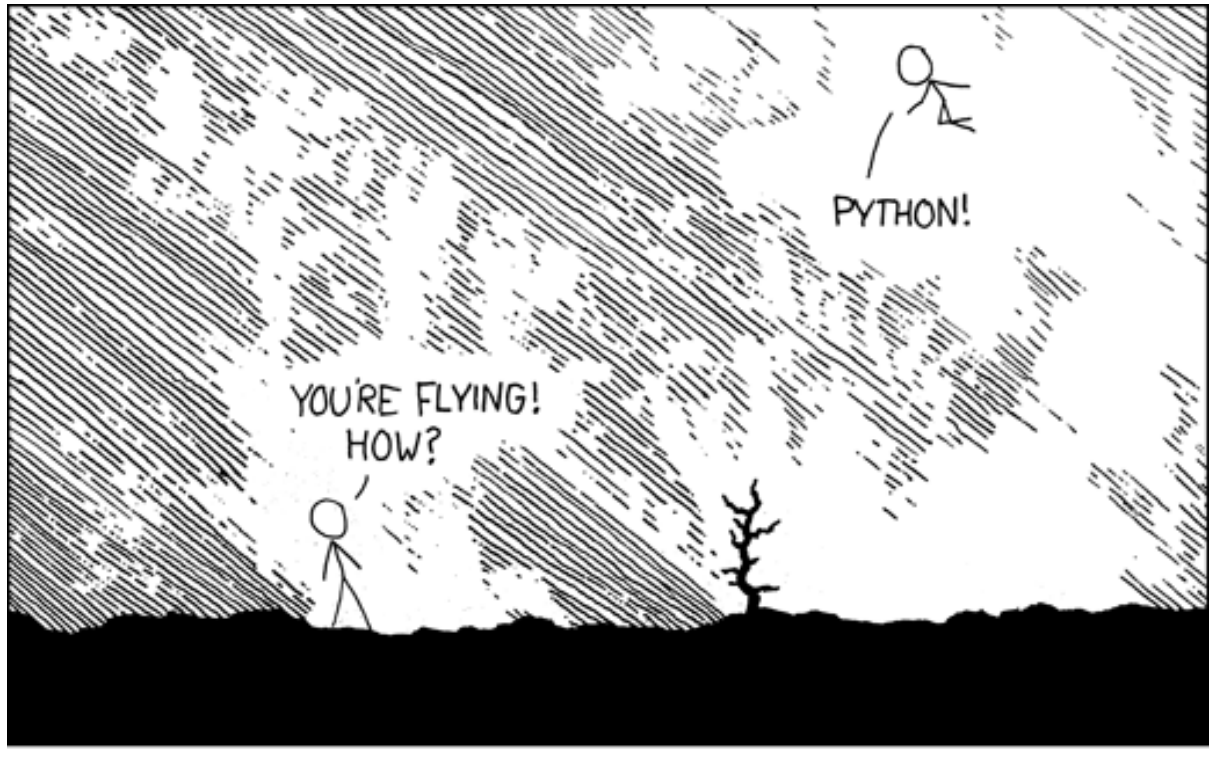
¿Porqué hacer algo simple cuando podrías hacer algo extremadamente complejo *de la manera más simple posible*, aprendiendo mucho en el camino?



Solución Nivel 3

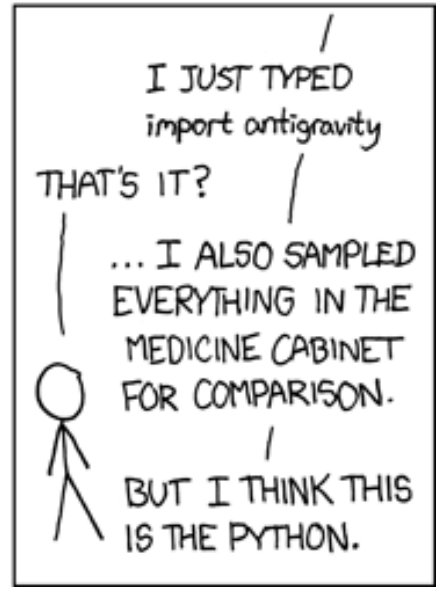
¿xkcd en python?

In [24]:
```
import antigravity
```


PYTHON!
YOU'RE FLYING! HOW?


