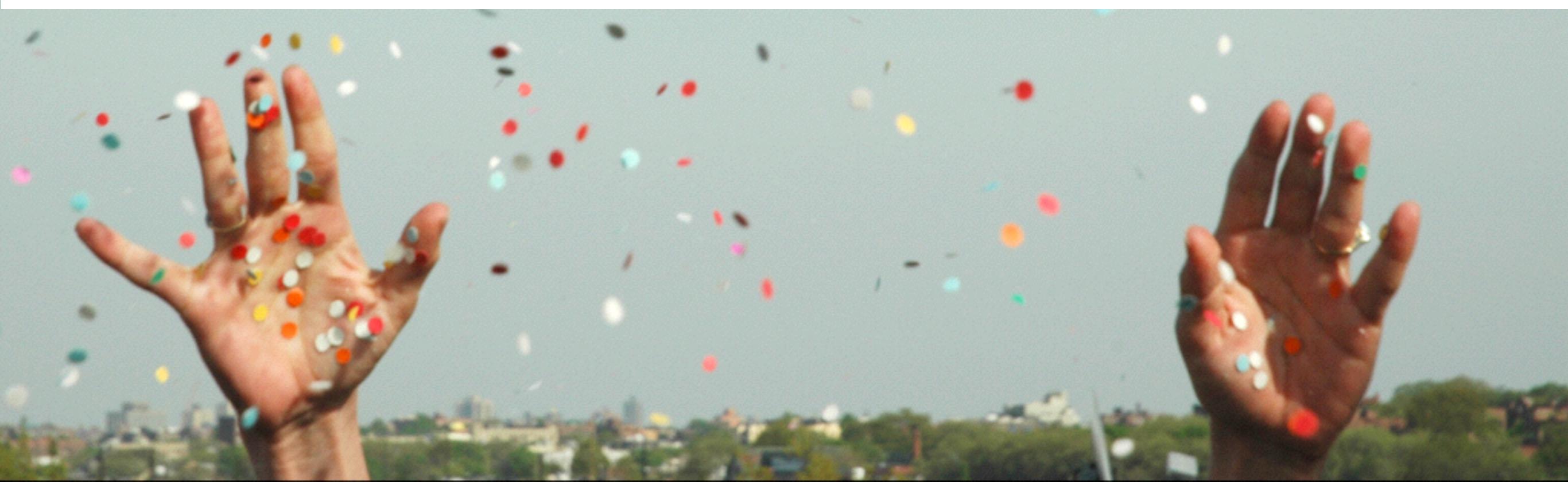




20 Cool Things You Can Do With Python,

Did someone say par-tay?



20. Write the Next Dropbox

Python is powerful enough to power Dropbox.



Your stuff, anywhere

[Sign up](#)

or [Sign in](#)

20. Write the Next Dropbox

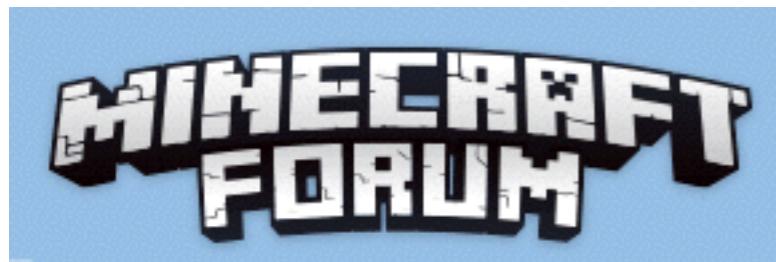


Python creator Guido van Rossum left Google to work for Dropbox in 2012.

19. Teach Kids to Code

You can use Python to teach your kids how to code.

That's how simple and well-designed Python is.



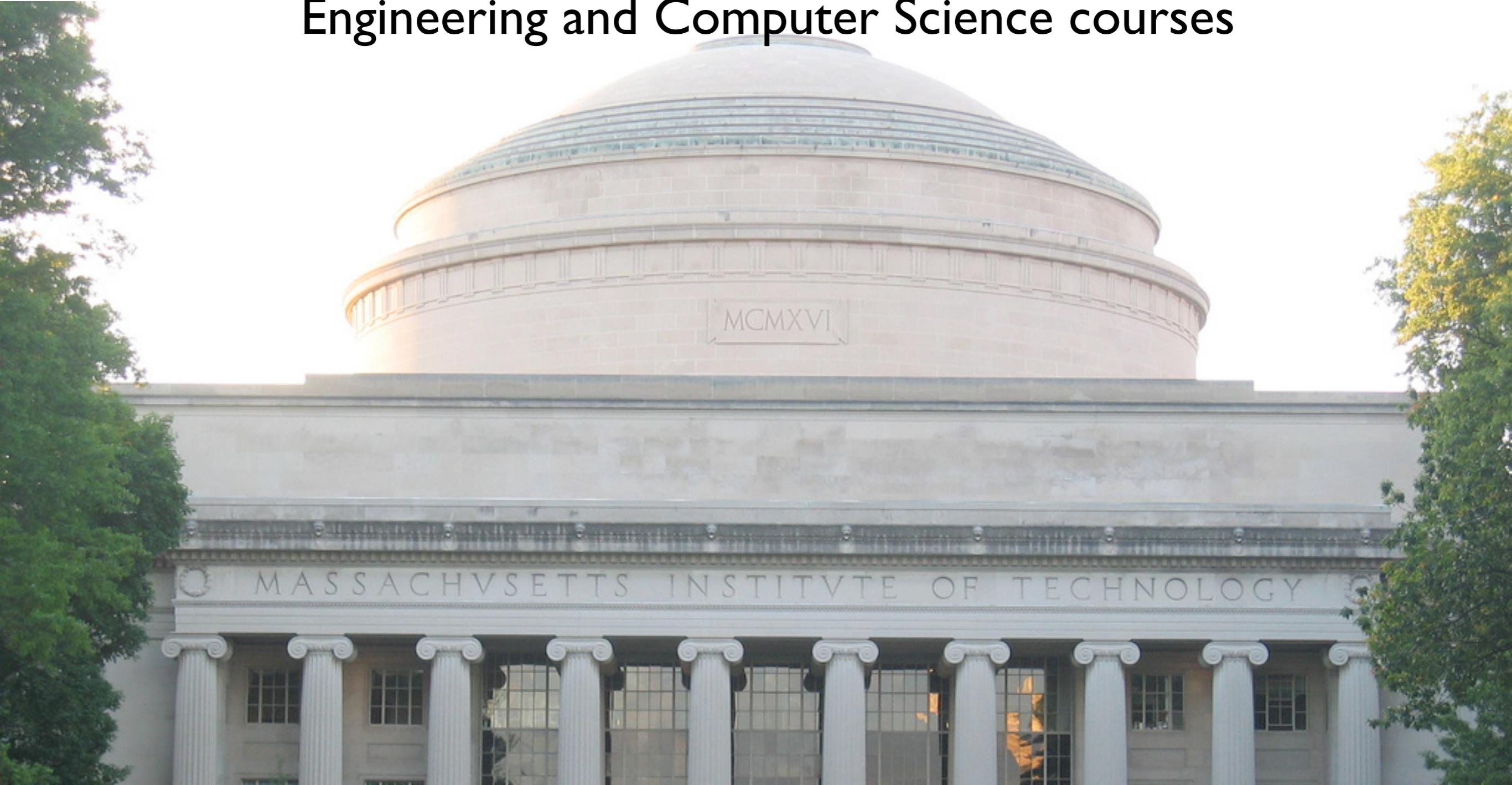
But what happens at a kid's day at PyconUK?

Glad you asked! Mums and sons, dads and daughters and developers and teachers come together for exciting adventures in code. One example (among many) from last year was Rebecca, Penelope and Emily who worked with three professional developers to create a flying-saucer game in Minecraft. They ended up presenting their work to the *whole conference* and got a huge round of applause! Here they are explaining what they got up to:



18. Teach MIT Students to Code

Python is used in MIT's Introduction to Electrical
Engineering and Computer Science courses



18. Teach MIT Students to Code

<https://www.flickr.com/photos/joiseyshowaa/1279750389>



6.01 Introduction to EECS 1

software engineering - feedback and
control - circuits - probability and
planning



17. Make or Lose Millions

You can write Python code to automatically buy and sell stocks with real money.

