

## TikZ & PGF & les « node »

Comment placer du texte ou des objets L<sup>A</sup>T<sub>E</sub>X dans tes dessins.

Première partie : le positionnement des « node »

Bertrand Masson

10 février 2010

Le placement du texte, ou de tout autre objet  $\text{\LaTeX}$  (tableau, boîte, image...) dans un dessin  $\text{\TikZ}$  s'effectue à l'aide de la commande : `\draw (x,y) node{texte};`  
Afin de bien comprendre le positionnement du texte les exemples contiendront parfois une croix rouge sur l'origine (coordonnées (0,0)) qui ne sera pas reprise dans la description des commandes. Voici trois exemple de placement de texte :

```
\tikz \draw (0,0) node{A};  
\tikz \draw (0,0) node{\Huge A B C D E};  
\tikz \draw (0,0) node{Un texte quelconque};
```

A

A B C D E

Un texte quelconque

Comme tu peux le voir le texte est centrée sur l'origine. Pour modifier ce comportement `node` accepte des options, qui comme toutes les options  $\text{\LaTeX}$  se notent entre `[]` :

`above` (au dessus), `below` (en dessous), `right` (à droite), `left` (à gauche),  
et leurs combinaisons :

`above left` (en haut à gauche), `above right` (en haut à droite), `below left` (en bas à gauche), `below right` (en bas à droite).

La page suivante en donne des exemples.

```
\draw (0,0) node[above]{A};
```

A  
+

```
\draw (0,0) node[right]{A};
```

+A

```
\draw (0,0) node[below]{A};
```

+

```
\draw (0,0) node[left]{A};
```

A+

```
\draw (0,0) node[above right]{A};
```

A  
+

```
\draw (0,0) node[above left]{A};
```

A  
+

```
\draw (0,0) node[below right]{A};
```

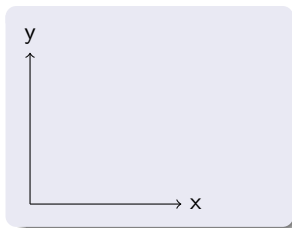
+

```
\draw (0,0) node[bellow left]{A};
```

A  
+

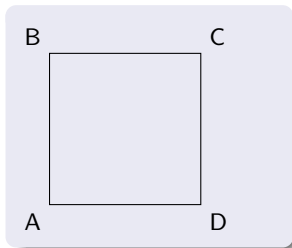
Attention aux espaces dans les noms des options. Contrairement à  $\text{\LaTeX}$ , **TikZ** accepte les espaces pour les noms des options.

Voici par exemple comment noter les axes d'un repère orthonormé :



```
\begin{tikzpicture}
\draw[->] (0,0) -- (2,0) node[right] {x};
\draw[->] (0,0) -- (0,2) node[above] {y};
\end{tikzpicture}
```

et les angles d'un carré



```
\begin{tikzpicture}
\draw[->] (0,0) rectangle (2,2);
\draw(0,0) node[below left] {A};
\draw(0,2) node[above left] {B};
\draw(2,2) node[above right] {C};
\draw(2,0) node[below right] {D};
\end{tikzpicture}
```

Tu peux préciser la distance à laquelle le node sera positionné :

```
\draw (0,0) node[above=5pt]{A};    \draw (0,0) node[right=1cm]}{A};
```

A

+

+

A

```
\draw (0,0) node[below right=10pt]{A};    \draw (0,0) node[bellow left=0.5cm]{A};
```

+

A

+

A

**Attention** `\draw (0,0) node[below=10pt,right=5pt]{A};`, n'aura pas l'effet escompté, car dans ce cas, seul le dernier emplacement précisé est pris en compte, dans notre exemple `right=5pt`.

