**VUE.JS 3**

|  |  |  |
| --- | --- | --- |
| Créer un objet Vue  Avec **template**  Créer un objet Vue  Avec **data()**  Monte sur élément |  | Créer un élément vu avec un **template** dans lequel on peut mettre du code html.  Utiliser **`** pour écrire en multi ligne  Monte notre objet sur l’élément avec l’id app  Ici on définit la méthode **data** dans lequel on stock … nos data qu’ont appel directement dans le html avec **{{ }}**    On peut aussi stocker le « **mount** » dans un attribut, ce qui permettra de le modifier  Exemple : ici on modifie la propriété prenom de l’objet personne |
| Affichage des variables |  | On peut afficher nos variable définie en JS dans notre code HTML avec **{{ }}** et utiliser le JS dessus  Ternaire possible |
| Propriété calculer  **conputed** |  | Si on a des calculs à faire sur des propriétés définie dans data on utilisera **conputed** qui renverra ce calcule    Ici un calcul est fait sur une propriété time (qui renvois un **new Date()** ) |
| v-bind |  | **v-bind:** permet de remplir le contenu de la balise src par l’attribut src définie dans le composant vue (dans data() ).  **:** est un raccourci pour **v-bind:** |
| v-text  v-html |  | La propriété texteBrut sera insérer en tant que contenu  Idem mais le HTML sera interprété contrairement à **v-text** |
| v-model |  | Permet de récupérer des données (seulement pour input) |
| v-on et @  (Evenement) |  | Réagie à un évènement de type frappe clavier  Idem avec un click. **@** est une propriété raccourcie de **v-on**  On peut passer **$event** en paramètre pour passer l’évènement |
| Gestion des évènements  **Methods**  **Annuler comportement par défaut** |  | **méthods** est une propriété de notre objet Vue dans lequel on vient mettre nos méthode lié au évènements par exemple  On peut accéder aux propriétés d’un évènement (coordonnées, touche taper etc.) en donnant en paramètre dans **un v-on :** ou **@** l’attribut **$event**  Permet d’annuler le comportement par defaut d’un evenement (ex : click sur submit etc) |
| Modificateur évènement |  | Ici on ajoute un **preventDefaut** + **stopPropagation**  Ici on réagie sur un évènement de type click gauche et droite pour lesquels on appelle une méthode largeur()  Evènement sur la touche espace (il en existe pleins d’autre) |
| v-for |  | Permet d’itéré sur chaque élément d’une liste, par valeur ou par valeur et index |
| Argument dynamique [] |  | Permet de binder un évènement dynamique définie par l’attribut **evenement** dans **data()** |
| v-cloak |  | Combined with CSS rules such as [v-cloak] { display: none }, this directive can be used to hide un-compiled mustache bindings until the component instance is ready. |
| v-once |  | v-once fixe les valeur entre {{ }} meme si elle sont modifier, ici si nombre est incrémenter ce ne sera visible que dans la 2e div |
| Classe conditionnelles |  | Affectera la classe rouge SI lan.etatCase vaut true |
| Style inline |  | Affectera la variable couleur au style color etc… |
| v-if  v-else-if / v-else |  | SI kangage.etat vaut true alors on affichera le texte est coché  SINON on affichera est décoché |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |