

PrairieLearn Auto-Drawing

Senior Design May 2024 - 33

Mitch Hudson
Tyler Weberski
Chris Costa
Andrew Winters
Carter Murawski
Matt Graham

Table of Contents

Table of Contents	2
Introduction	3
Setting Up Drawing	3
Initial Drawing Element	3
Drawing Each Element	3
Line	3
Triangle	4
Circle	4
Rectangle	4
Text	5
Vector	5
Polygon	6
Connecting HTML and Python	6
HTML	6
Python	7
Resources	

Introduction

This document outlines how to use the PrairieLearn implementation of drawing, in order to auto-draw a diagram to randomize a question. This will include the multiple different elements which we currently have experience with when developing questions, and how we connect those pieces together to randomize with html and python.

Setting Up Drawings

Initial Drawing Element

The initial Drawing Element is set up within the drawing element you will make. This drawing initial has no parameters that you will set, it is simply stating that within this element, this drawing will be placed when you create the question, and cannot be changed or editable by a user.

Drawing Each Element

Line

Drawing a line within a drawing element starts with indicating the 'x1' and 'y1' attributes. These attributes indicate the position the line will start at, with x1 being the pixels from the left border of the drawing element, and y1 being the pixels from the top border of the drawing element. To determine the length and angle of the line, there are two approaches which you can take. The first approach is to change the 'width' and 'angle' attributes. The width is the length of the line, and the angle is the rotation around the starting 'x1' and 'y1' attributes, from the horizontal axis, going in the clockwise direction. The second approach you can take is specifically setting the attributes 'x2' and 'y2' which will determine the necessary angle and width for you, if you know the desired endpoint. The other attributes which are manually set to a value for you are the 'opacity', 'stroke-color', and 'stroke-width' attributes. The opacity attribute sets the opacity of the line (How transparent the line is). The stroke-color is what color you desire the line to be. PrairieLearn has a few set colors which can be found [here](#), or you can give a 6 digit hex value for the color. The stroke-width attribute will let you customize the width of the line. The last customizable option with the line element is the option to do a dashed line. The 'dashed-size' attribute allows you to input a float value which creates an equally spaced dashed line at the desired size set.

Example Code:

```
<pl-line x1="60" y1="80" x2="140" y2="120"></pl-line>
```

Triangle

Drawing a triangle in the drawing element starts with 6 attributes needed to be set. These attributes set the 3 vertices of the triangle with attributes 'x1', 'y1', 'x2', 'y2', 'x3', and 'y3'. Each x attribute is the amount of pixels from the left border the vertex will be at. Each y attribute is the amount of pixels from the top border the vertex will be at. The other attributes in which you can customize a triangle are 'opacity', 'color', 'stroke-color', and 'stroke-width'. The opacity attribute sets the opacity of the line (How transparent the line is). The stroke-color is what color you desire the line to be. PrairieLearn has a few set colors which can be found [here](#), or you can give a 6 digit hex value for the color. The stroke-width attribute will let you customize the width of the line.

Example Code:

```
<pl-triangle x1="40" y1="180" x2="160" y2="180" x3="160" y3="40"></pl-triangle>
```

Circle

Drawing a circle in the drawing element starts with 2 attributes, 'x1' and 'y1'. These attributes set the point that is the center of the circle, with x1 being the distance from the left border, and y1 being the distance from the top border of the drawing element. In order to set the radius of the circle, you use the attribute 'radius'. The other attributes in which you can customize a triangle are 'opacity', 'color', 'stroke-color', and 'stroke-width'. The opacity attribute sets the opacity of the line (How transparent the line is). The stroke-color is what color you desire the line to be. PrairieLearn has a few set colors which can be found [here](#), or you can give a 6 digit hex value for the color. The stroke-width attribute will let you customize the width of the line. In order to label a circle, there are 3 different attributes you can use: 'label', 'offsetx', and 'offsety'. The label attribute takes a string which is the text that labels the center of the circle by default. If you want to change the position of the label, then the offsetx and offsety attributes change the position of the label, based on the center of the circle.