+

A

Avant de poursuivre avec les « node », il faut comprendre un peu comment ça marche. Un « node » est une boîte rectangulaire qui englobe un contenu. L'option `draw` (`\draw (0,0) node[draw]{A};`) permet de dessiner le contour de la boîte, voici trois exemples :

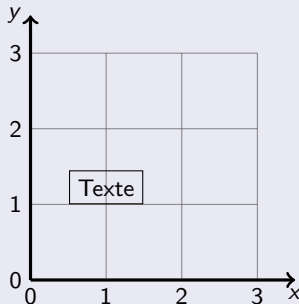
A

Un petit texte



Même un « node » vide prend de la place (le troisième exemple est écrit comme ceci : `\draw (0,0) node[draw]{};`)

Donc les distances sont mesurées par rapport à la boîte et non pas au contenu. Voici le résultat de la commande `\draw (1,0) node[draw,above=1cm]{Texte}`



On vient de voir la position d'un texte par rapport à un point, mais il est possible avec **TikZ** de positionner du texte par rapport à une ligne, avec les options **midway** (au milieu), **near start** (près du début) et **near end** (vers la fin).

```
\tikz \draw (0,0) -- (4,0) node[near start] {A} node[midway] {B}  
      node[near end] {C};
```

Voici le résultat (le B est exactement au centre de la ligne) :

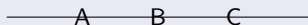


Diagram showing a horizontal line with three nodes: A, B, and C. A is near the start, B is at the midpoint, and C is near the end.

tu peux utiliser les options **above** et **below** pour placer le texte au-dessus ou au-dessous de la ligne :

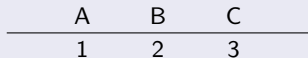


Diagram showing a horizontal line with three nodes: A, B, and C. A is near the start, B is at the midpoint, and C is near the end. The nodes are placed above the line.

```
\tikz \draw (0,0) -- (4,0) node[near start,above] {A}  
      node[midway,above] {B} node[near end,above] {C}  
      node[near start,below] {1} node[midway,below] {2}  
      node[near end,below] {3};
```

## Une petite remarque en passant

Une commande **TikZ** commence par `\tikz` et se termine par un point virgule « ; », ou bien est contenue dans un environnement `tikzpicture` (`\begin{tikzpicture} ... \end{tikzpicture}`). Les passages à la ligne et les espaces entre les parties d'une commande n'ont aucune influence. Donc ne t'en prives pas pour rendre ton code plus clair.

```
\tikz \draw (0,0) -- (4,0) node[near start,above] {A}
                                node[midway,above]      {B}
                                node[near end,above]     {C}
                                node[near start,below] {1}
                                node[midway,below]      {2}
                                node[near end,below]     {3};
```

est plus lisible que

```
\tikz \draw (0,0)--(4,0)node[near start,above]{A}
node[midway,above]{B}node[near end,above]{C}node[near start,below]{1}
node[midway,below]{2}node[near end,below]{3};
```

Les deux écritures donnent le même résultat ! Sur la page précédente, j'ai un peu tassé par manque de place. **Attention** toutefois de ne pas mettre de ligne blanche entre les instructions d'une commande `\tikz`, cela génère une erreur. Par contre aucun problème dans un environnement `tikzpicture`.

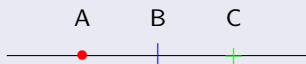


Tu peux mettre des repères sur ta droite :



```
\tikz \draw (0,0)--(4,0) node[near start,above]{A}
      node[red,near start]{$\bullet$}
      node[midway,above]{B}
      node[blue,midway]{|}
      node[near end,above]{C}
      node[green,near end]{+};
```

Tu peux bien entendu tout écrire en coordonnées absolues :



```
\begin{tikzpicture}
\draw (0,0)--(4,0);
\draw (1,0.5) node{A};
\draw (1,0) node[red]{$\bullet$};
\draw (2,0.5) node{B};
\draw (2,0) node[blue]{|};
\draw (3,0.5) node {C};
\draw (3,0) node[green]{+};
\end{tikzpicture}
```

Tu n'es pas limité au trois positions que l'on vient de voir (`midway`, `near start` et `near end`). Il existe également `at end`, `at start`, `very near start` et `very near end`. Tu peux même donner une valeur absolue en employant l'option `pos=valeur` où `valeur` est une fraction comprise entre 0 et 1.