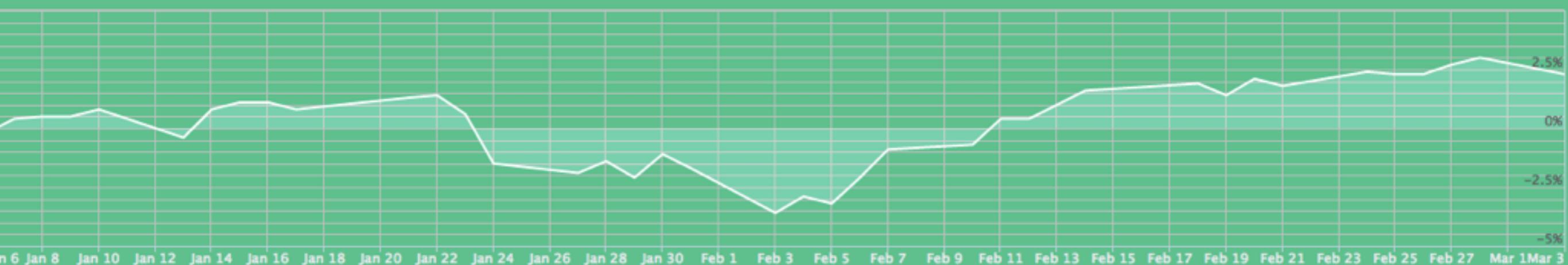
Your algorithms, trading live.

With Quantopian, you turn your investment goals into an algorithm that **trades stocks**. We help you develop your algorithm by providing **intelligent automation**, our **community** of investors, and professional-grade data. Then we turn your strategy into reality by **executing your algorithm** in the market.

[CODE A SAMPLE ALGORITHM](#)

[NO-CODE ALGORITHM BUILDER](#)

Cumulative performance since Jan 06, 2014



17. Make or Lose Millions

Tools like Quantopian let you take on Wall Street with just a few lines of Python code.

The screenshot shows the Quantopian platform interface. On the left, there is a code editor window titled "Sample Algorithm for Live Trading 2" containing Python code. On the right, there is a backtest summary and a line chart comparing the algorithm's performance against the SPY benchmark from April 1, 2014, to September 18, 2014.

Code Editor (Left):

```
1  """
2      This algorithm defines a long-only equal weight portfolio and
3      rebalances it at a user-specified frequency
4      NOTE: This algo is intended to run in minute-mode simulation and
5      is compatible with LIVE TRADING.
6
7  """
8
9  # Import the libraries we will use here
10 import datetime
11 import pytz
12 import pandas as pd
13
14 def initialize(context):
15     # This initialize function sets any data or variables
16     # that you'll use in your algorithm.
17     # You'll also want to define any parameters or values
18     # you're going to use.
19
20     # In our example, we're looking at 9 sector ETFs.
21     context.secs = symbols('XLY', # XLY Consumer Discretionary SPDR
22                           Fund
23                           'XLF', # XLF Financial SPDR Fund
24                           'XLK', # XLK Technology SPDR Fund
25                           'XLE', # XLE Energy SPDR Fund
26                           'XLV' # XLV Health Care SPDR Fund
```

Backtest Summary (Top Right):

Parameter	Value
Start Date	01/04/2014
End Date	09/18/2014
Initial Capital	\$ 50000
Frequency	Daily
RETURNS	0%
ALPHA	-0.03
BETA	0.00
SHARPE	--
DRAWDOWN	0%