Example Code:

```
<pl-circle x1="40" y1="100" radius="30" color="#95A5A6" stroke-width="3"></pl-circle>  
<pl-circle x1="120" y1="100" radius="50" color="#D0D3D4"></pl-circle>
```

Rectangle

Drawing a rectangle in the drawing element starts with the 'x1' and 'y1' attributes. These attributes identify the center of the rectangle, with x1 being the distance from the left border, and y1 being the distance from the top border of the drawing element. In order to adjust the height and width of the rectangle, you change the attributes 'height' and 'width'. The height attribute affects how tall the rectangle is, and the width is how wide the rectangle is. If you want to have the rectangle at an angle, the 'angle' attribute,

rotates the rectangle around the center point identified by 'x1' and 'y1'. The angles are measured from the horizontal axis, and travel clockwise. The opacity attribute sets the opacity of the line (How transparent the line is). The stroke-color is what color you desire the line to be. PrairieLearn has a few set colors which can be found [here](#), or you can give a 6 digit hex value for the color. The stroke-width attribute will let you customize the width of the line.

Example Code:

```
<pl-rectangle x1="80" y1="50" width="100" height="40"></pl-rectangle>  
<pl-rectangle x1="120" y1="140" width="120" height="20" angle="30" color="pink1"></pl-rectangle>
```

Text

The text in the drawing element starts with 3 attributes. These variables are the 'x1', 'y1', and 'label' attributes. X1 and y1 both identify where the text will be displaced, with these values being the top left corner of the text. X1 is the distance from the left border, and y1 being the distance from the top border of the drawing element. If you want to use latex text, you can use the attribute 'latex'. This is on by default, and can be set to false to turn it off. You can also change the font size, with the attribute 'font-size'. The last customizable option is the offset you can adjust, with the attributes 'offsetx' and 'offsety', which offset in the x(to the right of the drawing element) or y (going down of the drawing element) direction.

Example Code:

```
<pl-text x1="20" y1="40" label="Moment of inertia:" latex="false"></pl-text>  
<pl-text x1="20" y1="80" label="\int_A y^2 dA"></pl-text>
```

Vector

The vector in the drawing element has lots of customizable properties, but it starts with the 'x1' and 'y1' attributes. These attributes identify the anchor point for the vector, with x1 being the distance from the left border, and y1 being the distance from the top border of the drawing element. Some other familiar attributes include 'width', 'angle', 'label', 'offsetx', 'offsety', 'color', and 'stroke-width'. Width determines the length of the vector. Angle determines what angle the vector is at, rotating around the x1 and y1 attribute values, measured from the horizontal axis going clockwise. The color attribute is what color you desire the line to be. PrairieLearn has a few set colors which can be found [here](#), or you can give a 6 digit hex value for the color. The stroke-width attribute will let you customize the width of the line. Label, offsetx, and offsety attributes all go together, with label being the text to label a vector. Offsetx and offsety, determine how far from the head of the vector the label is away. The unique features to the vector include 'arrow-head-width' and 'arrow-head-length'. These attributes let you customize the head

of the arrow which the vector is, with the width being how wide the arrowhead is, and the length is how long the arrowhead is.

Example Code:

```
<pl-vector x1="100" y1="100" width="80" angle="45" label="F"></pl-vector>
<pl-vector
  x1="100"
  y1="100"
  width="80"
  angle="45"
  color="green3"
  anchor-is-tail="false"
></pl-vector>
```

Polygon

The polygon element is very unique within the drawing element. To start with this element, you start with the 'plist' attribute. This allows you to list the vertices of the polygon you want to form. It also draws the lines from one point to the next, listed in the plist attribute. The format for this is shown in the example code, but follows the format `plist = [{"x": x1, "y": y1}, {"x": x2, "y": y2}, ... , {"x": xn, "y": yn}]`. Other than this, the rest of the attributes are pretty familiar being 'opacity', 'color', 'stroke-color', and 'stroke-width'. The opacity attribute sets the opacity of the line (How transparent the line is). The color is what color you want the polygon to be filled with. The stroke-color is what color you desire the outline to be. PrairieLearn has a few set colors which can be found [here](#), or you can give a 6 digit hex value for the color. The stroke-width attribute will let you customize the width of the outline.

Example Code:

```
<pl-polygon plist='[{"x": 40, "y": 40}, {"x": 140, "y": 80}, {"x": 60, "y": 180}]'></pl-polygon>
```

Connecting HTML and Python

HTML

The HTML side of the drawing element is where you make the frame, as well as drawing the individual elements to put in the frame. This can be hardcoded, to which python is not needed. If you want to randomize the drawing everytime a individual tries to access a question, then python is needed, and will be explained in the python section. Having a drawing, and drawing-group allows you to place drawings as desired, and account for overlapping features on one drawing.