0 étant le début de la ligne, 1 la fin et 0.5 le milieu. Voici les correspondances entre les noms et les fractions :

`midway`  $\Leftrightarrow$  `pos=0.5`

`at start`  $\Leftrightarrow$  `pos=0`

`at end`  $\Leftrightarrow$  `pos=1`

`near start`  $\Leftrightarrow$  `pos=0.25`

`near end`  $\Leftrightarrow$  `pos=0.75`

`very near start`  $\Leftrightarrow$  `pos=0.125`

`very near end`  $\Leftrightarrow$  `pos=0.875`

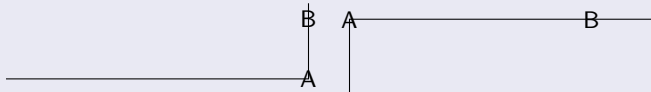
```
\tikz \draw (0,0)--(4,0) node[pos=0,above]{A}
                        node[pos=0.3,above]{B}
                        node[at end,above]{C};
```

A                      B                      C

---

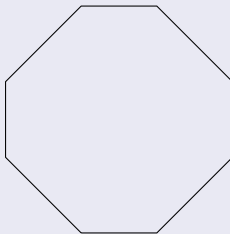
Le cas des tracés perpendiculaires obtenus avec les commandes `|-` et `-|` est particulier. En effet dans ce cas `pos=0.5` ou `midway`, correspondent à l'angle droit. Voici deux exemples ou A est placé à `pos=0.5` et B à `pos=0.9`

```
\tikz \draw (0,0)-|(4,1) node[pos=0.5]{A}  
                                node[pos=0.9]{B};  
\tikz \draw (0,0)|-(4,1) node[pos=0.5]{A}  
                                node[pos=0.9]{B};
```



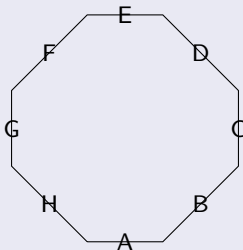
Voici un octogone, obtenu par :

```
\draw (1,0) -- (2,0) -- (3,1) -- (3,2) -- (2,3) -- (1,3) -- (0,2) -- (0,1) -- (1,0);
```



Voici un octogone, obtenu par :

```
\draw (1,0) -- (2,0) -- (3,1) -- (3,2) -- (2,3) -- (1,3) -- (0,2) -- (0,1) -- (1,0);
```

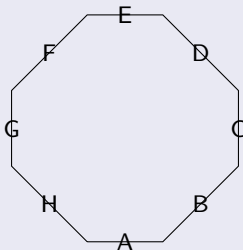


Tu peux numéroté les côtés avec l'option `midway` comme ceci :

```
\draw (1,0) -- (2,0) node[midway]{A} -- (3,1) node[midway]{B} -- (3,2) node[midway]{C}...
```

Voici un octogone, obtenu par :

```
\draw (1,0) -- (2,0) -- (3,1) -- (3,2) -- (2,3) -- (1,3) -- (0,2) -- (0,1) -- (1,0);
```



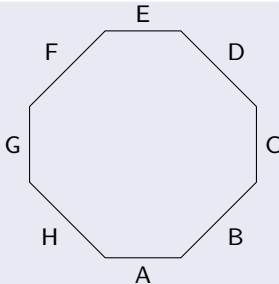
Tu peux numéroté les côtés avec l'option `midway` comme ceci :

```
\draw (1,0) -- (2,0) node[midway]{A} -- (3,1) node[midway]{B} -- (3,2) node[midway]{C}...
```

Par contre si tu ne veux pas que les numéros soit sur la ligne, c'est un petit peu plus compliqué. Il va falloir employer `below` avec A, `above` avec E, `above left` avec D...

Voici un octogone, obtenu par :

```
\draw (1,0) -- (2,0) -- (3,1) -- (3,2) -- (2,3) -- (1,3) -- (0,2) -- (0,1) -- (1,0);
```



Tu peux numéroter les côtés avec l'option `midway` comme ceci :

```
\draw (1,0) -- (2,0) node[midway]{A} -- (3,1) node[midway]{B} -- (3,2) node[midway]{C}...
```

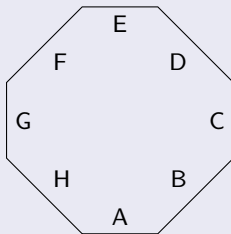
Par contre si tu ne veux pas que les numéros soit sur la ligne, c'est un petit peu plus compliqué. Il va falloir employer `below` avec A, `above` avec E, `above left` avec D... Il y a une solution plus simple `auto=right`

```
\draw[auto=right] (1,0) -- (2,0) node[midway]{A} -- (3,1) node[midway]{B} -- (3,2) node[midway]{C}...
```



