venndiagram v1.0: Drawing Simple Venn Diagrams

Nicola L. C. Talbot

http://www.dickimaw-books.com/

2012-10-24

The venndiagram package is provided to assist generating simple two- and three-set Venn diagrams for lectures or assignment sheets. This package requires the tikz package.

Contents

1 Available Commands and Environments

This package defines two environments:

venndiagram3sets

```
\begin{venndiagram3sets}[⟨options⟩]
```

and

venndiagram2sets

```
\begin{venndiagram2sets}[⟨options⟩]
```

The optional argument ⟨*options*⟩ is a comma-separated list of key=value settings.

1

> If the ⟨*value*⟩ contains commas or equal signs, make sure you enclose the entire value in braces. For example:
>
> ```
> \begin{venndiagram3sets}[tikzoptions={scale=2,thick}]
> ```

The following keys are available:

shade The name of the colour used to shade regions (default: `lightgray`).

labelA The label for the first set (default: `$A$`).

labelB The label for the second set (default: `$B$`).

labelC (Not available for the 2 set version.) The label for the third set (default: `$C$`).

labelOnlyA The label for the region given by $A \setminus (B \cup C)$ (for 3 set version) or $A \setminus B$ (for 2 set version). (Default: empty.)

labelOnlyB The label for the region given by $B \setminus (A \cup C)$ (for 3 set version) or $B \setminus A$ (for 2 set version). (Default: empty.)

labelOnlyC (Not available for 2 set version.) The label for the region given by $C \setminus (A \cup B)$. (Default: empty.)

labelOnlyAB (Not available for 2 set version.) The label for the region given by $(A \cap B) \setminus C$. (Default: empty.)

labelOnlyAC (Not available for 2 set version.) The label for the region given by $(A \cap C) \setminus B$. (Default: empty.)

labelOnlyBC (Not available for 2 set version.) The label for the region given by $(B \cap C) \setminus A$. (Default: empty.)

labelABC (Not available for 2 set version.) The label for the region given by $A \cap B \cap C$. (Default: empty.)

labelNotABC (Not available for 2 set version.) The label for the region given by $(A \cup B \cup C)^c$. (Default: empty.)

labelAB (Not available for 3 set version.) The label for the region given by $A \cap B$. (Default: empty.)

labelNotAB (Not available for 3 set version.) The label for the region given by $(A \cup B)^c$. (Default: empty.)

radius The radius of each set. (Default: 1.2cm.)

hgap The horizontal gap between the outer vertical edge and the nearest set edge. (Default: 0.5cm.)

vgap The vertical gap between the outer horizontal edge and the nearest set edge. (Default: 0.5cm.)

overlap The overlap between the sets. (Default: 0.75cm.)

tikzoptions Any options to pass to tikzpicture.

Both environments draw the outline of the sets and the rectangular outline of the encompassing universal set. Within the Venn diagram environments commands are provided to shade various regions. (The commands have a cumulative effect, possibly drawing over each other. The set outlines and labels are drawn at the end of the environment.) Available commands are as follows:

\fillA
```
\fillA
```

Shades set A.

\fillB
```
\fillB
```

Shades set B.

\fillC
```
\fillC
```

(Only for 3 set version.) Shades set C.

\fillAll
```
\fillAll
```

Shades the entire Venn diagram.

\fillNotABC
```
\fillNotABC
```

(Not available for 2 sets version.) Fills $(A \cup B \cup C)^c$.

\fillOnlyA
```
\fillOnlyA
```

Shades set $A \setminus (B \cup C)$ (for 3 sets version) or $A \setminus B$ (for 2 sets version).

\fillOnlyB
```
\fillOnlyB
```

Shades set $B \setminus (A \cup C)$ (for 3 sets version) or $B \setminus A$ (for 2 sets version).

\fillOnlyC
```
\fillOnlyC
```

(Not available for 2 sets version.) Shades $C \setminus (A \cup B)$.

\fillNotA
```
\fillNotA
```

Shades everything except A (that is A^c).

\fillNotB

```
\fillNotB
```

Shades everything except B (that is B^c).

\fillNotC

```
\fillNotC
```

(Not available for 2 set version.) Shades everything except C (that is C^c).

\fillNotAorB

```
\fillNotAorB
```

(Not available for 3 set version.) Shades $(A \cup B)^c$

\fillNotAorNotB

```
\fillNotAorNotB
```

(Not available for 3 set version.) Shades $(A \cap B)^c$

\fillANotB

```
\fillANotB
```

Shades $A \setminus B$.

\fillBNotA

```
\fillBNotA
```

Shades $B \setminus A$.

\fillANotC

```
\fillANotC
```

(Not available for 2 set version.) Shades $A \setminus C$.

\fillCNotA

```
\fillCNotA
```

(Not available for 2 set version.) Shades $C \setminus A$.

\fillBNotC

```
\fillBNotC
```

(Not available for 2 set version.) Shades $B \setminus C$.

\fillCNotB

```
\fillCNotB
```

(Not available for 2 set version.) Shades $C \setminus B$.

\fillACapB

```
\fillACapB
```

Shades $A \cap B$. (\fillBCapA is equivalent to \fillACapB.)

\fillACapC

```
\fillACapC
```

(Not available for 2 set version.) Shades $A \cap C$. (\fillCCapA is equivalent to \fillACapC.)

\fillBCapC   `\fillBCapC`

(Not available for 2 set version.) Shades $B \cap C$. (\fillCCapB is equivalent to \fillBCapC.)

\fillACapBNotC   `\fillACapBNotC`

(Not available for 2 set version.) Shades $A \cap B \setminus C$. (\fillBCapANotC is equivalent to \fillACapBNotC.)

\fillACapCNotB   `\fillACapCNotB`

(Not available for 2 set version.) Shades $A \cap C \setminus B$. (\fillCCapANotB is equivalent to \fillACapCNotB.)

\fillBCapCNotA   `\fillBCapCNotA`

(Not available for 2 set version.) Shades $B \cap C \setminus A$. (\fillCCapBNotA is equivalent to \fillBCapCNotA.)

\fillACapBCapC   `\fillACapBCapC`

(Not available for 2 set version.) Shades $A \cap B \cap C$. (Synonyms: \fillACapCCapB, \fillBCapACapC, \fillBCapCCapA, \fillCCapACapB, \fillCCapBCapA.)

