

November 2016

Wednesday

نوفمبر

الأربعاء

١٦

16

١٦ صفر
٢٠١٦
٧ هاتور

دعا

Part 1 { Basics }

١٤٣٦ هـ صفر ١٥

November

2016

٢٠١٦

نوفمبر
الثلاثاء

١٥
١٥

١٧٧٧ هـ ماتور

Tuesday

Vue.js - V2 (Basics)

- 1- Vue.js basics
- 2- work with vue cdn
- 3- Directives
- 4- computed property
- 5- watch property
- 6- Vue components
- 7- Refs
- 8- Lifecycle hooks
- 9- form

Name

Tel.

Email

نوفمبر ٢٠١٦

نوفمبر ٢٠١٦ November 2016

Monday

الاثنين
١٤٣١ هـ
٥ هاتور

نوفمبر

November

٢٠١٦

١٤

الاثنين

١٤
14

① Vue.js basics

أين هو ال دلالة؟

SPA بناءً على ما ذكرنا سابقاً

ما هي library و framework؟

إذن لأن ال Vue بسيط في إنشاء complete solutions اقتصر على building complete web application من غير ما احتاج

إنه على سبيل المثال react، لا يحتاج بعدها library . 3rd party library

يمكن إنشاء application بالтайك هو مجرد library .

(الخاصة) ال complete solution يوضح framework إنما ال library يحتاج تنزيل libraries ثانية خارجية معاً.

أين هو ال SPA؟

تصفح بROWSER عن صفحة html واصحة بجيش كل request يتوجه خفلي ، والصفحة رئيسية تحتوي على كل ال content يتوجه المتصفح وال end-user content بينها لا تنتقل من ال browser إلى server حتى لا يحصل أي bad header للprogram .

يتنتقل من ال browser إلى ملف من خلال ملف أو ال دلالة bundle .

سواء بتوجه ال Vue أو react أو angular

ال SPA مبنية على browser reload لأنها ببساطة run في ال browser وليس السيرفر بالтайك هو مسارها من ال MPA . يعني كل ما أعمل حاجة لدارم اروح للسيرفر ، كل حاجة من المتوجه

نوفمبر ٢٠١٦

November

Sunday

١٣ صفر ١٤٣٨

١٣ هاتور ١٤٣٣

الأحد

١٣

١) أية الفرق بين ال SPA و ال MPA

في ال MPA كل صفحة html يكون لها url ، يعني كل صفحة تدعى request من المتصفح وبعدها request للسيرفر وأما عن ال SPA response يرجع بمصفحة ال html التي انا عايزها.

اما عن ال SPA هو مصفحة html واصحه هي كل صفحه في المشروع ، يعني يمكن تخيل ان احنا نضغط على MPA كل صفحة عاصمه اي file html واحد.

٢) أية الفرق بين ال Vue و ال Angular و ال React

١) في ال React يكتب كود html منفصل عن الكود js / jsx.

٢) في ال Angular يكتب كود ال html منفصل عن الكود ts .
يعني ال Angular يقسم ال component ل ٣ اجزاء مت分明ين :
ملف html فيه ال UI ، ملف ts فيه ال logic ، ملف css.

٣) في ال Vue يكتب كود ال html منفصل عن الكود js .
ال Angular و Vue الفرق ان ال Vue يجعل ال component منفصل اجزاء ال component .
عن طريق واصه باسمدار (vue) و يفصل اجزاء ال component .
من vue و vue و vue و vue .

٤) العامل المستمر بينهم انهم يتعلموا SPA

Name

Tel.

Email

نوفمبر ٢٠١٦
٢٠١٦ ١٢ ١١ ٢٣

٢) أي نستخدم Vue / React / Angular

الـ SPA هي مفهوم الوظيفة الذي هو العامل الفاصل ما بينهم هو تفاصيل التصفح + صيغة + صيغة.