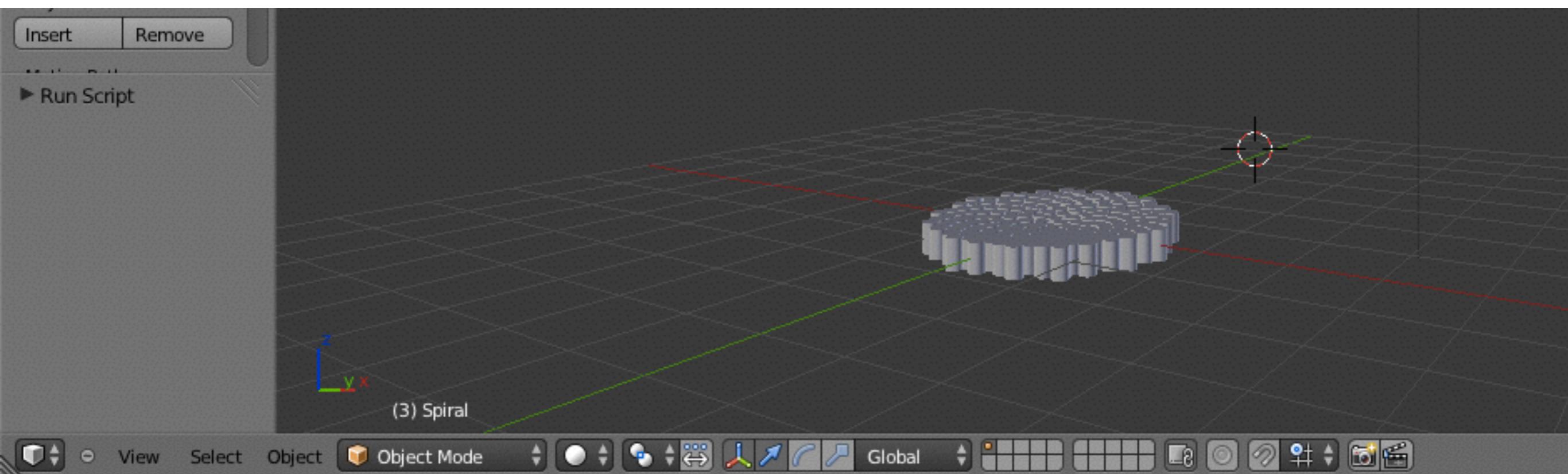
Line Chart (Bottom Right):

The chart displays the performance of the algorithm (blue line) and the SPY benchmark (red line) from February to September 2014. The Y-axis represents percentage return, ranging from -10% to 10%. The X-axis shows monthly intervals. The algorithm starts at 0% and remains flat until May, after which it begins to rise, reaching approximately 10% by September. The SPY benchmark starts at 0% and rises steadily throughout the period, reaching about 10% by September.

Logs (Bottom Right):

This backtest didn't generate any logs.

16. Make 3D Printed Jewelry



```
mesh = bpy.data.meshes.new(name="Spiral")

for i in range(0, n):
    theta = i * math.radians(137.5)
    r = c * math.sqrt(i)
    #bm.verts.new((math.cos(theta) * r, math.sin(theta) * r, 0.0))
    bpy.ops.mesh.primitive_cylinder_add(
        radius=0.1,
        location=[
            math.cos(theta) * r,
            math.sin(theta) * r,
            0.0])
```

15. Make 2D Art

I was a professional artist for a few years, with a studio practice in SF.

I created some of my artwork with Python code (PIL, PyCairo).



14. GIS

Shapely for GIS analysis.
Based on GEOS.

Fiona to read/write GIS data.

Powerful, free Python tools for
geospatial programming.

14. GIS

You can also write Python scripts for Esri's ArcGIS platform.

[Industries](#)[Products](#)[Support & Services](#)[About](#)[Community](#)[Sign In](#)