\setpostvennhook   `\setpostvennhook{⟨cmds⟩}`

Sets the hook applied at the very end of the Venn diagram environments (after the outline and labels are drawn but before the end of the tikzpicture environment). The Venn diagram environments create coordinate nodes venn bottom left, venn top left, venn top right and venn bottom right, which may be referenced within the environment or in the hook.

The set labels may also be referenced *but only in \setpostvennhook*: labelOnlyA, labelOnlyB, labelOnlyC (three set version only), labelNotABC (three set version only), labelNotAB (two set version only), labelA, labelB, labelC (three set version only), labelOnlyAB, labelOnlyAC (three set version only), labelOnlyBC (three set version only) and labelAB (two set version only).

2 Examples

1. (Three sets) $A \cup B \cup C$

   ```
   \begin{venndiagram3sets}
    \fillA \fillB \fillC
   \end{venndiagram3sets}
   ```
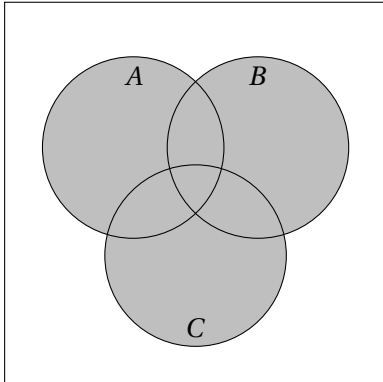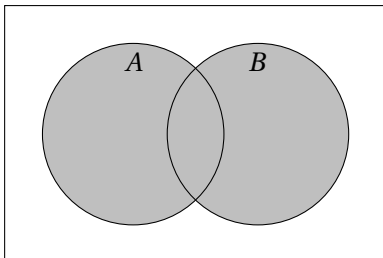
   Produces:

2. (Two sets) $A \cup B$

   ```
   \begin{venndiagram2sets}
    \fillA \fillB
   \end{venndiagram2sets}
   ```

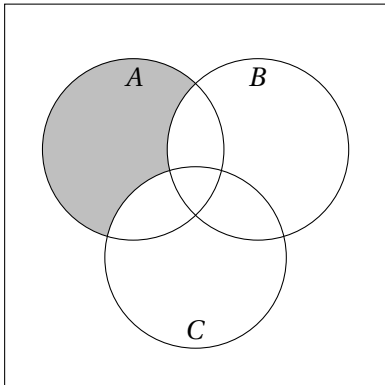   Produces:

3. (Three sets) $A \setminus (B \cup C)$

   ```
   \begin{venndiagram3sets}
    \fillOnlyA
   \end{venndiagram3sets}
   ```
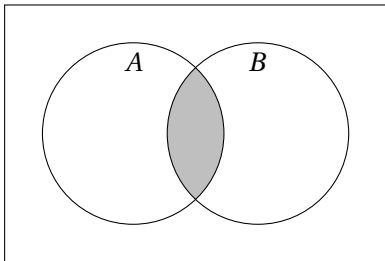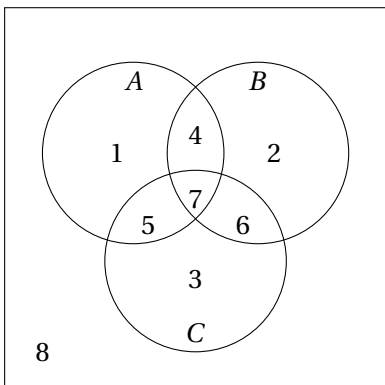
   Produces:

4. (Two sets) $A \cap B$:

```
\begin{venndiagram2sets}
\fillACapB
\end{venndiagram2sets}
```


5. (Three sets) region labels:

```
\begin{venndiagram3sets}[labelOnlyA={1},labelOnlyB={2},labelOnlyC={3},
 labelOnlyAB={4},labelOnlyAC={5},labelOnlyBC={6},labelABC={7},
 labelNotABC={8}]
\end{venndiagram3sets}
```

Produces:

6. Annotating the diagram:

```
\begin{venndiagram3sets}[labelOnlyA={1},labelOnlyB={2},labelOnlyC={3},
 labelOnlyAB={4},labelOnlyAC={5},labelOnlyBC={6},labelABC={7},
 labelNotABC={8}]
\setpostvennhook
{
  \draw[<-] (labelA) -- ++(135:3cm) node[above] {Students who eat
artichokes};
  \draw[<-] (labelB) -- ++(45:3cm) node[above] {Students who eat
broccoli};
  \draw[<-] (labelC) -- ++(-90:3cm) node[below] {Students who eat
carrots};
  \draw[<-] (labelABC) -- ++(0:3cm)
    node[right,text width=4cm,align=flush left]
   {7 students eat artichokes, broccoli and carrots};
  \draw[<-] (labelNotABC) -- ++(-135:3cm)
    node[below,text width=4cm,align=flush left]
   {8 students don't eat artichokes, broccoli or carrots};
}
\end{venndiagram3sets}
```