Voici un octogone, obtenu par :

```
\draw (1,0) -- (2,0) -- (3,1) -- (3,2) -- (2,3) -- (1,3) -- (0,2) -- (0,1) -- (1,0);
```



Tu peux numéroter les côtés avec l'option `midway` comme ceci :

```
\draw (1,0) -- (2,0) node[midway]{A} -- (3,1) node[midway]{B} -- (3,2) node[midway]{C}...
```

Par contre si tu ne veux pas que les numéros soit sur la ligne, c'est un petit peu plus compliqué. Il va falloir employer `below` avec A, `above` avec E, `above left` avec D...

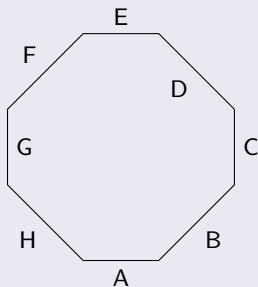
Il y a une solution plus simple `auto=right`

```
\draw[auto=right] (1,0) -- (2,0) node[midway]{A} -- (3,1) node[midway]{B} -- (3,2) node[midway]{C}...
```

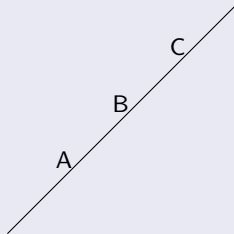
`auto=left` provoque une numérotation à l'intérieur de la figure.

Lors de l'utilisation de `auto=left` ou de `auto=right`, il peut être utile, dans certain cas, qu'une valeur n'ait pas le même comportement que les autres. Pour cela il faut employer `swap` qui inverse localement la valeur de `auto`.

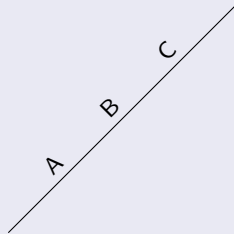
```
\begin{tikzpicture}
\draw[auto=right] (1,0) -- (2,0) node[midway]{A} -- (3,1)
node[midway]{B} -- (3,2) node[midway]{C} -- (2,3)
node[midway,swap]{D} -- (1,3) node[midway]{E}
-- (0,2) node[midway]{F} -- (0,1) node[midway,swap]{G}
-- (1,0) node[midway]{H};
\end{tikzpicture}
```



L'option `sloped` permet d'écrire perpendiculairement au trait, compare les deux exemples ci-dessous :



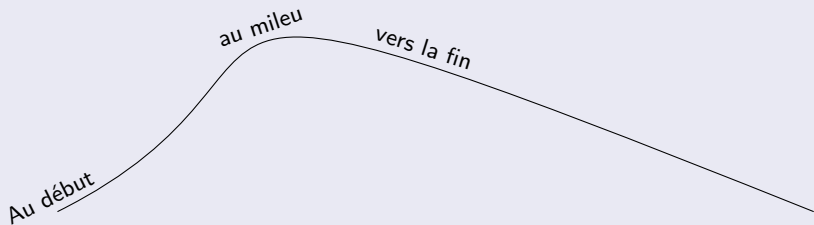
```
\tikz \draw (0,0)--(3,3)
node[near start,above]{A}
node[midway,above]{B}
node[near end,above]{C};
```



```
\tikz \draw (0,0)--(3,3)
node[near start,above,sloped]{A}
node[midway,above,sloped]{B}
node[near end,above,sloped]{C};
```

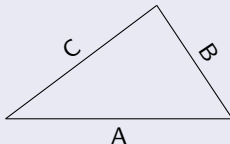
Ça marche aussi avec des lignes courbes.

```
\begin{tikzpicture}  
  \draw(0,1) ..controls (4,3) and (0,5) .. (10,1)  
    node[at start,above,sloped] {Au début}  
    node[midway,above,sloped] {au milieu}  
    node[near end,above,sloped] {vers la fin};  
\end{tikzpicture}
```



Voici une manière de noter les côtés d'un triangle :

```
\begin{tikzpicture}  
  \draw (0,0) -- (3,0) node[midway,below]{A};  
  \draw (3,0) --(2,1.5) node[midway,above,sloped]{B};  
  \draw(2,1.5) --(0,0) node[midway,above,sloped]{C};  
\end{tikzpicture}
```



Quand tu écris `\draw[>] (0,0) -- (5,0) node[draw]{A} -- (8,0);` TikZ va d'abord dessiner le trait de (0,0) à (8,0) puis va placer le « node » à la coordonnée (5,0) dans notre exemple. On obtient :



Ceci n'est pas toujours souhaitable :



Pour obtenir le bon résultat on va faire appelle à deux nouvelles notions concernant les « node » : leur nommage et la commande `at`.

