codeanatomy – Draw Code Anatomy*

Reference

Hồng-Phúc Bùi†

Released 2025/10/16

Contents

1 Hints **1**

2 Implementation **1**
2.1 Package Dependenies 1
2.2 Setup styles 2
2.2.1 Colors 2
2.2.2 Ti*k*Z styles for code in a Code Anatomy 2
2.3 Command used to set code and code anatomy 4

3 Known Bugs **6**

Index **7**

Change History **7**

1 Hints

Usage of this Package can be found in `codeanatomy.usage.pdf` and `codeanatomy.lstlisting.pdf`. This document show only generated reference of commands in this Package.

2 Implementation

2.1 Package Dependenies

```
\RequirePackage{expl3}
\RequirePackage{xparse}
\RequirePackage{tikz}
```

Load necsessary Ti*k*Z libraries.

```
\usetikzlibrary{
```

*This file describes v0.5-Beta, last revised 2025/10/16.

†E-mail: hong-phuc.bui (at) htwsaar dot de

```
    tikzmark
   ,fit
   ,arrows.meta
   ,bending
   ,shapes
   ,chains
   ,backgrounds
   ,scopes
   ,decorations
   ,decorations.pathmorphing
}
```

2.2 Setup styles

2.2.1 Colors

Define colors which are used in codeanatomy

annotationcolor 

```
\definecolor{annotationcolor}
            {rgb}{0,0.50002,1} % Blue
```

bgcmdcolor 

```
\colorlet{bgcmdcolor}{gray}    % Grey
```

2.2.2 Ti*k*Z styles for code in a Code Anatomy

anatomy Ti*k*Z style for annotation labels:
`\tikz{\node(code) [anatomy] at (0,0) {code line 1\\code line 2}; }`
code line 1
yields code line 2

```
\tikzset{anatomy/.style={%
      anchor=south west,%
      inner sep=0,%
      align=left,%
      font=\ttfamily
    }
}
```

code part Ti*k*Z style to mark a piece of code in an anatomy:
`\tikz{\node(code) [code part] at (0,0) {let a = 12;};}`
yields `let a = 12;`

```
\tikzset{code part/.style={%
      rectangle,%
      draw=annotationcolor,%
      align=left,%
      minimum height=1.175em,%
      inner sep=1.75pt,%
      outer sep=0.1pt,%
      font=\ttfamily
    }
}
```

ignored code part Ti*k*Z style to make a pice of code in an anatomy as not important in currently talking context `\tikz{\node(ignore code) [ignored code part] at (0,0) {/*some comment*/} }`
yields `/*some comment*/`

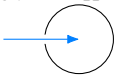
```
\tikzset{ignored code part/.style={%
    code part,%
    draw=none,color=bgcmdcolor,%
    inner sep=0.75pt
  }
}
```

`fit extrem` TikZ style to mark a piece of multiple line code in an anatomy:
`\tikz{ \node(c)[fit extrem, fit={(0,0) (0.5,0.975) (1,0)}] {}; }`

yields 

```
\tikzset{fit extrem/.style={%
      rectangle,%
      draw=annotationcolor,%
      align=left,%
      minimum height=1.175em,%
      inner sep=1.75pt,%
      outer sep=0.1pt,%
      font=\ttfamily
    }
}
```

`annotation` TikZ style of arrows from annotation labels to code parts:
`\tikz{\draw[] (1,0) circle(3ex); \draw[->,annotation] (0,0) -- (1,0);}`

yields 

```
\tikzset{annotation/.style={%
      preaction={
          draw=white,%
          line width=3.5pt,%
          arrows={-Triangle Cap[]},%
      },%
      draw=annotationcolor,%
      arrows={-Latex[%
          round,%
          color=annotationcolor,
          fill=annotationcolor
        ]
      },
      shorten >=0.25pt
    }
}
```

`code annotation` TikZ style for a annotation label function name

```
\tikzset{code annotation/.style={%
      inner sep=2pt,%
      text=annotationcolor,%
      align=center,%
      font=\sffamily\small
    }
}
```

`code grid debug` TikZ style to draw debug grid on the background of anatomy

```
\tikzset{code grid debug/.style={%
      step=1.0,%
      draw=gray!20,%
      very thin,%
      on background layer
    }
}
```

2.3 Command used to set code and code anatomy

`\codeBlock` `{`⟨*code*⟩`}`

Complete code listing of a Code Anatomy figure is typeset by this command. Whereas `{`⟨*code*⟩`}` is the *formatted* code listing. This command can be used if there are no other packages to typeset code listing in use.

```
\NewDocumentCommand{\codeBlock}{m}%
    {\node(code) [anatomy] at (0,0) {#1};}%
```

`\cPart` `[`⟨*style*⟩`]` `{`⟨*node name*⟩`}{`⟨*piece of code*⟩`}`

Assign a piece of typeset code –typical in one line– to a Ti*k*Z Node, so that it can be annotated.