Produces:
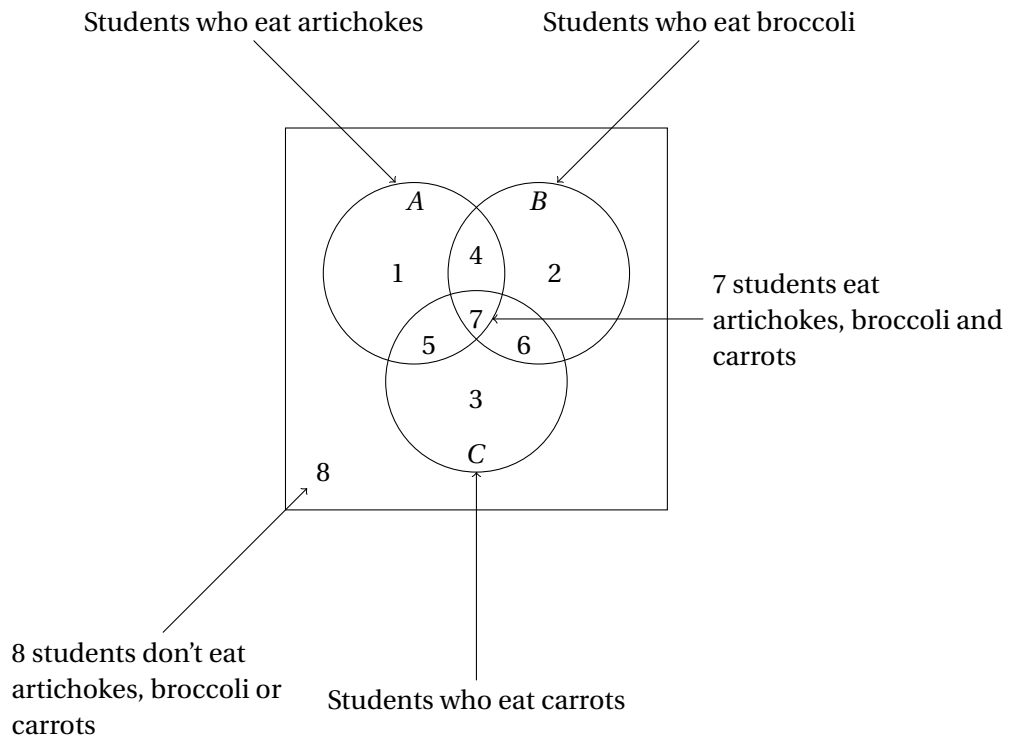
3 The Code

Package identification:

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{venndiagram}[2012/10/24 v1.0 (NLCT) Venn diagrams]
```

Required packages:

```
3 \RequirePackage{xkeyval}
4 \RequirePackage{tikz}
5 \RequirePackage{etoolbox}
```

TiKZ intersections library needed:

```
6 \usetikzlibrary{intersections}
```

3.1 Initialising the Default Values

Set up macros used by the keys for the Venn diagram options. First the default set labels.

\@venn@label@A   Set A:

```
7 \newcommand*{\@venn@label@A}{$A$}
```

\@venn@label@B   Set B:

```
8 \newcommand*{\@venn@label@B}{$B$}
```

\@venn@label@C   Set C:

```
9 \newcommand*{\@venn@label@C}{$C$}
```

\@venn@shade   The colour used to shade regions.

```
10 \newcommand*{\@venn@shade}{lightgray}
```

The default labels for all the other regions are empty.

\@venn@label@OnlyA   Only set A:

```
11 \newcommand*{\@venn@label@OnlyA}{}
```

\@venn@label@OnlyB   Only set B:

```
12 \newcommand*{\@venn@label@OnlyB}{}
```

\@venn@label@OnlyC   Only set C:

```
13 \newcommand*{\@venn@label@OnlyC}{}
```

\@venn@label@OnlyAB   Sets A and B but not C:

```
14 \newcommand*{\@venn@label@OnlyAB}{}
```

\@venn@label@OnlyAC   Sets A and C but not B:

```
15 \newcommand*{\@venn@label@OnlyAC}{}
```

`\@venn@label@OnlyBC`	Sets *B* and *C* but not *A*:
	`16 \newcommand*{\@venn@label@OnlyBC}{}`
`\@venn@label@ABC`	Intersection of sets *A*, *B* and *C*:
	`17 \newcommand*{\@venn@label@ABC}{}`
`\@venn@label@NotABC`	Everything except *A*, *B* or *C*:
	`18 \newcommand*{\@venn@label@NotABC}{}`
`\@venn@label@NotAB`	Everything except *A* or *B* (two set version only):
	`19 \newcommand*{\@venn@label@NotAB}{}`
`\@venn@label@AB`	Intersection of *A* and *B* (two set version only):
	`20 \newcommand*{\@venn@label@AB}{}`

Now the default dimensions of the diagrams.

`\@venn@radius`	The radius of the sets.
	`21 \newcommand*{\@venn@radius}{1.2cm}`
`\@venn@hgap`	The horizontal distance between the edge of the diagram and the outer edge of the nearest set.
	`22 \newcommand*{\@venn@hgap}{0.5cm}`
`\@venn@vgap`	
	`23 % The vertical distance between the edge of the diagram and the`
	`24 % outer edge of the nearest set.`
	`25 \newcommand*{\@venn@vgap}{0.5cm}`
`\@venn@overlap`	The size of the set overlap.
	`26 \newcommand*{\@venn@overlap}{0.75cm}`
`\@venn@tikzoptions`	Any options to be passed to the tikzpicture environment.
	`27 \newcommand*{\@venn@tikzoptions}{}`

Lengths to store the centres of the sets and the overall width and height of the diagram.

`\@venn@Ax`	The *x*-coordinate of set *A*:
	`28 \newlength\@venn@Ax`
`\@venn@Ay`	The *y*-coordinate of set *A*:
	`29 \newlength\@venn@Ay`
`\@venn@Bx`	The *x*-coordinate of set *B*:
	`30 \newlength\@venn@Bx`

\@venn@By   The *y*-coordinate of set *B*:

```
31 \newlength\@venn@By
```

\@venn@Cx   The *x*-coordinate of set *C*:

```
32 \newlength\@venn@Cx
```

\@venn@Cy   The *y*-coordinate of set *C*:

```
33 \newlength\@venn@Cy
```

\@venn@w   The width of the entire Venn diagram.

```
34 \newlength\@venn@w
```

\@venn@h   The height of the entire Venn diagram.

```
35 \newlength\@venn@h
```

3.2 Defining the key=value Options

Now define the keys for the optional argument of venndiagram2sets and venndiagram3sets. They are all in the family venn.

shade   Option to set the shading.

```
36 \define@key{venn}{shade}{\def\@venn@shade{#1}}
```

labelA   Option to set the label for set *A*.

```
37 \define@key{venn}{labelA}{\def\@venn@label@A{#1}}
```

labelB   Option to set the label for set *B*.

```
38 \define@key{venn}{labelB}{\def\@venn@label@B{#1}}
```

labelC   Option to set the label for set *C*.

```
39 \define@key{venn}{labelC}{\def\@venn@label@C{#1}}
```

Now for the region labels.

labelOnlyA   Option to set the label for only set *A*.

```
40 \define@key{venn}{labelOnlyA}{\def\@venn@label@OnlyA{#1}}
```

labelOnlyB   Option to set the label for only set *B*.

```
41 \define@key{venn}{labelOnlyB}{\def\@venn@label@OnlyB{#1}}
```

labelOnlyC   Option to set the label for only set *C*.

```
42 \define@key{venn}{labelOnlyC}{\def\@venn@label@OnlyC{#1}}
```

labelOnlyAB   Option to set the label for the intersection of *A* and *B*.