## Une autre façon de placer les « node »

Tu peux placer des « node » à l'aide de la syntaxe suivante :

`\path node at (0,0) {Raymonde}` ou `\node at (0,0) {Raymonde}` puisque `\node` et équivalent à `\path node` (voir la fiche « **Ti**kZ & **P**GF : Introduction »)

## Nommer les « node »

Tu peux donner un nom à un « node » afin d'y faire référence ultérieurement. Pour cela il y a deux façons de faire soit en utilisant l'option `name=leNom` soit en mettant se nom entre parenthèse (des parenthèses pas des accolades) juste après la commande `node` comme ceci :

`\node[name=filles] at (0,0) {Raymonde}`

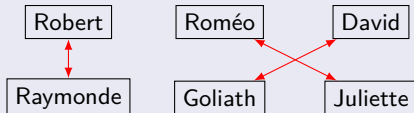
ou bien

`\node (le gars) at (0,0) {Robert}`

Si l'on transforme le code de notre diagramme précédent par :

```
\begin{tikzpicture}
\node[draw] (a1) at (0,1) {Robert};
\node (a2) at (0,0) [draw]{Raymonde};
\node[draw] (b1) at (2,1) {Roméo};
\node[draw] (b2) at (4,0) {Juliette};
\node[draw] (c1) at (4,1) {David};
\node[draw] (c2) at (2,0) {Goliath};
\draw[red,<->,>=latex] (a1) -- (a2);
\draw[red,<->,>=latex] (b1) -- (b2);
\draw[red,<->,>=latex] (c1) -- (c2);
\end{tikzpicture}
```

Dans ce cas on ne trace plus un trait sur lequel on place un node, mais on relie entre eux des « node » par un trait. On obtient :



Ce qui est mieux.



On peut placer encore différemment du texte. Pour cela `TikZ` définit des points d'ancrage. Tu te souviens je t'ai dit qu'un node, c'est avant tout une boîte. L'option `draw` permettait de la matérialiser. Par exemple voici un node contenant le mot « Texte » avec sa boîte :

```
\tikz\draw (0,0) node[draw]{Texte};
```



On a vu également que par défaut `TikZ` prenait comme ancrage le centre de la boîte, c'est à dire que `TikZ` place à la coordonnée précisée (ici (0,0)) le centre de la boîte. Voici le même node toujours positionné à (0,0) avec une croix rouge marquant cette coordonnée (0,0).



`TikZ` définit d'autres points d'ancrage, qui sont décrits dans le schéma page suivante.

Voici un schéma matérialisant un node contenant le mot Texte et sa boite englobante



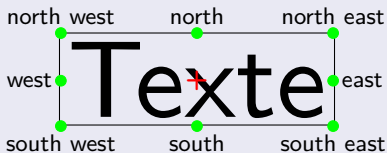
Texte

Voici un schéma matérialisant un node contenant le mot Texte et sa boite englobante



L'ancrage par défaut est matérialisé par la croix rouge

Voici un schéma matérialisant un node contenant le mot Texte et sa boite englobante



L'ancrage par défaut est matérialisé par la croix rouge  
Voici les autres ancrages possible.

Voici quelques exemples toujours avec notre node « Texte », sa boîte et la croix rouge à (0,0).

```
\draw(0,0) node[draw,anchor=north]{Texte};
```



```
\draw(0,0) node[draw,anchor=north east]{Texte};
```



```
\draw(0,0) node[draw,anchor=north west]{Texte};
```



```
\draw(0,0) node[draw,anchor=west]{Texte};
```



```
\draw(0,0) node[draw,anchor=east]{Texte};
```



```
\draw(0,0) node[draw,anchor=south east]{Texte};
```



```
\draw(0,0) node[draw,anchor=south]{Texte};
```



```
\draw(0,0) node[draw,anchor=south west]{Texte};
```

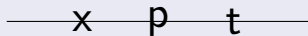


Ces options de placement sont équivalentes à celles que l'on a vue au début de cette fiche : `above`  $\Leftrightarrow$  `anchor=south` ; `below right`  $\Leftrightarrow$  `anchor=north west` ...