- `[`⟨*style*⟩`]` a defined Ti*k*Z style to be applied to this node, the style `code part` is applied to the node per default.
- `{`⟨*node name*⟩`}` is a unique Ti*k*Z node name in the `tikzpicture`
- `{`⟨*piece of code*⟩`}` is a single code part to be marked.

```
\NewDocumentCommand\cPart{O{code part}mm}
   {\tikzmarknode[#1]{#2}{#3}}
```

`\iPart` `{`⟨*node name*⟩`}{`⟨*piece of code*⟩`}`

Assign a piece of typeset code –typical in one line– to a Ti*k*Z Node, so that it can be annotated. It does not plot border around the pice of code as `\cPart` does.

- `[`⟨*style*⟩`]` a defined Ti*k*Z style to be applied to this node, the style `ignored code part` is applied to the node per default.
- `{`⟨*node name*⟩`}` is a unique Ti*k*Z node name in the `tikzpicture`
- `{`⟨*piece of code*⟩`}` is a single code part to be marked.

```
\NewDocumentCommand{\iPart}{O{ignored code part}mm} %
    {\tikzmarknode[#1]{#2}{#3}}
```

`\mtPoint` `{`⟨*node name*⟩`}`

Marks a point as a **m**ost **t**op in a Code Block.

```
\NewDocumentCommand{\mtPoint}{m}
    {\tikzmarknode{#1}{\phantom{\rule[1.8ex]{0.1ex}{0.1ex}}}}
```

`\hmtPoint` `{`⟨*node name*⟩`}`

Marks a point as a **h**eigher **m**ost **t**op point in a Code Block.

```
\NewDocumentCommand{\hmtPoint}{m}
    {\tikzmarknode{#1}{\phantom{\rule[2.5ex]{0.1ex}{0.1ex}}}}
```

`\mbPoint` `{`⟨*node name*⟩`}`

Marks a point as a **d**eeper **m**ost **b**ottom point in a Code Block.

```
\NewDocumentCommand{\mbPoint}{m}
    {\tikzmarknode{#1}{\phantom{\rule[-0.55ex]{0.1ex}{0.1ex}}}}
```

`\dmbPoint` `{`⟨*node name*⟩`}`

Marks a point as a **d**eeper **m**ost **b**ottom point in a Code Block.

```
\NewDocumentCommand{\dmbPoint}{m}
    {\tikzmarknode{#1}{\phantom{\rule[-2ex]{0.1ex}{0.1ex}}}}
```

`\extremPoint` `{`⟨*node name*⟩`}[`⟨*yshift*⟩`][`⟨*xshift*⟩`][`⟨*style*⟩`]`

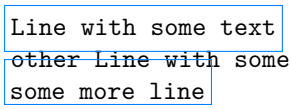
Create a Ti*k*Z Node as reference point for later use in `\fitExtrem`.

- `{`⟨*node name*⟩`}` is the Ti*k*Z node name which is used in `\fitExtrem` to reference to this point
- `[`⟨*yshift*⟩`]` a length, default 0ex which places this markpoint on the base line, shift this mark point vertical, for positive value over base line, negative value under base line.
- `[`⟨*xshift*⟩`]` same as `[`⟨*yshift*⟩`]` but for horizontal direction.
- `[`⟨*style*⟩`]` is a Ti*k*Z style (may be defined by user).

For example:

```
\begin{tikzpicture}[remember picture]
\node(code) [anatomy] at (0,0) {
   \extremPoint{tl}[2ex]Line with some text\extremPoint{br}[-1ex]\\
   \extremPoint{tl2}other Line with some text\\
   some more line\extremPoint{br2}\\
};
\fitExtrem{box1}{(tl) (br)}
\fitExtrem{box2}{(tl2) (br2)}
\end{tikzpicture}
```

yields




```
\NewDocumentCommand{\extremPoint}{m O{0ex} O{0.1ex} O{} }
    {\tikzmarknode[#4]{#1}{\phantom{\rule[#2]{#3}{0.1ex}}}}
```

`\fitExtrem` `{`⟨*node name*⟩`}{`⟨*extrem points*⟩`}`

Create a rectangle box over given extrem points defined by `\*Point{}`.

