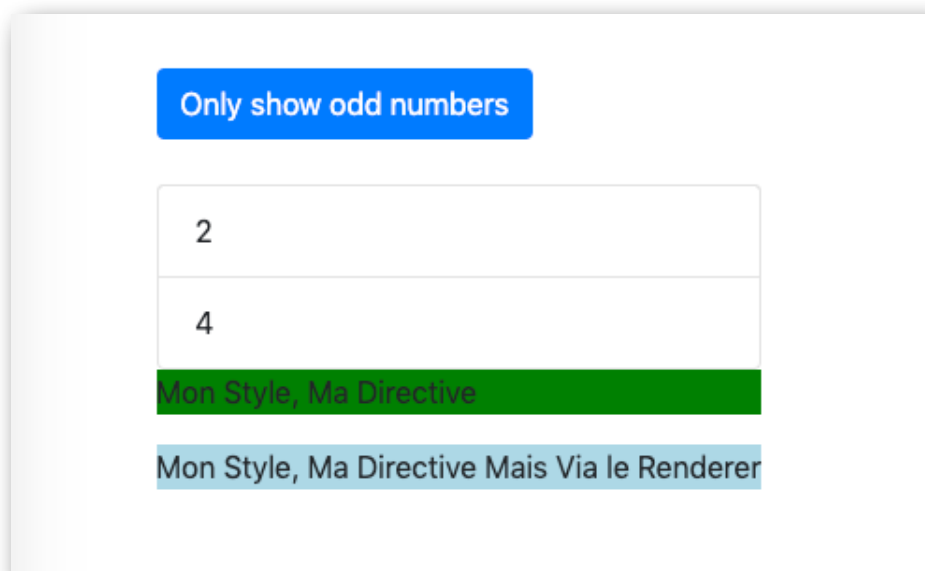
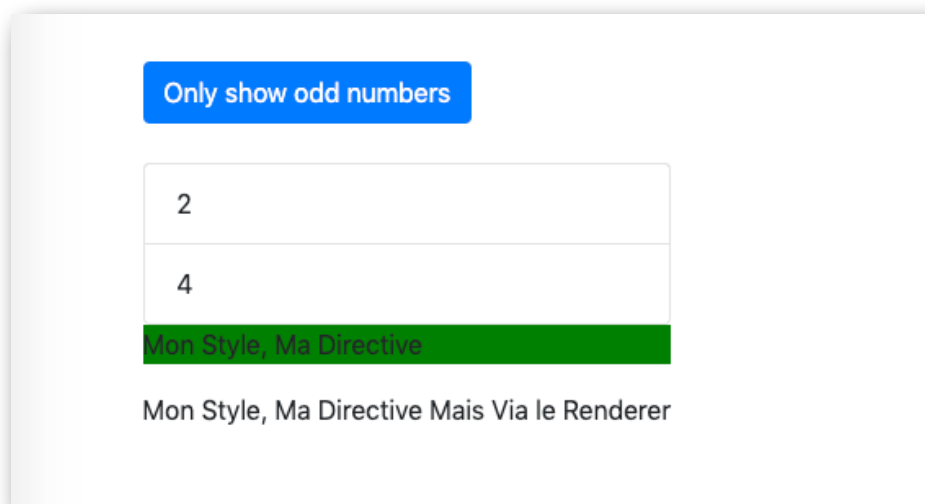