Python

With the python side of drawing, if you want to customize the variability of a question, by making it random for every student who opens your question, then python scripts are needed. By setting an attribute such as 'x1' equal to `{{params.YOUR_PARAMATER_NAME}}` made in the python script, you are able to customize what values are chosen to be set for that drawing. Below is code from the git drawing tutorial (link [here](#)). Where the variables are assigned in python, then set in the html file.

Example Code (HTML):

```
<pl-drawing grid-size="20" gradable="true" answers-name="point" hide-answer-panel="false" height={{params.canvas_height}} width={{params.canvas_width}}>

  <pl-drawing-answer draw-error-box="true">

    <pl-drawing-group visible={{params.visible1}}>
      <pl-point x1={{params.xP}} y1={{params.yP}} > </pl-point>
    </pl-drawing-group>

    <pl-drawing-group visible={{params.visible2}}>
      <pl-vector x1={{params.xB}} y1={{params.y}} angle={{params.alpha}} anchor-is-tail="true" > </pl-vector>
    </pl-drawing-group>

    <pl-drawing-group visible={{params.visible3}}>
      <pl-arc-vector x1={{params.xC}} y1={{params.y}} clockwise-direction={{params.arc_orientation}} anchor-is-tail="true" > </pl-arc-vector>
    </pl-drawing-group>

    <pl-drawing-group visible={{params.visible4}}>
      <pl-distributed-load x1={{params.xE}} y1={{params.y}} width={{params.w}} angle={{params.theta}} anchor-is-tail="true" > </pl-distributed-load>
    </pl-drawing-group>

    <pl-drawing-group visible={{params.visible5}}>
      <pl-distributed-load x1={{params.xE}} y1={{params.y}} w1={{params.w1}} w2={{params.w2}} width={{params.w}} angle={{params.theta}} anchor-is-tail="true" > </pl-distributed-load>
    </pl-drawing-group>

    <pl-drawing-group visible={{params.visible6}}>
      <pl-double-headed-vector x1={{params.xC}} y1={{params.y}} angle={{params.gamma}} > </pl-double-headed-vector>
    </pl-drawing-group>

  </pl-drawing-answer>

  <pl-drawing-initial >
    <pl-drawing-group visible={{params.visible1}}>
      <pl-coordinates x1={{params.x0}} y1={{params.y0}} label="0" offsetx="-10" offsety="5"> </pl-coordinates>
    </pl-drawing-group>

    <pl-drawing-group visible={{params.visible0}}>
      <pl-rod x1={{params.xA}} y1={{params.y}} x2={{params.xD}} y2={{params.y}}
        label1="A" offsetx1="-10" offsety1="10"
        label2="D" offsetx2="-10" offsety2="10" > </pl-rod>
      <pl-point x1={{params.xB}} y1={{params.y}} label="B" offsetx="-10" offsety="10"></pl-point>
      <pl-point x1={{params.xC}} y1={{params.y}} label="C" offsetx="-10" offsety="10"></pl-point>
      <pl-coordinates x1={{params.x0}} y1={{params.y0}} label="0" offsetx="-10" offsety="5"> </pl-coordinates>
    </pl-drawing-group>
  </pl-drawing-initial>
```

Example Code (Python):

```
canvas_width = 300
canvas_height = 300
data["params"]["canvas_width"] = canvas_width
data["params"]["canvas_height"] = canvas_height

# Creating all variables, even knowing that just a few would be used in this Example
# Could improve later
# -----
# point
# -----
x0 = 40
y0 = 40
interval = np.linspace(100, 200, 11)
xP = random.choice(interval)
data["params"]["xP"] = xP
yP = random.choice(interval)
data["params"]["yP"] = yP
data["params"]["x0"] = x0
data["params"]["y0"] = canvas_height - y0
```

Resources

PrairieLearn Resources

Full Link for Drawing: <https://prairielearn.readthedocs.io/en/latest/pl-drawing/>

Initial Element:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-drawing-initial-element>

Drawing Lines:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-line-element>

Drawing Triangles:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-triangle-element>

Drawing Circles:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-circle-element>

Drawing Rectangles:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-rectangle-element>

Drawing Text:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-text-element>

Drawing Vectors:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-vector-element>

Drawing Polygon:

<https://prairielearn.readthedocs.io/en/latest/pl-drawing/#pl-polygon-element>

PrairieLearn Colors:

<https://prairielearn.readthedocs.io/en/latest/course/#colors>

PrairieLearn Drawing Tutorial Git:

<https://github.com/PrairieLearn/PrairieLearn/tree/master/exampleCourse/questions/demo/drawing>