لو أتيحت المترددة كبيرة جداً يفضل نستخدم Angular لـ `Lazy loading` أكثر وأتمتة `structure` بمحاجنا نكتب كود تفاصيل سهل التديل عليه.

لواجهة المستخدم غير متوسط يفضل نستخدم Vue أو React.

٣) أي من مميزات الـ Vue

أهم ميزة من الـ Vue هي راحيتها في التعلم في أي مشروع حيث تقدر بخفق بيئة موافقها كامل أو ينبع `Widget` / صيغة من الموقفي من غير ما نذهب بناء الموقفي من أول وجهة (اد Vue سهل التعلم في أي مشروع)

ذلك يعكس الـ angular أو الـ react ، المترددة من أدوات لأهمية لازم تعمل بهم بالكامل.

الـ Vue يبحث Virtual DOM في الـ React او ورقة `performance`.

الـ Vue بنية بسيطة الـ app يمكن `declaratively` يعني `logics` يمكن بتحلل للـ Vue عاليه امثل اية وهو يستخدم الـ `HTML` من غير ما أخذه من حفاظه.

الـ Vue مخفيف جداً.

الـ Vue هو `reusable UI component-based` يمكن تكرار تحلل

٢٠١٦

نوفمبر

November

Friday

١١ صفر ١٤٣٨

٢ هاتور ١٩٧٣

١١
11

الجمعة

١) اشرح فكره vue DOM

ال vue هي اضافة من خارج DOM الحقيقي ويرتبط
نحوه وصيغة منه، اي dom manipulation يحمل التغيير
يسعى الى dom الوهم من الحقيقي، مثلاً كده ال vue
هي تقارن التغيير ما بين dom الوهم وال الحقيقي ويفاد
يغير الاصدقاء الـ `innerHTML` وتغيير فقط من غير ما يفعل
reload لكي يكون dom الحقيقي.

٢) يحيى ابيه ال vue هوdeclarative

يعنى declarative بقوله ما يكتب `log` معن
وهو صيغة من غير ما يكتب من تفاصيل او `log`.

(مثال) ① يستخدم interpolation `{{ }}` علامة

تطبقي على `variable` من ال vue للـ `html`.

وكل ده من غير ما يكتب اي `log` بنسخنا.

② نعمصر `repeat` و`bind` و`loop` ونعمل `bind` من كود ال `html` يستعمل declarative يعني
وهي من غير ما يكتب اي `log` بنفسه.

declarative يعني وصيغ / بوصفت الـ vue
عايز اعمل ايه، وار vue صيغة الـ `if` عايز من
غير ما يدخلني من تفاصيل يعني عيادة `log` مش
مكتتب اي `log` بمنظري.

Name

Tel.

Email

نوفمبر ٢٠١٦

Q) ازای نعل vue install لر
ازای نعل ستوده vue

۱۰ الـ Vue بیوخر چیزیست ؟

۱۱ نتنهل بـ cdn تو عایزن نعمل widget
یعنی حصه من المتروده وليس المتروده کامل.

۱۲ سختم الـ vue چه ابزار
تبجز الـ Vue بتوصلنا شویه ادامر بتسلیمها بین
مشروعی چال باز Vue

Q) آیه هو ال directives چه اد میزند ؟

۱۳ عباره من دیگره spetial html attributes تبجز الـ Vue
بنعل بیوخر element dom manipulation معنی ملعون
الک بنخواهد فریه