```
43 \define@key{venn}{labelOnlyAB}{\def\@venn@label@OnlyAB{#1}}
```

labelOnlyAC Option to set the label for the intersection of *A* and *C*.

44 `\define@key{venn}{labelOnlyAC}{\def\@venn@label@OnlyAC{#1}}`

labelOnlyBC Option to set the label for the intersection of *B* and *C*.

45 `\define@key{venn}{labelOnlyBC}{\def\@venn@label@OnlyBC{#1}}`

labelABC Option to set the label for the intersection of *A*, *B* and *C*. (Three set version only)

46 `\define@key{venn}{labelABC}{\def\@venn@label@ABC{#1}}`

labelNotABC Option to set the label for the region outside the three sets. (Three set version only)

47 `\define@key{venn}{labelNotABC}{\def\@venn@label@NotABC{#1}}`

labelAB Option to set the label for the intersection of *A* and *B*. (Two set version only)

48 `\define@key{venn}{labelAB}{\def\@venn@label@AB{#1}}`

labelNotAB Option to set the label for the region outside the two sets. (Two set version only)

49 `\define@key{venn}{labelNotAB}{\def\@venn@label@NotAB{#1}}`

Now for the dimension options.

radius Option to set the radius.

50 `\define@key{venn}{radius}{\def\@venn@radius{#1}}`

hgap Option to set the horizontal gap between the outer edge of the diagram and the nearest set edge.

51 `\define@key{venn}{hgap}{\def\@venn@hgap{#1}}`

vgap Option to set the vertical gap between the outer edge of the diagram and the nearest set edge.

52 `\define@key{venn}{vgap}{\def\@venn@vgap{#1}}`

overlap Option to set the set overlap.

53 `\define@key{venn}{overlap}{\def\@venn@overlap{#1}}`

Finally the option to set the information to pass to the tikzpicture environment.

tikzoptions

54 `\define@key{venn}{tikzoptions}{\def\@venn@tikzoptions{#1}}`

3.3 Environment Definitions

venndiagram3sets Environment to draw Venn diagram with three sets.

```
55 \newenvironment{venndiagram3sets}[1][]%
56 {%
```

Disable the keys that aren't applicable.

```
57   \disable@keys{venn}{labelAB,labelNotAB}%
```

Set the key values given in the optional argument.

```
58   \setkeys{venn}{#1}%
```

Calculate centre of set *C*

```
59   \pgfmathsetlength{\@venn@Cx}{\@venn@hgap + 2*\@venn@radius
60     -0.5*\@venn@overlap}%
61   \pgfmathsetlength{\@venn@Cy}{\@venn@vgap+\@venn@radius}%
```

Calculate centre of set *A*

```
62   \pgfmathsetlength{\@venn@Ax}{\@venn@hgap+\@venn@radius}%
63   \pgfmathsetlength{\@venn@Ay}{\@venn@Cy
64     + (\@venn@radius - 0.5*\@venn@overlap)*1.73205}%
```

Calculate centre of set *B*

```
65   \pgfmathsetlength{\@venn@Bx}{\@venn@hgap+3*\@venn@radius
66     -\@venn@overlap}%
67   \setlength{\@venn@By}{\@venn@Ay}%
```

Compute dimensions of entire diagram

```
68   \pgfmathsetlength{\@venn@w}{2*\@venn@hgap+4*\@venn@radius
69     -\@venn@overlap}%
70   \pgfmathsetlength{\@venn@h}{2*\@venn@vgap+4*\@venn@radius
71     -\@venn@overlap}%
```

Define filling commands. Fill all of set *A*:

```
72   \def\fillA{\path[fill=\@venn@shade] (\@venn@Ax,\@venn@Ay)
73     circle (\@venn@radius);}%
```

Fill all of set *B*:

```
74   \def\fillB{\path[fill=\@venn@shade] (\@venn@Bx,\@venn@By)
75     circle (\@venn@radius);}%
```

Fill all of set *C*:

```
76   \def\fillC{\path[fill=\@venn@shade] (\@venn@Cx,\@venn@Cy)
77     circle (\@venn@radius);}%
```

Fill everything:

```
78   \def\fillAll{\path[fill=\@venn@shade] (0,0)
79     rectangle (\@venn@w,\@venn@h);}%
```

Fill everything except set *A*:

```
80   \def\fillNotA{\path[fill=\@venn@shade,even odd rule]
81     (0,0) rectangle (\@venn@w,\@venn@h)
82     (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);}%
```

Fill everything except set *B*:

```
83   \def\fillNotB{\path[fill=\@venn@shade,even odd rule]
84      (0,0) rectangle (\@venn@w,\@venn@h)
85      (\@venn@Bx,\@venn@By) circle (\@venn@radius);}%
```

Fill everything except set *C*:

```
86   \def\fillNotC{\path[fill=\@venn@shade,even odd rule]
87      (0,0) rectangle (\@venn@w,\@venn@h)
88      (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);}%
```

Fill only set *A*:

```
89   \def\fillOnlyA{%
90      \begin{scope}
91      \path[name path=A] (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
92      \path[name path=BC] (\@venn@Bx,\@venn@By) circle (\@venn@radius)
93         (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
```

Get intersection points of paths A and BC

```
94      \path[name intersections={of=A and BC,name=ABintersect}]
95         (ABintersect-1) ;
96      \pgfgetlastxy{\@venn@AB@xi}{\@venn@AB@yi}
97      \path (ABintersect-2);
98      \pgfgetlastxy{\@venn@AB@xii}{\@venn@AB@yii}
```

Compute the start and end angles of arc between intersection points

```
99      \pgfmathparse{atan2(\@venn@AB@xi-\@venn@Ax,\@venn@AB@yi-\@venn@Ay)}
100     \let\@venn@start@i=\pgfmathresult
101     \pgfmathparse{360+atan2(\@venn@AB@xii-\@venn@Ax,\@venn@AB@yii-\@venn@Ay)}
102     \let\@venn@end@i=\pgfmathresult
```

Get intersection point of B and C

```
103     \path[name path=B] (\@venn@Bx,\@venn@By) circle (\@venn@radius);
104     \path[name path=C] (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
105     \path[name intersections={of=B and C,name=BCintersect}]
106        (BCintersect-1);
107     \pgfgetlastxy{\@venn@BC@x}{\@venn@BC@y}
```

Compute start and end angles

