Technical Report

Brian Dant

Pellentesque tincidunt lobortis orci non venenatis. Cras in justo luctus, pulvinar augue id, suscipit diam. Morbi aliquet fringilla nibh, vel pellentesque dui venenatis eget. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec ultricies ultrices magna gravida porta.

# Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras egestas auctor molestie. In hac habitasse platea dictumst. Duis turpis tellus, scelerisque sit amet lectus ut, ultricies cursus enim. Integer fringilla a elit at fringilla. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla congue consequat consectetur.

# Heading

## Sub-heading

Pellentesque tincidunt *lobortis orci non venenatis*. **Cras in justo luctus, pulvinar augue id, suscipit diam**. Morbi aliquet fringilla nibh, vel pellentesque dui venenatis eget. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

* Donec ultricies ultrices magna gravida porta.
* Maecenas accumsan diam dui, auctor ornare ex pellentesque id.
* Integer tempus massa id augue finibus convallis.

Duis magna nunc, rutrum sodales elementum a, egestas eget dolor. Vivamus non pretium nunc, at vestibulum arcu. Vivamus pharetra, justo vel blandit feugiat, magna neque feugiat mi, a bibendum erat arcu a est.

Nulla vestibulum, nisl id tempor pulvinar, felis dui pellentesque lacus, quis bibendum metus enim sed ex.

Sed vestibulum, leo non elementum condimentum, dui ipsum vehicula quam, a vehicula turpis ipsum a nibh. Mauris posuere lobortis porta. Aenean augue neque, varius vitae [dapibus](http://authorea.com). (Gardner 1998)

## Code snippets: inline and block

To type code inline hit the <> button in the toolbar or hit shift-cmd-c. You can also include code blocks from the Insert menu. Syntax highlighting is automatic.

export function transformDomArray(domArray, options={}){  
 let mappings = {  
 'P': 'DIV',  
 'SPAN': null, // a comment here  
 'EM': 'I'  
 }  
}

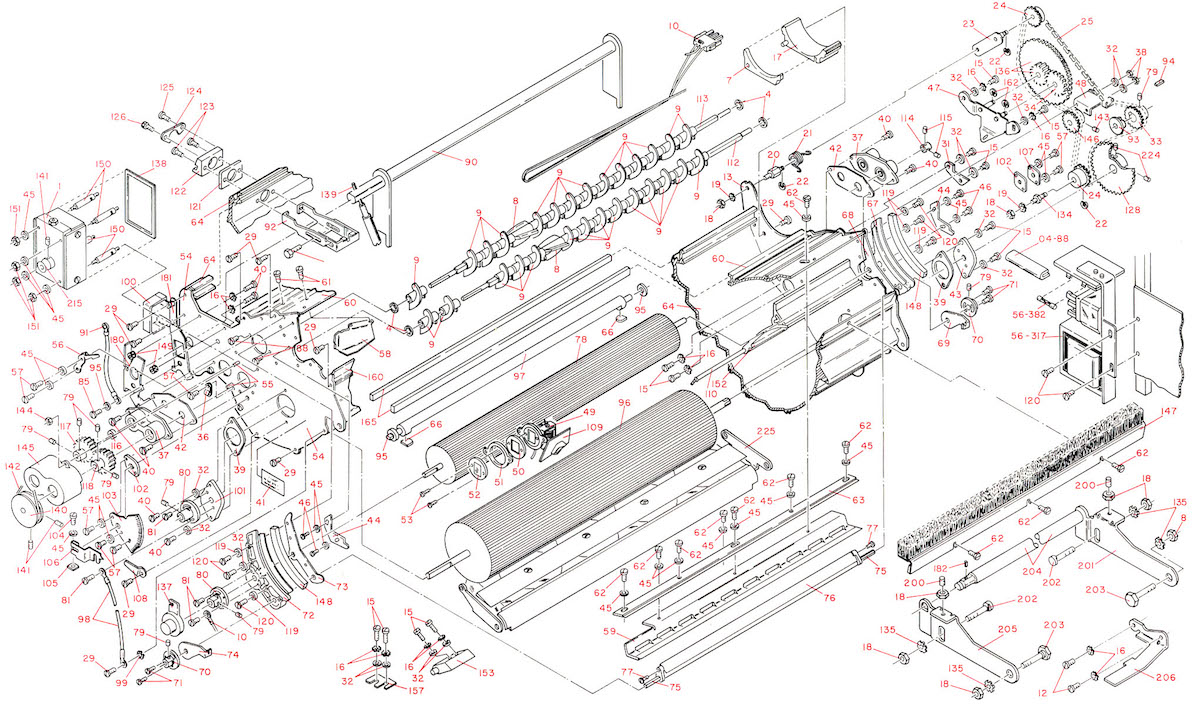
## Code blocks via LaTeX listings

### Python

class MyClass(Yourclass):  
 def \_\_init\_\_(self, my, yours):  
 bla = '5 1 2 3 4'  
 print bla

## Subheading

Etiam efficitur, arcu ut euismod sagittis, dui ex suscipit lectus, in efficitur sem sem et velit. Quisque pellentesque sapien eu ligula convallis, ut gravida massa vehicula. Ut mollis nulla nec dictum tempus. In dapibus accumsan nisl, vel facilisis arcu fermentum sit amet. Mauris sed consequat odio. Nulla auctor, elit in fermentum gravida, nulla neque mollis tortor, eu pretium nunc nisi sit amet lectus. Mauris suscipit felis vitae nunc malesuada rhoncus (Wilm and Murakami 2015). Morbi nibh nunc, egestas sed vehicula sed, ultricies vehicula magna Fig. ???.



Ut mollis nulla nec dictum tempus. In dapibus accumsan nisl, vel facilisis arcu fermentum sit amet. Mauris sed consequat odio. Nulla auctor, elit in fermentum gravida, nulla neque mollis tortor, eu pretium nunc nisi sit amet lectus.

# Interactive figures

An interactive figure built with d3.js ([this is how](https://intercom.help/authorea/host-data/interactive-figures/inserting-an-interactive-figure))



This interactive figure is built with d3.js ([instructions](https://intercom.help/authorea/host-data/interactive-figures/inserting-an-interactive-figure)). Click near the nodes to create new ones. Connect the nodes!

# References

Gardner, F.M. 1998. “HTML Sourcebook: A Complete Guide To HTML 3.2 And HTML Extensions [Book Reviews]”. *IEEE Communications Magazine* 36 (6). Institute of Electrical and Electronics Engineers (IEEE): 26–28. doi:10.1109/mcom.1998.685344.

Wilm, Johannes, and Shinyu Murakami. 2015. “Vivliostyle - Web Browser Based CSS Typesetting Engine”. In *XML London 2015 Conference Proceedings*. XML London. doi:10.14337/xmllondon15.wilm01.