(v-.....) vuc directive كل
v-for , v-if (ex,

Q) آیه هو ال interpolation چه اد میزند ؟

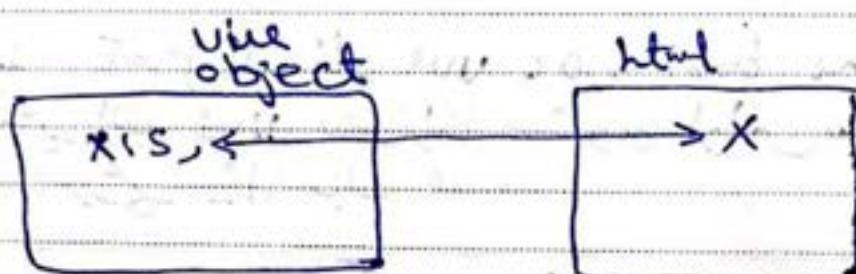
۱۴ dynamic data چه اد میزند
من الـ Vue نحطلا من الـ View/template او سمعه وده من
خلال اث اصنا بناحاط / بنعل الـ data property أو الـ expression
ما بين {{ }}

لا syntax mustache بسمه



Q) يعني اية ال vue هو reactive

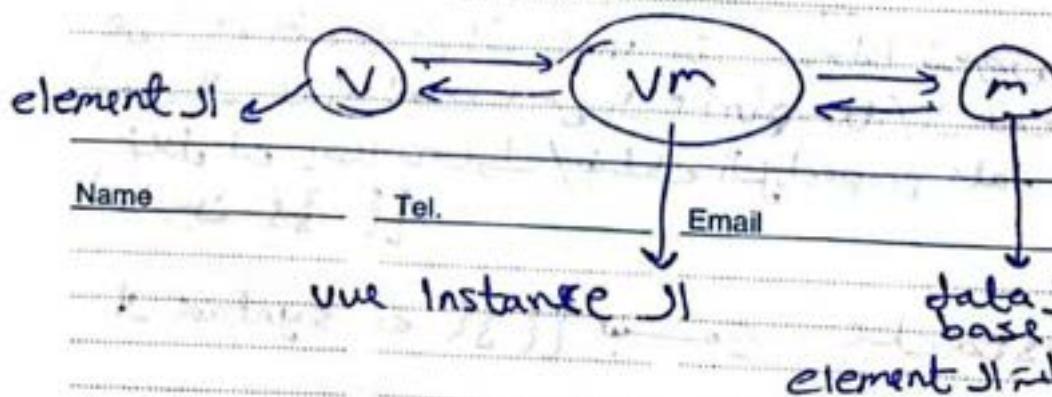
يعني reactive يعني يحس بالتحت من ال data underlying data يعني لو حصل تغيير من ال data تحس بالتحت ويرجح بعمل re-render وبعرضة من مكانة من ال view



Q) ايه هو ال architecture pattern اللى مبني على vue

MVVM

- ال view هو ال view اللى صو ال UI بيتامنا.
- ال model هو ال model اللى صو ال data بيتامنا واللوجيك.
- ال viewmodel هو ال viewmodel اللى برتبط ما بين ال view instance وUI ، ال viewmodel خسر ال connection database .



نوفمبر ٢٠١٦

٢٣	٢٤	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠
٢٤	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠	١
٢٥	٢٦	٢٧	٢٨	٢٩	٣٠	١	٢
٢٦	٢٧	٢٨	٢٩	٣٠	١	٢	٣
٢٧	٢٨	٢٩	٣٠	١	٢	٣	٤
٢٨	٢٩	٣٠	١	٢	٣	٤	٥
٢٩	٣٠	١	٢	٣	٤	٥	٦
٣٠	١	٢	٣	٤	٥	٦	٧

اد vue م two way binding يعني بتومر MVVM

② work with vue cdn

```
<body>
  <div id='app'>
    {{name}}
  </div>
```

`<script src="—cdn of vue"></script>`

`<script>`

vue instance (view model)

(vm)

```
var vm = new Vue({
  el: '#app',
  data: {
    name: 'Ali',
  }
});
```

`</script>`

١) vue instance حوالاً

application trigger هو vue object يمثل بيكور مسؤول عن التحفيز وإدارة الـ data، والـ logic.

widget معين / جزء معين من application

dynamic UI بـ vue object كل عملية انتقاء

element يمثل vue object معين.

element vue object كل لدزيم ناحية الـ data كل صرکب عليه / الـ trigger عملية / الـ logic يتغير عليه.

vue object كل يحتوى على الـ data والـ logic بتوجه معين.