```
108     \pgfmathparse{atan2(\@venn@AB@xii-\@venn@Cx, \@venn@AB@yii-\@venn@Cy)}
109     \let\@venn@start@ii=\pgfmathresult
110     \pgfmathparse{atan2(\@venn@BC@x-\@venn@Cx, \@venn@BC@y-\@venn@Cy)}
111     \let\@venn@end@ii=\pgfmathresult
112     \pgfmathparse{atan2(\@venn@BC@x-\@venn@Bx, \@venn@BC@y-\@venn@By)}
113     \let\@venn@start@iii=\pgfmathresult
114     \pgfmathparse{atan2(\@venn@AB@xi-\@venn@Bx, \@venn@AB@yi-\@venn@By)-360}
115     \let\@venn@end@iii=\pgfmathresult
116     \path[fill=\@venn@shade] (ABintersect-1)
117      arc[radius=\@venn@radius,
118          start angle=\@venn@start@i,end angle=\@venn@end@i]
119      arc[radius=\@venn@radius,
120          start angle=\@venn@start@ii,end angle=\@venn@end@ii]
121      arc[radius=\@venn@radius,
```

14

```
122        start angle=\@venn@start@iii,end angle=\@venn@end@iii]
123      -- cycle;
124    \end{scope}
125  }%
```

Fill only set *B*:

```
126  \def\fillOnlyB{%
127    \begin{scope}
128    \path[name path=B] (\@venn@Bx,\@venn@By) circle (\@venn@radius);
129    \path[name path=AC] (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
130      (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
```

Get intersection points of B and AC

```
131    \path[name intersections={of=B and AC,name=BAintersect,sort by=B}]
132      (BAintersect-1);
133    \pgfgetlastxy{\@venn@BA@xi}{\@venn@BA@yi}
134    \path (BAintersect-4);
135    \pgfgetlastxy{\@venn@BA@xii}{\@venn@BA@yii}
```

Compute the start and end angles of arc between intersection points

```
136    \pgfmathparse{atan2(\@venn@BA@xi-\@venn@Bx,\@venn@BA@yi-\@venn@By)}
137    \let\@venn@start@i=\pgfmathresult
138    \pgfmathparse{atan2(\@venn@BA@xii-\@venn@Bx,\@venn@BA@yii-\@venn@By)}
139    \let\@venn@end@i=\pgfmathresult
```

Get intersection point of A and C

```
140    \path[name path=A] (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
141    \path[name path=C] (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
142    \path[name intersections={of=A and C,name=ACintersect}]
143      (ACintersect-2);
144    \pgfgetlastxy{\@venn@AC@x}{\@venn@AC@y}
```

Compute start and end angles

```
145    \pgfmathparse{atan2(\@venn@BA@xii-\@venn@Cx, \@venn@BA@yii-\@venn@Cy)}
146    \let\@venn@start@ii=\pgfmathresult
147    \pgfmathparse{atan2(\@venn@AC@x-\@venn@Cx, \@venn@AC@y-\@venn@Cy)}
148    \let\@venn@end@ii=\pgfmathresult
149    \pgfmathparse{atan2(\@venn@AC@x-\@venn@Ax, \@venn@AC@y-\@venn@Ay)}
150    \let\@venn@start@iii=\pgfmathresult
151    \pgfmathparse{atan2(\@venn@BA@xi-\@venn@Ax, \@venn@BA@yi-\@venn@Ay)}
152    \let\@venn@end@iii=\pgfmathresult
153    \path[fill=\@venn@shade] (BAintersect-1)
154    arc[radius=\@venn@radius,
155        start angle=\@venn@start@i,end angle=\@venn@end@i]
156    arc[radius=\@venn@radius,
157        start angle=\@venn@start@ii,end angle=\@venn@end@ii]
158    arc[radius=\@venn@radius,
159        start angle=\@venn@start@iii,end angle=\@venn@end@iii]
160    -- cycle ;
161    \end{scope}
162  }%
```

Fill only set *C*:

```
163  \def\fillOnlyC{%
164    \begin{scope}
165    \path[name path=C] (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
166    \path[name path=BA] (\@venn@Bx,\@venn@By) circle (\@venn@radius)
167      (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
```

Get intersection points of C and BA

```
168    \path[name intersections={of=C and BA,name=CBintersect,sort by=C}]
169      (CBintersect-1);
170    \pgfgetlastxy{\@venn@CB@xi}{\@venn@CB@yi}
171    \path (CBintersect-4);
172    \pgfgetlastxy{\@venn@CB@xii}{\@venn@CB@yii}
```

Compute the start and end angles of arc between intersection points

```
173    \pgfmathparse{atan2(\@venn@CB@xi-\@venn@Cx,\@venn@CB@yi-\@venn@Cy)}
174    \let\@venn@start@i=\pgfmathresult
175    \pgfmathparse{atan2(\@venn@CB@xii-\@venn@Cx,\@venn@CB@yii-\@venn@Cy)-360}
176    \let\@venn@end@i=\pgfmathresult
```

Get intersection point of B and A

```
177    \path[name path=B] (\@venn@Bx,\@venn@By) circle (\@venn@radius);
178    \path[name path=A] (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
179    \path[name intersections={of=B and A,name=BAintersect}]
180      (BAintersect-2);
181    \pgfgetlastxy{\@venn@BA@x}{\@venn@BA@y}
```

Compute start and end angles

```
182    \pgfmathparse{atan2(\@venn@CB@xii-\@venn@Ax, \@venn@CB@yii-\@venn@Ay)}
183    \let\@venn@start@ii=\pgfmathresult
184    \pgfmathparse{atan2(\@venn@BA@x-\@venn@Ax, \@venn@BA@y-\@venn@Ay)}
185    \let\@venn@end@ii=\pgfmathresult
186    \pgfmathparse{atan2(\@venn@BA@x-\@venn@Bx, \@venn@BA@y-\@venn@By)}
187    \let\@venn@start@iii=\pgfmathresult
188    \pgfmathparse{atan2(\@venn@CB@xi-\@venn@Bx, \@venn@CB@yi-\@venn@By)}
189    \let\@venn@end@iii=\pgfmathresult
190    \path[fill=\@venn@shade] (CBintersect-1)
191    arc[radius=\@venn@radius,
192        start angle=\@venn@start@i,end angle=\@venn@end@i]
193    arc[radius=\@venn@radius,
194        start angle=\@venn@start@ii,end angle=\@venn@end@ii]
195    arc[radius=\@venn@radius,
196        start angle=\@venn@start@iii,end angle=\@venn@end@iii]
197    -- cycle;
198    \end{scope}
199  }%
```

Fill everything except *A, B* or *C*.