- `{`⟨*node name*⟩`}` is a unique Ti*k*Z node name in the current anatomy
- `{`⟨*extrem points*⟩`}` is a list of Ti*k*Z node name created by `\*Point`, each name is surrounded by `()`.

Example:

```
\begin{tikzpicture}[remember picture]
\node(code) [anatomy] at (0,0) {
\mtPoint{left}Line 1\\
Long Line 2\extremPoint{right}\\
Line 3\mbPoint{bottom}
};
\fitExtrem{box} { (left) (bottom) (right) }
\end{tikzpicture}
```

yields

```
Line 1
Long Line 2
Line 3
```

```
\NewDocumentCommand{\fitExtrem}{mm}
    {\node(#1)[fit extrem,fit={#2}]{};}
```

`\bgcode` `{`⟨*piece of code*⟩`}`
Typeset a piece of code in color `bgcmdcolor`. For example
`\tikz{\codeBlock{let a := 12\bgcode{;}}}`
yields `let a := 12;`

```
\NewDocumentCommand{\bgcode}{m}{\textcolor{bgcmdcolor}{#1}}
```

`\ptab` Produce a horizontal space of 4 small characters `h` respective 1 small character `h`
`\phspace` for example: `\tikz{\codeBlock{a\ptab{}b}}` yields `a    b`

```
\NewDocumentCommand{\ptab}{}{\phantom{hhhh}}
\NewDocumentCommand{\phspace}{}{\phantom{h}}
```

`\codeAnnotation` `{`⟨*node name*⟩`}(`⟨*coordinate*⟩`){`⟨*label text*⟩`}`
Typeset Annotation labels for a code part.

- `{`⟨*node name*⟩`}` is a unique Ti*k*Z node name in the `tikzpicture`,
- `(`⟨*coordinate*⟩`)` is the coordinate of the annotation label, surrounded by a `()`,
- `{`⟨*label text*⟩`}` text content to be typeset.

For example:

```
\begin{tikzpicture}[remember picture]
\codeBlock{a \cPart{a}{:=} 12 + 13}
\codeAnnotation{codeLabel} (1,-0.5) {assignment}
\draw[->,annotation] (codeLabel) -- (a);
\end{tikzpicture}
```

yields

```
a := 12 + 13
```
assignment

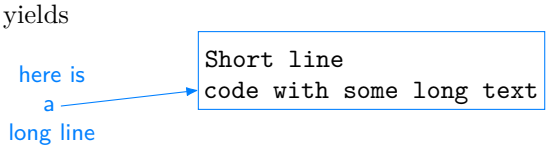
```
\NewDocumentCommand{\codeAnnotation}{m r() m } %
    { \node(#1)[code annotation] at (#2) {#3}; }
```

3 Known Bugs

~~**Arrows color**~~ Arrows appear in some cases with mysterious color. I don't know why! For example:

```
\begin{tikzpicture}[remember picture]
\node(code) [anatomy] at (0,0) {
\hmtPoint{a}Short line\\
code with some long text\extremPoint{b}[-0.5ex]
};
\fitExtrem{l}{(a) (b)}
\codeAnnotation{n} (-2,0){here is
                          a\extremPoint{point}[0.75ex][0.5ex]
                        long line}
\draw[->, annotation] (point) -- (l);
\end{tikzpicture}
```

yields



Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

A

\anatomy 2
\annotation 3
\annotationcolor 2

B

\bgcmdcolor 2
\bgcode 6
\bgcode 100

C

\code␣annotation 3
\code␣grid␣debug 3
\code␣part 2
\codeAnnotation 6
\codeAnnotation 103
\codeBlock 4
\codeBlock 82
\cPart 4
\cPart 84

D

\dmbPoint 4
\dmbPoint 94

E

\extremPoint 5
\extremPoint 96

F

\fit␣extrem 3
\fitExtrem 5
\fitExtrem 98

H

\hmtPoint 4
\hmtPoint 90

I

\ignored␣code␣part 2
\iPart 4
\iPart 86

M

\mbPoint 4
\mbPoint 92
\mtPoint 4
\mtPoint 88

P

\phspace *6*
\phspace 102
\ptab *6*
\ptab 101

Change History

v0.2-Alpha
General: This package does not load xcolor anymore. It relies on tikz, that tikz loads xcolor in a way that codeanatomy can define RGB color 1
v0.4-Alpha
General: Set fill to annotationcolor explicit for arrow style 3
v0.4-Beta
General: Add new Ti*k*Z Style ignored code part 2
Add option [___codedoc_meta:n style] to cPart 4
Add option [___codedoc_meta:n style] to iPart 4