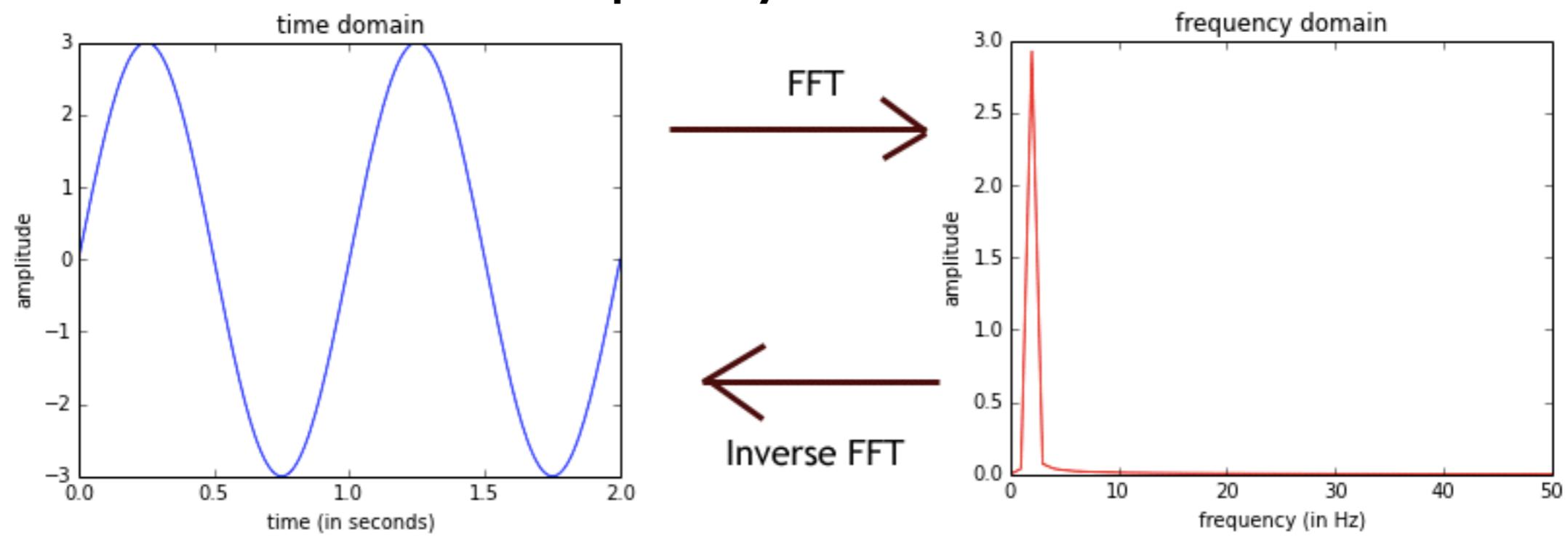
Esri Tapestry Segmentation

We're a nation of ZIP codes + neighborhoods. See how the next generation of Tapestry describes yours →



13. Sound Analysis/ Generation

Represent & analyze sound waves in time and frequency domains.

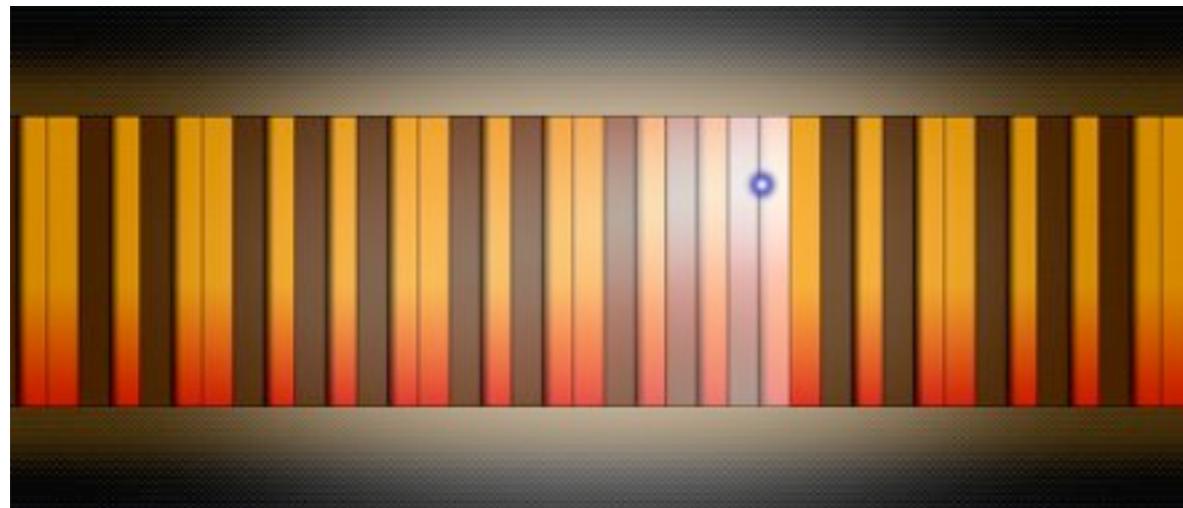


<https://github.com/calebmadrigal/FourierTalkOSCON>

And make funny noises.