```
200  \def\fillNotABC{%
201    \begin{scope}
202    \path[name path=A] (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
```

```
203    \path[name path=B] (\@venn@Bx,\@venn@By) circle (\@venn@radius);
204    \path[name path=C] (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
```

Get first intersection point of A and B

```
205    \path[name intersections={of=A and B,name=ABintersect}]
206      (ABintersect-1);
207    \pgfgetlastxy{\@venn@AB@x}{\@venn@AB@y}
```

Get intersection point of A and C

```
208    \path[name intersections={of=A and C,name=ACintersect}]
209      (ACintersect-1);
210    \pgfgetlastxy{\@venn@AC@x}{\@venn@AC@y}
```

Get intersection point of C and B

```
211    \path[name intersections={of=C and B,name=CBintersect}]
212      (CBintersect-1);
213    \pgfgetlastxy{\@venn@CB@x}{\@venn@CB@y}
```

Compute start and end angles for first arc

```
214    \pgfmathparse{atan2(\@venn@AB@x-\@venn@Ax, \@venn@AB@y-\@venn@Ay)}
215    \let\@venn@start@i=\pgfmathresult
216    \pgfmathparse{atan2(\@venn@AC@x-\@venn@Ax, \@venn@AC@y-\@venn@Ay)+360}
217    \let\@venn@end@i=\pgfmathresult
```

Compute start and end angles for second arc

```
218    \pgfmathparse{atan2(\@venn@AC@x-\@venn@Cx,\@venn@AC@y-\@venn@Cy)}
219    \let\@venn@start@ii=\pgfmathresult
220    \pgfmathparse{atan2(\@venn@CB@x-\@venn@Cx,\@venn@CB@y-\@venn@Cy)+360}
221    \let\@venn@end@ii=\pgfmathresult
```

Compute start and end angles for third arc

```
222    \pgfmathparse{atan2(\@venn@CB@x-\@venn@Bx,\@venn@CB@y-\@venn@By)}
223    \let\@venn@start@iii=\pgfmathresult
224    \pgfmathparse{atan2(\@venn@AB@x-\@venn@Bx,\@venn@AB@y-\@venn@By)}
225    \let\@venn@end@iii=\pgfmathresult
226  \path[fill=\@venn@shade]
227    (0,0) rectangle (\@venn@w,\@venn@h)
228    (ABintersect-1)
229    arc[radius=\@venn@radius,
230        start angle=\@venn@start@i,end angle=\@venn@end@i]
231    arc[radius=\@venn@radius,
232        start angle=\@venn@start@ii,end angle=\@venn@end@ii]
233    arc[radius=\@venn@radius,
234        start angle=\@venn@start@iii,end angle=\@venn@end@iii]
235    -- cycle;
236  \end{scope}
237 }%
```

Fill *A* but not *B*

```
238  \def\fillANotB{%
239    \begin{scope}
240      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
```

```
241      \path[fill=\@venn@shade,even odd rule]
242        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
243        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
244    \end{scope}
245  }%
```

Fill *B* but not *A*

```
246  \def\fillBNotA{%
247    \begin{scope}
248      \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
249      \path[fill=\@venn@shade,even odd rule]
250        (\@venn@Bx,\@venn@By) circle (\@venn@radius)
251        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
252    \end{scope}
253  }%
```

Fill *A* but not *C*

```
254  \def\fillANotC{%
255    \begin{scope}
256      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
257      \path[fill=\@venn@shade,even odd rule]
258        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
259        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
260    \end{scope}
261  }%
```

Fill *C* but not *A*

```
262  \def\fillCNotA{%
263    \begin{scope}
264      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
265      \path[fill=\@venn@shade,even odd rule]
266        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
267        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
268    \end{scope}
269  }%
```

Fill *B* but not *C*

```
270  \def\fillBNotC{%
271    \begin{scope}
272      \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
273      \path[fill=\@venn@shade,even odd rule]
274        (\@venn@Bx,\@venn@By) circle (\@venn@radius)
275        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
276    \end{scope}
277  }%
```

Fill *C* but not *B*

```
278  \def\fillCNotB{%
279    \begin{scope}
280      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
281      \path[fill=\@venn@shade,even odd rule]
```

```
282          (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
283          (\@venn@Bx,\@venn@By) circle (\@venn@radius);
284      \end{scope}
285  }%
```

Fill *A* intersect *B*

```
286  \def\fillACapB{%
287      \begin{scope}
288        \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
289        \path[fill=\@venn@shade]
290          (\@venn@Bx,\@venn@By) circle (\@venn@radius);
291      \end{scope}
292  }%
```

Define a synonym:

```
293  \let\fillBCapA\fillACapB
```

Fill *A* intersect *C*

```
294  \def\fillACapC{%
295      \begin{scope}
296        \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
297        \path[fill=\@venn@shade]
298          (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
299      \end{scope}
300  }%
```

Define a synonym:

```
301  \let\fillCCapA\fillACapC
```

Fill *B* intersect *C*

```
302  \def\fillBCapC{%
303      \begin{scope}
304        \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
305        \path[fill=\@venn@shade]
306          (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
307      \end{scope}
308  }%
```

Define a synonym:

```
309  \let\fillCCapB\fillBCapC
```

Fill *A* intersect *B* but not *C*

```
310  \def\fillACapBNotC{%
311      \begin{scope}
312        \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
313        \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
314        \path[fill=\@venn@shade,even odd rule]
315          (\@venn@Bx,\@venn@By) circle (\@venn@radius)
316          (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
317      \end{scope}
318  }%
```

Define a synonym:

```
319  \let\fillBCapANotC\fillACapBNotC
```

Fill *A* intersect *C* but not *B*

```
320  \def\fillACapCNotB{%
321    \begin{scope}
322      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
323      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
324      \path[fill=\@venn@shade,even odd rule]
325        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
326        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
327    \end{scope}
328  }%
```

Define a synonym:

```
329  \let\fillCCapANotB\fillACapCNotB
```

Fill *B* intersect *C* but not *A*

```
330  \def\fillBCapCNotA{%
331    \begin{scope}
332      \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
333      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
334      \path[fill=\@venn@shade,even odd rule]
335        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
336        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
337    \end{scope}
338  }%
```

Define a synonym:

```
339  \let\fillCCapBNotA\fillBCapCNotA
```

Fill the intersection of all three sets

```
340  \def\fillACapBCapC{%
341    \begin{scope}
342      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
343      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
344      \path[fill=\@venn@shade]
345        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
346    \end{scope}
347  }%
```

Define synonyms:

```
348  \let\fillACapCCapB\fillACapBCapC
349  \let\fillBCapACapC\fillACapBCapC
350  \let\fillBCapCCapA\fillACapBCapC
351  \let\fillCCapACapB\fillACapBCapC
352  \let\fillCCapBCapA\fillACapBCapC
```

Start the tikzpicture environment.