La première solution (`above`, `below`...) est beaucoup plus intuitive et c'est personnellement celle que j'utilise. Je t'ai montré cette deuxième façon de faire, car on la trouve souvent dans des exemples de code `TikZ`.

On a vu que par défaut quand tu positionnes des « node » sur une ligne ils sont centrés (`anchor=center`) :

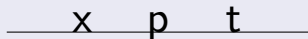
```
\draw (0,0) --(1,0) node{\Huge x}-- (2,0) node{\Huge p}-- (3,0)node{\Huge t}--(4,0) ;
```



A horizontal line segment with three large, black, serif characters 'x', 'p', and 't' centered on it. The characters are of the same size and are evenly spaced along the line.

L'option `anchor=base` permet de placer le texte sur la ligne de base, comme un texte classique, avec les lettres p, q, y... qui débordent sous la ligne de base (profondeur).

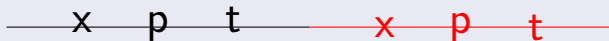
```
\draw[anchor=base] (0,0) --(1,0) node{\Huge x}-- (2,0) node{\Huge p}-- (3,0)node{\Huge t}--(4,0) ;
```



A horizontal line segment with three large, black, serif characters 'x', 'p', and 't' placed on it. The character 'x' is centered on the line. The characters 'p' and 't' are placed on the line but extend downwards below it, as they are not centered vertically.

L'option `anchor=mid` permet de centrer verticalement des « node » ayant des profondeurs différentes. Pour bien comprendre j'ai remis en rouge l'option `anchor=center`

```
\draw[anchor=mid] (0,0) --(1,0) node{\Huge x}-- (2,0) node{\Huge p}-- (3,0)node{\Huge t}--(4,0) ;
```



Two horizontal line segments are shown. The first segment has three large, black, serif characters 'x', 'p', and 't' centered on it. The second segment, which is colored red, also has three large, red, serif characters 'x', 'p', and 't' centered on it.

Il existe également les options `anchor=mid west` et `anchor=mid east`.

Voici respectivement et l'une en dessous de l'autre

`anchor=mid`

`anchor=mid west`

`anchor=mid east`

— x — p — t —

— x — p — t —

— x — p — t —

On vient de voir comment placer du texte sur la ligne de base à l'intérieur d'une figure `TikZ`. Mais comment faire la même chose dans le texte d'un document `LATEX`. En effet quand je place du texte, ici les mots `hello word` à l'aide de `TikZ` voici ce qui se passe

`hello word` . C'est pas terrible. Le `hello word` de `TikZ` est placé dans une boîte et c'est elle qui est placée sur la ligne de base `hello word`. Pour faire les choses correctement on va employer l'option `baseline` et le nommage du « node ». En effet `TikZ` va devoir préalablement évaluer le texte à placer pour voir notamment la présence de lettre comme `p`, `q` ou `y` qui possède une profondeur (une partie sous la ligne de base). Voici la commande qui va nous permettre de placer correctement notre `hello word`.

```
\tikz[baseline=(A.base)] \node (A) {hello word};
```

Voici notre `hello word` dans une phrase. Notre `hello word` est toujours dans une boîte d'où les espaces trop larges avant et après, mais il est sur la ligne de base. Tu me diras, quel est l'intérêt de placer du texte de cette façon ? Eh bien à faire des trucs comme sur la page suivante :



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, **hello** vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, word ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci **hello** risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec **word** ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna.

Voilà c'est la fin de cette fiche. Mais on n'en a pas fini avec le positionnement des « node ». Il existe une librairie **positioning** dont le principal intérêt est de pouvoir faire du positionnement relatif. Mais se sera l'objet d'une autre fiche. Il nous faut voir également ce que l'on met dans nos « node ». Il y a encore du boulot !