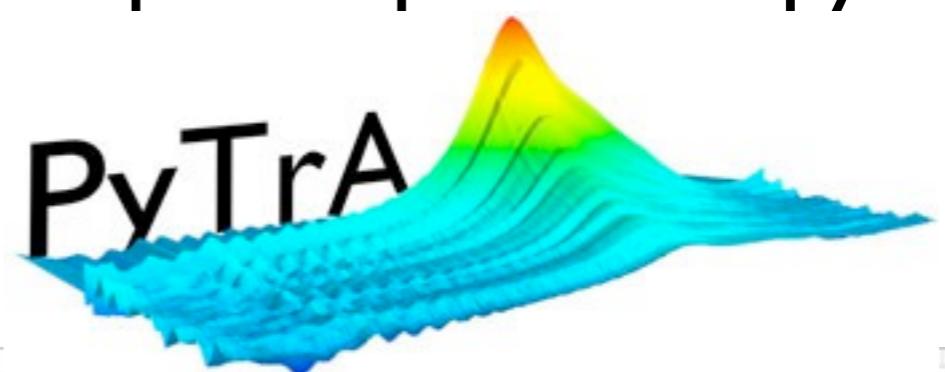
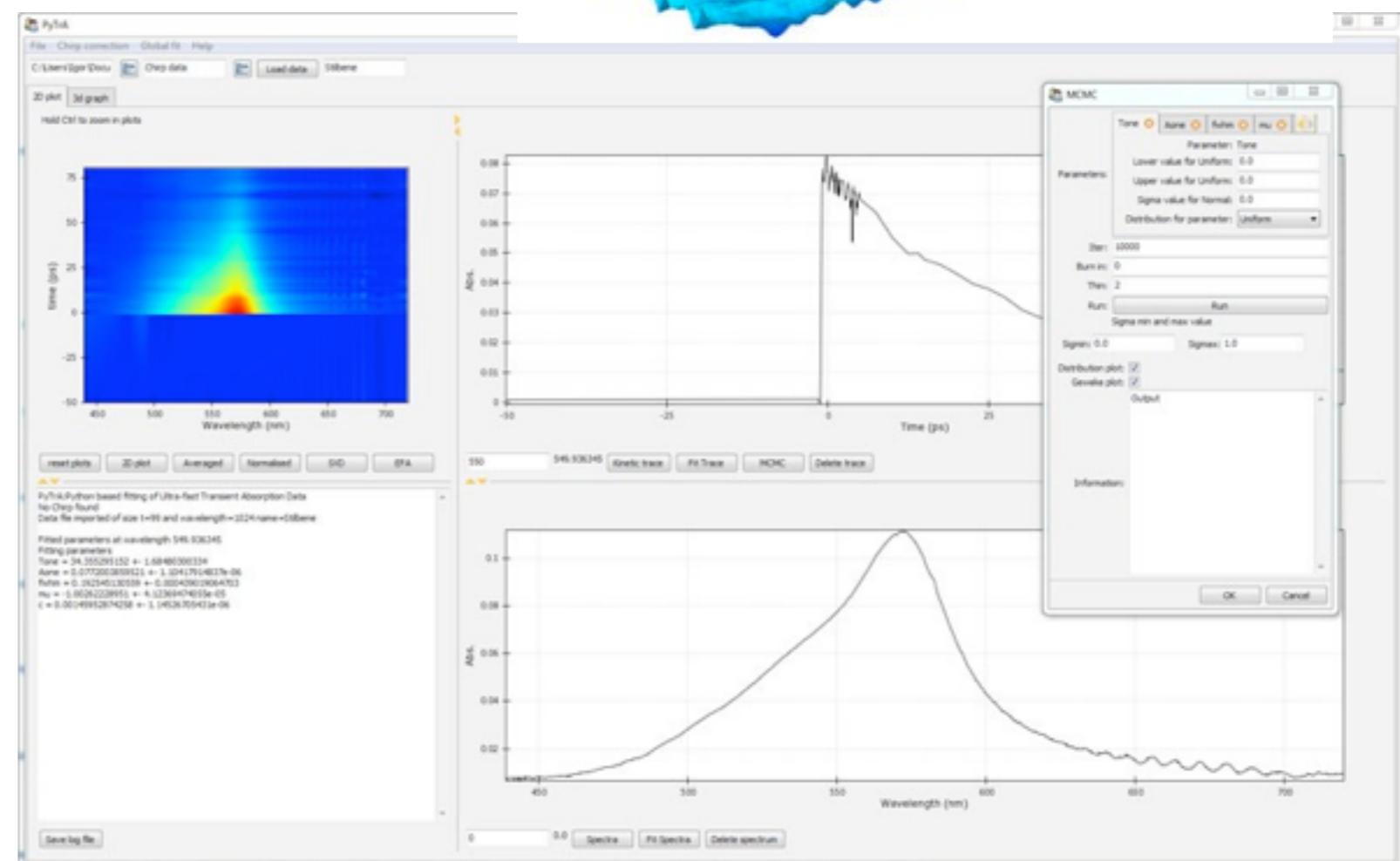
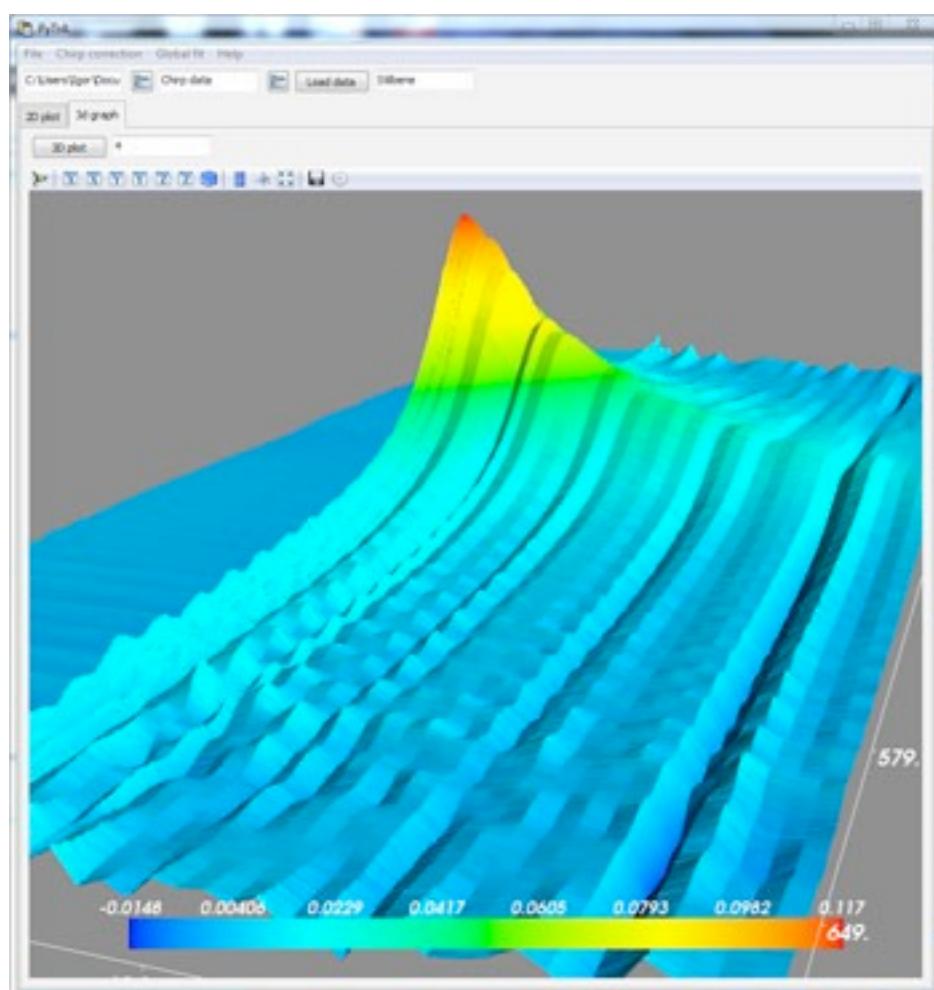
12. Mobile/ Touchscreen Apps

With Kivy, you can write apps in Python for mobile and touchscreen devices.



11. Scientific Data Analysis

Need a Python library for transient absorption spectroscopy?
Yep, there's one for that.



11. Scientific Data Analysis

PyTrA is a project from the Photon Factory laser facility of the University of Auckland, New Zealand.

Photon Factory

Python-Based Transient Absorption Spectroscopy Data Analysis

PyTrA combines many of the common fitting techniques used in ultrafast transient absorption spectroscopy in an easy to use package.

Transient absorption spectroscopy is a pump probe technique that provides details of how excited molecules' absorbance changes just after being excited. In the Photon Factory, a pump pulse 150 fs in duration can be tuned to different single wavelengths to excite the molecules into their excited state. Then a probe pulse that contains a full spectrum of colour is used to take a snapshot of the absorbance of the molecules at set time intervals after their excitation. The probe can then be delayed relative to the pump and the decay in the excited molecules can be observed.