```
353  \ifdefempty{\@venn@tikzoptions}%
354  {%
355    \def\@venn@dobegin{\begin{tikzpicture}}}%
```

```
356    }%
357    {%
358      \edef\@venn@dobegin{\noexpand\begin{tikzpicture}%
359        [\expandonce\@venn@tikzoptions]}%
360    }%
361    \@venn@dobegin
```

coordinates of the Venn diagram corners

```
362    \path (0,0) coordinate (venn bottom left)
363          (0,\@venn@h) coordinate (venn top left)
364          (\@venn@w,\@venn@h) coordinate (venn top right)
365          (\@venn@w,0) coordinate (venn bottom right);
366  }%
```

End environment code:

```
367 {%
```

Draw outlines

```
368      \draw (0,0) rectangle (\@venn@w,\@venn@h);
369      \draw (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
370      \draw (\@venn@Bx,\@venn@By) circle (\@venn@radius);
371      \draw (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
```

Draw labels

```
372      \draw (\@venn@Ax,\@venn@Ay) node[above,left] (labelOnlyA) {\@venn@label@OnlyA};
373      \draw (\@venn@Bx,\@venn@By) node[above,right] (labelOnlyB) {\@venn@label@OnlyB};
374      \draw (\@venn@Cx,\@venn@Cy) node[below] (labelOnlyC) {\@venn@label@OnlyC};
```

Region labels

```
375      \draw (\@venn@vgap,\@venn@hgap) node (labelNotABC) {\@venn@label@NotABC};
376      \draw (\@venn@Ax,\@venn@Ay+\@venn@radius)
377        node[below] (labelA) {\@venn@label@A};
378      \draw (\@venn@Bx,\@venn@By+\@venn@radius)
379        node[below] (labelB) {\@venn@label@B};
380      \draw (\@venn@Cx,\@venn@vgap) node[above] (labelC) {\@venn@label@C};
381      \draw (\@venn@Cx,0.5*\@venn@h) node (labelABC) {\@venn@label@ABC};
382      \draw (\@venn@Cx,\@venn@Ay) node[above] (labelOnlyAB) {\@venn@label@OnlyAB};
383      \draw (\@venn@Ax,\@venn@Ay) ++(-60:\@venn@radius-0.5*\@venn@overlap)
384        node[below left] (labelOnlyAC) {\@venn@label@OnlyAC};
385      \draw (\@venn@Bx,\@venn@By) ++(-120:\@venn@radius-0.5*\@venn@overlap)
386        node[below right] (labelOnlyBC) {\@venn@label@OnlyBC};
387      \@postvennhook
388    \end{tikzpicture}
389  }
```

\@postvennhook   Hook called just before the end of the tikzpicture environment.

```
390 \newcommand*{\@postvennhook}{}
```

\setpostvennhook   User interface to set the post hook.

```
391 \newcommand*{\setpostvennhook}[1]{\def\@postvennhook{#1}}
```

392 `\newenvironment{venndiagram2sets}[1][]%`
393 `{%`

Disable the keys that aren't applicable.

394 `  \disable@keys{venn}{labelABC,labelOnlyC,labelOnlyAC,labelOnlyBC,%`
395 `    labelNotABC,labelC,labelOnlyAB}%`

Set the key values given in the optional argument.

396 `  \setkeys{venn}{#1}%`

Calculate centre of A

397 `  \pgfmathsetlength{\@venn@Ax}{\@venn@hgap+\@venn@radius}%`
398 `  \pgfmathsetlength{\@venn@Ay}{\@venn@vgap+\@venn@radius}%`

Calculate centre of B

399 `  \pgfmathsetlength{\@venn@Bx}{\@venn@hgap+3*\@venn@radius`
400 `    -\@venn@overlap}%`
401 `  \setlength{\@venn@By}{\@venn@Ay}%`

Compute dimensions of entire diagram

402 `  \pgfmathsetlength{\@venn@w}{2*\@venn@hgap+4*\@venn@radius`
403 `    -\@venn@overlap}%`
404 `  \pgfmathsetlength{\@venn@h}{2*\@venn@vgap+2*\@venn@radius}%`

Define filling commands

405 `  \def\fillA{\path[fill=\@venn@shade] (\@venn@Ax,\@venn@Ay)`
406 `    circle (\@venn@radius);}%`
407 `  \def\fillB{\path[fill=\@venn@shade] (\@venn@Bx,\@venn@By)`
408 `    circle (\@venn@radius);}%`
409 `  \def\fillAll{\path[fill=\@venn@shade] (0,0)`
410 `    rectangle (\@venn@w,\@venn@h);}%`
411 `  \def\fillOnlyA{%`
412 `    \begin{scope}`
413 `    \path[name path=A] (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);`
414 `    \path[name path=B] (\@venn@Bx,\@venn@By) circle (\@venn@radius);`

Get intersection points of A and B

415 `    \path[name intersections={of=A and B,name=ABintersect}]`
416 `      (ABintersect-1) ;`
417 `    \pgfgetlastxy{\@venn@AB@xi}{\@venn@AB@yi}`
418 `    \path (ABintersect-2);`
419 `    \pgfgetlastxy{\@venn@AB@xii}{\@venn@AB@yii}`

Compute the start and end angles of arc between intersection points

420 `    \pgfmathparse{atan2(\@venn@AB@xi-\@venn@Ax,\@venn@AB@yi-\@venn@Ay)}`
421 `    \let\@venn@start@i=\pgfmathresult`
422 `    \pgfmathparse{360+atan2(\@venn@AB@xii-\@venn@Ax,\@venn@AB@yii-\@venn@Ay)}`
423 `    \let\@venn@end@i=\pgfmathresult`

Compute start and end angles

424 `    \pgfmathparse{atan2(\@venn@AB@xii-\@venn@Bx, \@venn@AB@yii-\@venn@By)}`
425 `    \let\@venn@start@ii=\pgfmathresult`

22