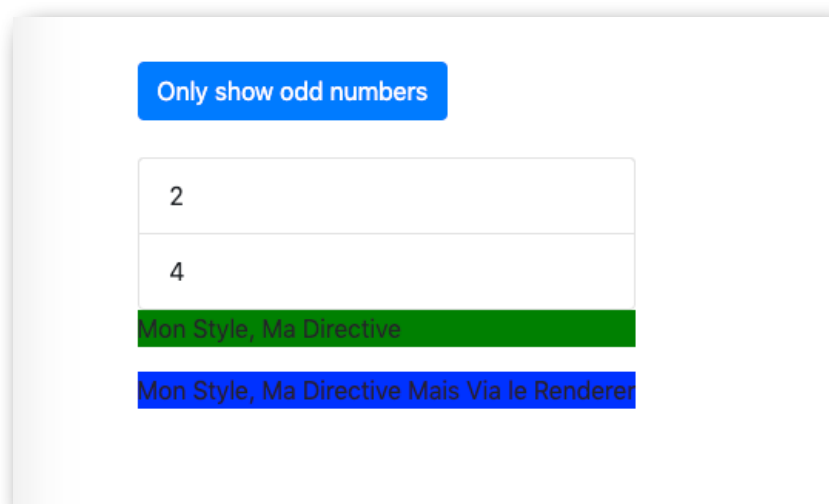
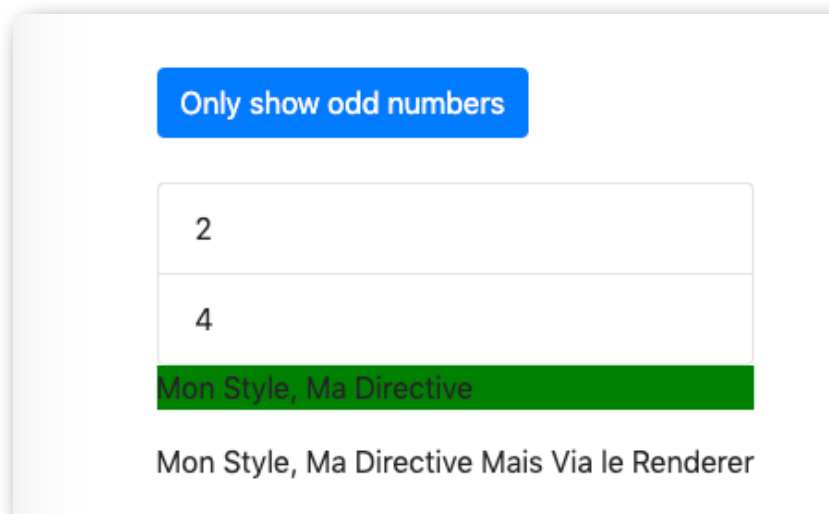
## FDR - Angular

### Application nombres pairs / impairs

- Se renseigner sur `@HostListener()`, `mouseover()`, `mouseleave()`
- On va modifier /améliorer notre directive `betterSurlign` :
  - Quand l'utilisateur passe la souris sur le `<p>` sur lequel on a placé la directive, alors le Background est bleu
  - Quand l'utilisateur enlève la souris du `<p>` sur lequel on a placé la directive, alors le Background est transparent
- (On utilise toujours le `render2`)



- Se renseigner sur @HostBinding()
- On va modifier /améliorer notre directive betterSurlign :
  - Même système que l'exo précédent mais cette fois ci on va binder le backgroundColor avec une variable (string) bgColor
  - Comme ça dans les HostListener :
    - Quand la souris passe sur le <p> on modifie juste la variable bgColor
    - Quand la souris sort du <p> on modifie juste la variable bgColor



- Se renseigner sur ngSwitch, ngSwitchCase, ngSwitchDefault
- On va modifier /améliorer appComponent :
  - Côté TS on rajoute une variable « value », initialisée à 10
  - Côté HTML on va rajouter une div qui contient 4 paragraphes
    - 1 <p> qui dit « la valeur est de 5
    - 1 <p> qui dit « la valeur est de 10
    - 1 <p> qui dit « la valeur est de 15
    - 1 <p> qui dit « valeur par défaut »
  - Sur ces paragraphes on va utiliser des ngSwitch et sur le dernier ngSwitchDefault
- Du coup en fonction de la valeur qu'on a attribué à la variable value (Côté TS) cela affichera le bon paragraphe

