

On a un texte qu’on veut inverser et on l’affiche de 3 manières :

* on fait met let code directement dans le template (avec une expression)  
  - **avantage**: je n’en vois pas vraiment  
  - **inconvénient**: ce n’est pas lisible, à chaque update ça se ré-exécute
* on met let code dans une fonction  
  - **avantage** : c’est plus lisible, on peut passer des paramètres  
  - **inconvénient** : à chaque update, ça ré-exécute même si le résultat ne change pas
* on met let code dans un **computed**  
  - **avantage** : c’est aussi lisible qu’une fonction, le résultat sera mis en cache, si une donnée interne a changé, le cache sera mis à jour  
  - **inconvénient** : même s’il y a moins d’exécutions, chaque exécution en interne est plus lourde qu’une **method**car Vue fait la mise en cache

[Bref, comprendre le computed en profondeur dans Vue JS & ses différences avec watch & methods | by @maisonfutari | Medium](https://medium.com/@arieldi/bref-comprendre-le-computed-en-profondeur-dans-vue-js-ses-diff%C3%A9rences-avec-watch-methods-62635c2bb394)

* **Watch**  
  - réagi à une **data** ou une **prop** qui change pour effectuer une action  
  - c’est comme **écouter un événement**“si la **X data** change”  
  - prends en paramètre **la nouvelle data et l’ancienne**- ne fait **pas de return** (il ne fait que réagir à un changement de données)  
  - n’est pas utilisable dans le template
* **Computed**  
  - comme une **method**(return une valeur) mais ne prend pas de paramètre  
  - fais du cache grâce à une **data** ou **prop** détectée avec **this.taData**  
  - utilisable dans le template et **ne sera pas toujours ré-exécuté**
* **Methods**  
  - une fonction utilisable dans le template, qui sera **toujours ré-exécutée**- doit faire **un return pour afficher** une donnée dans le template