I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"


I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?


I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Solución Nivel 3

In [19]:
```
from IPython.display import IFrame
IFrame("http://localhost:5000/", width=1200, height=800)
```

Out[19]:

Solución Nivel 3

El código de polite se encuentra en https://github.com/sebastiandres/surveys_with_flask_and_xkcd_charts (https://github.com/sebastiandres/surveys_with_flask_and_xkcd_charts).

Hace uso de muchos recursos públicamente disponibles (además de mysql y python+librerías):

- Font xkcd.ttf de ipython, https://github.com/ipython/xkcd-font (https://github.com/ipython/xkcd-font)
- Librería xkcd_chart en javascript: https://timqian.com/chart.xkcd/ (https://timqian.com/chart.xkcd/)
- Una cantidad vergonzosamente grande de búsquedas en internet y stackoverflow.

Solución Nivel 3

Una de las cosas que más me llamó la atención fue cómo hacer una función que pudiese transformar un texto markdown en la serie de preguntas y opciones de respuesta.

Requería definir convenciones:

- La pregunta es todo lo que va antes de `:`
- Las preguntas de respuesta única se definen por `*`
- Las preguntas de respuesta única se definen por `^`

Otra opción podría haber sido `-` y `+`.

Es una mala opción usar `o` y `m`.

Solución Nivel 3

In [26]:

```
def markdown_parser(my_text):
    if my_text.count(":")!=1:
        print("Cannot parse, there's an error in the format")
        return {"is_format_ok":False, "markdown_str":my_text}
    single_option = 0
    multiple_option = 0
    if ("* " in my_text):
        single_option = 1
        split_char = "*"
    if ("^ " in my_text):
        multiple_option = 1
        split_char = "^"
    # If both False or both True, simultaneoulsy, there's an error
    if single_option==multiple_option:
        print("Cannot parse, there's an error in the format")
        return {"is_format_ok":False, "markdown_str":my_text}
    question_str, answer_str = my_text.split(":")
    question = question_str.strip()
    answer_list = [_.strip() for _ in answer_str.split(split_char)[1:6]] # Skip th
e empty string, reach to the fifth existing one
    answer_list = answer_list + ["" for _ in range(5-len(answer_list))] # Fill wit
h empty ones if needed
    question_type = single_option*"radio"+multiple_option*"checkbox" # This is the
html convention
    # Create the dict
    md_dict = {}
    md_dict["is_format_ok"] = True
    md_dict["markdown_str"] = my_text
    md_dict["type_str"] = question_type
     md_dict["question_str"] = question
    md_dict["option_1_str"] = answer_list[0]
    md_dict["option_2_str"] = answer_list[1]
    md_dict["option_3_str"] = answer_list[2]
    md_dict["option_4_str"] = answer_list[3]
    md_dict["option_5_str"] = answer_list[4]
    return md_dict
```

Solución Nivel 3

In [28]:
```
markdown_parser("Q: ^ A * B * C * D * E * F * G ")
```

```
Cannot parse, there's an error in the format
```

Out[28]:
```
{'is_format_ok': False, 'markdown_str': 'Q: ^ A * B * C * D * E * F * G '}
```

In [ ]:
```
markdown_parser("""Q: ^ A ^ B ^ C ^ D ^ E ^ F ^ G ^ H""")
```

Conclusión: Encuestas nivel 1

	Ventaja	**Desventaja**
Complejidad	Trivial	-
Precio	Versión limitada gratis	Versión pagada con más funcionalidades
Gráficos	Gráficos predefinidos	Gráficos predefinidos
Datos	-	Sin accceso a datos

Conclusión: Encuestas nivel 2

	Ventaja	**Desventaja**
Complejidad	Simple	
Precio	Versión gratis es suficiente	
Gráficos	Configurables	Requieren código
Datos	Con accceso a datos	

Conclusión: Encuestas nivel 3

	Ventaja	**Desventaja**
Complejidad		Bastante complejo. Toma tiempo. Deployment puede ser complejo.
Precio	Hosting gratis es suficiente	Time=Money?
Gráficos	Configurables	Hay que configurarlos completamente.
Datos	Con accceso completo a datos	Hay que almacenar los datos.

Conclusión

¿Preguntas?

Encuesta sobre la charla:

https://bit.ly/2UzrYGU