```
426    \pgfmathparse{atan2(\@venn@AB@xi-\@venn@Bx, \@venn@AB@yi-\@venn@By)-360}
427    \let\@venn@end@ii=\pgfmathresult
428    \path[fill=\@venn@shade] (ABintersect-1)
429     arc[radius=\@venn@radius,
430         start angle=\@venn@start@i,end angle=\@venn@end@i]
431     arc[radius=\@venn@radius,
432         start angle=\@venn@start@ii,end angle=\@venn@end@ii]
433     -- cycle;
434    \end{scope}
435  }%
```

Fill only set *B*

```
436  \def\fillOnlyB{%
437    \begin{scope}
438    \path[name path=A] (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
439    \path[name path=B] (\@venn@Bx,\@venn@By) circle (\@venn@radius);
```

Get intersection points of A and B

```
440    \path[name intersections={of=A and B,name=ABintersect}]
441      (ABintersect-1) ;
442    \pgfgetlastxy{\@venn@AB@xi}{\@venn@AB@yi}
443    \path (ABintersect-2);
444    \pgfgetlastxy{\@venn@AB@xii}{\@venn@AB@yii}
```

Compute the start and end angles of arc between intersection points

```
445    \pgfmathparse{atan2(\@venn@AB@xi-\@venn@Bx,\@venn@AB@yi-\@venn@By)}
446    \let\@venn@start@i=\pgfmathresult
447    \pgfmathparse{atan2(\@venn@AB@xii-\@venn@Bx,\@venn@AB@yii-\@venn@By)}
448    \let\@venn@end@i=\pgfmathresult
```

Compute start and end angles

```
449    \pgfmathparse{atan2(\@venn@AB@xii-\@venn@Ax, \@venn@AB@yii-\@venn@Ay)}
450    \let\@venn@start@ii=\pgfmathresult
451    \pgfmathparse{atan2(\@venn@AB@xi-\@venn@Ax, \@venn@AB@yi-\@venn@Ay)}
452    \let\@venn@end@ii=\pgfmathresult
453    \path[fill=\@venn@shade] (ABintersect-1)
454     arc[radius=\@venn@radius,
455         start angle=\@venn@start@i,end angle=\@venn@end@i]
456     arc[radius=\@venn@radius,
457         start angle=\@venn@start@ii,end angle=\@venn@end@ii]
458     -- cycle;
459    \end{scope}
460  }%
```

Fill everything except *A*

```
461  \def\fillNotA{\path[fill=\@venn@shade,even odd rule]
462    (0,0) rectangle (\@venn@w,\@venn@h)
463    (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);}%
```

Fill everything except *B*

```
464  \def\fillNotB{\path[fill=\@venn@shade,even odd rule]
465    (0,0) rectangle (\@venn@w,\@venn@h)
```

```
466      (\@venn@Bx,\@venn@By) circle (\@venn@radius);}%
```

Fill everything except *A* or *B* $((A \cup B)^c)$

```
467    \def\fillNotAorB{%
468      \begin{scope}
469      \path[clip]
470      (0,0) rectangle (\@venn@w,\@venn@h)
471      (\@venn@Bx,\@venn@By) circle (\@venn@radius)
472       ;
473      \path[fill=\@venn@shade,even odd rule]
474      (0,0) rectangle (\@venn@w,\@venn@h)
475      (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
476      ;
477      \end{scope}
478    }%
```

Fill not *A* or not *B* $((A \cap B)^c)$

```
479    \def\fillNotAorNotB{%
480      \path[fill=\@venn@shade,nonzero rule]
481      (0,0) rectangle (\@venn@w,\@venn@h)
482      (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
483      (0,0) rectangle (\@venn@w,\@venn@h)
484      (\@venn@Bx,\@venn@By) circle (\@venn@radius)
485      ;
486    }%
```

Fill *A* but not *B*

```
487    \def\fillANotB{%
488      \begin{scope}
489        \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
490        \path[fill=\@venn@shade,even odd rule]
491          (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
492          (\@venn@Bx,\@venn@By) circle (\@venn@radius);
493      \end{scope}
494    }%
```

Fill *B* but not *A*

```
495    \def\fillBNotA{%
496      \begin{scope}
497        \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
498        \path[fill=\@venn@shade,even odd rule]
499          (\@venn@Bx,\@venn@By) circle (\@venn@radius)
500          (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
501      \end{scope}
502    }%
```

Fill *A* intersect *B*

```
503    \def\fillACapB{%
504      \begin{scope}
505        \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
506        \path[fill=\@venn@shade]
```

```
507        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
508    \end{scope}
509  }%
```

Define synonym:

```
510  \let\fillBCapA\fillACapB
```

Start the tikzpicture environment.

```
511  \ifdefempty{\@venn@tikzoptions}%
512  {%
513    \def\@venn@dobegin{\begin{tikzpicture}}%
514  }%
515  {%
516    \edef\@venn@dobegin{\noexpand\begin{tikzpicture}%
517      [\expandonce\@venn@tikzoptions]}%
518  }%
519  \@venn@dobegin
```

coordinates of the Venn diagram corners

```
520  \path (0,0) coordinate (venn bottom left)
521        (0,\@venn@h) coordinate (venn top left)
522        (\@venn@w,\@venn@h) coordinate (venn top right)
523        (\@venn@w,0) coordinate (venn bottom right);
524 }%
```

End environment code

```
525 {%
```

Draw outlines

```
526    \draw (venn bottom left) rectangle (\@venn@w,\@venn@h);
527    \draw (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
528    \draw (\@venn@Bx,\@venn@By) circle (\@venn@radius);
```

Draw labels

```
529    \draw (\@venn@Ax,\@venn@Ay) node[above,left] (labelOnlyA)
530      {\@venn@label@OnlyA};
531    \draw (\@venn@Bx,\@venn@By) node[above,right] (labelOnlyB)
532      {\@venn@label@OnlyB};
```

Region labels

```
533    \draw (\@venn@vgap,\@venn@hgap) node (labelNotAB) {\@venn@label@NotAB};
534    \draw (\@venn@Ax,\@venn@Ay+\@venn@radius)
535      node[below] (labelA) {\@venn@label@A};
536    \draw (\@venn@Bx,\@venn@By+\@venn@radius)
537      node[below] (labelB) {\@venn@label@B};
538    \draw (0.5*\@venn@w,0.5*\@venn@h) node (labelAB) {\@venn@label@AB};
539    \@postvennhook
540  \end{tikzpicture}
541 }
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.