

tikz mindmap 调整节点位置的方法

宿宝臣 <subaochen@126.com>

2017 年 5 月 18 日

通常，tikz mindmap 自动生成的图形是可以接受的，如果要调整节点的位置，可以采取的手段有：

- **level distance** 决定了节点之间的距离。一般情况下，可以尝试让 tikz 自动决定节点之间的间距，如果必要，可以定义每一个层次的节点距离，比如：

```
level 1/.append style={level distance=3.5cm}  
level 2/.append style={level distance=2cm}
```

- **sibling angle** 决定了节点之间的间隔角度。一般的，需要定义一级节点的间隔角度，比如总共有 3 个一级节点，则一般定义其间隔角度为 120 平均分布：

```
level 1/.append style={level distance=3.5cm,sibling angle=120}
```

- **grow** 强制该节点从上一级节点的哪个方向（角度）开始“生长”。该命令非常强悍，但是副作用也比较大，可能引致连锁反应，和自动布局的节点重叠。一般的，如果某个节点使用了 **grow** 设置生长方向，另外受影响的节点也要随之设置 **grow** 属性，并且需要配合 **clockwise from** 纠正节点的布局。示例中，为了让“知识目标”位于“教学目标”的右边展示设置了 **grow=20**，并且 **clockwise from=100** 使得“知识目标”的节点能够均衡的展示。：

```
\node[root concept] {教学目标}  
child[ecommercial platform, visible on=<2->,grow=20] { node{知识目标}  
[clockwise from = 100]  
child { node {熟练配置 Java 开发环境} }  
child { node[important] {掌握面向对象的基本概念} }  
child { node {掌握 Java 的常用类} }  
child { node {掌握 Java 的异常处理方法} }  
child { node {掌握 Java 的 IO 和图形用户界面设计} }  
}
```

- **clockwise from** 影响节点从哪个角度开始布置，按照此设置角度的顺时针方向开始布局节点。
- **xshift/yshift** 可以在 x 方向和 y 方向微调节点的位置，比如：

```
child { node[xshift=-3mm] {掌握 Java 的 IO 和图形用户界面设计} }
```

综合示例如图 1 所示，其源代码代码列表。

```
% Keys to support piece-wise uncovering of elements in TikZ pictures:  
% \node[visible on=<2->](foo){Foo}
```

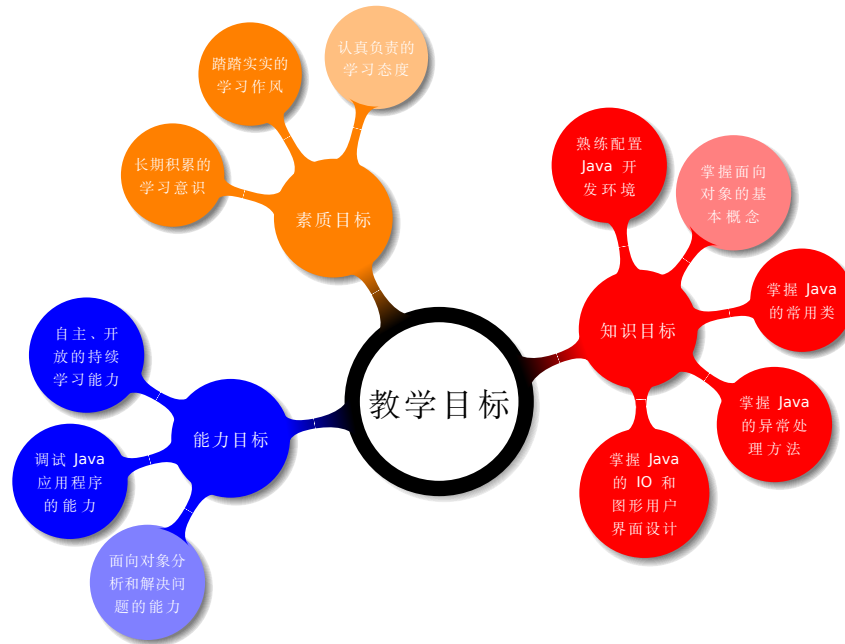


图 1 - tikz mindmap 示例

```
% \node[visible on=<{2,4}>](bar){Bar} % put braces around comma expressions
%
% Internally works by setting opacity=0 when invisible, which has the
% advantage (compared to \node<2->(foo){Foo} that the node is always there, hence
% always consumes space plus that coordinate (foo) is always available.
%
% The actual command that implements the invisibility can be overridden
% by altering the style invisible. For instance \tikzset{invisible/.style={opacity
% =0.2}}
% would dim the "invisible" parts. Alternatively, the color might be set to white, if
% the
% output driver does not support transparencies (e.g., PS)
%
\tikzset{
  invisible/.style={opacity=0},
  visible on/.style={alt={#1}{invisible}}},
  alt/.code args={<#1>#2#3}{%
    \alt<#1>{\pgfkeysalso{#2}}{\pgfkeysalso{#3}} % \pgfkeysalso doesn't change the
    path
  },
}
\usetikzlibrary{babel,mindmap,shadows,trees,graphs,graphdrawing}
\begin{tikzpicture} [small mindmap,
  % \hskip0pt will add hyphen to first word
  every node/.style={concept,circular drop shadow,execute at begin node = \hskip0pt
  },
  root concept/.append style={concept color=black, fill=white, line width=1ex,text=
  black,font=\large\scshape},
  text=white,
  ecommercial platform/.style={concept color=red,important/.style={concept color=red
  !50}},
  consult platform/.style={concept color=blue,important/.style={concept color=blue
```

```

!50}},
sharing platform/.style={concept color=orange,important/.style={concept color=
orange!50}},
vr platform/.style={concept color=green!50!black,important/.style={concept color=
green!50!black!50}},
grow cyclic,
level 1/.append style={sibling angle=120},
level 2/.append style={sibling angle=45},
]

\node[root concept] {教学目标} % root
child[ecommercial platform, visible on=<2->,grow=20] { node{知识目标}
[clockwise from = 100]
child { node {熟练配置Java开发环境} }
child { node[important] {掌握面向对象的基本概念} }
child { node {掌握Java的常用类} }
child { node {掌握Java的异常处理方法} }
child { node[xshift=-3mm] {掌握Java的IO和图形用户界面设计} }
}
child [consult platform, visible on=<3->,grow=-170] { node {能力目标}
[clockwise from = -120]
child { node[important] {面向对象分析和解决问题的能力} }
child { node {调试Java应用程序的能力} }
child { node {自主、开放的持续学习能力} }
}
child[sharing platform, visible on=<4->] { node {素质目标}
child { node[important] {认真负责的学习态度} }
child { node {踏踏实实的学习作风} }
child { node {长期积累的学习意识} }
}
}
;

\end{tikzpicture}

```