<body>

```
<div id="app1"></div>
```

```
<div id="app2"></div>
```

```
<script src="cdn.js"></script>
```

```
<script>
```

```
view model <new Vue({})> instance  

view <el: "#app1", .element> مع ايه اجزء يتركب من  

model <business logic  

  data: {}, dynamic data  

  methods: {}متغيرات الـ element  

}; بتاجر الـ element
```

```
new Vue({
```

```
  el: "#app1", }]
```

```
  data: {}, }]
```

```
  methods: {} }]
```

```
});
```

properties الـ

options الـ

instance باتحة الـ

configuration الـ

Name _____

Tel. _____

Email _____

fundamental unit. والـ instance

عندما ندخل vue app لـ vue instance
لو مدخلنا أكثر يعنى معه كثر
vue application

نوفمبر ٢٠١٦

السبت	٥
الاحد	٦
الاثنين	٧
الثلاثاء	٨
الأربعاء	٩
الخميس	١٠
الجمعة	١١

(يتذكر) أن vue instance هو الوجهة الأساسية لبيانات أي application object هي مسيرة عن entry point من الـ class ١، يمكن أن تكون مسيرة عن الـ features كل الـ vue framework والـ functionality.

(للاصط) vue interpolation نذكر نكتب فحية ١

- ١ bind ذي vue's data من الـ element
- ٢ نعمل method لـ invoke من الـ vue's methods
- ٣ نعمل expression ذي زر على حسابية مثلثة

(للاصط) كل vue instance نريد فحية الـ element ذي vue's data من الـ trigger mount يعني الـ vue's methods يبيس الـ vue's functionality features كل الـ vue instance بتاعة الـ vue عاليحة.

(للاصط) معاي نعمل ذي vue widget ذي vue لللزم نجرب ٣ طرق سهلاً من صنعنا الـ html:

- ١ جزء الـ component / div element
- ٢ واسطة ذي معاي بمحرك
- ٣ جزء الـ cdn أو نزل ملف الـ zip من vue instance

- ١) component
- ٢) cdn
- ٣) vue instance



③ Vue Directives

أيّه هو ال directive

هي عبارة عن علامات مخصوصة / markers ، العلامات توضع في dom element لـ dom element special behavior

ال directives تكون في شكل html attribute و كل حرف يبدأ بحرف (v-)

- behaviors يُضفّع directive على dom element
- condition يظهر أو خفي على element
- loop يكرر على element
- binding يربط vue's data مع element
- inject يأخذ html من vue

ال declarative directive هي الأك بخاكي أن vue تبعي منطقless من غير ماس تغلى باكي بالتفاصيل .

v-xxx = " " لـ js expression

Name _____

Tel. _____

Email _____

نوفمبر ٢٠١٦

السبت	الأحد	الاثنين
٢٣	٢٤	٢٥
٢٤	٢٥	٢٦

inject بـخليانات directive دیه، v-text
element یز text

```
en data: {
    message: "I will be injected into h1",
}
<h1 v-text="message" ></h1>
```

کور inject بـخليانات directive دیه، v-html
element یز html

```
en data: {
    blf: "<p>Hello</p>",
}
<div v-html="blf" ></div>
<div v-html="is-x ? blf : 'hi'" ></div>
```

directive یا value یا

- حاری js expression یو

```
<div v-html="is-x ? <p>-</p> : <div>-</div>" ></div>
```

data: { : v-for
 cars: [{x:'blf', y:15}, {}, {}],
}

```
<li v-for="car in cars" > {{ car.x }} </li>
<li v-for="(car, index) in cars" >
    {{ index }} ==> {{ car.x }}
```

```
</li>
<div v-for="(car, index) in cars :key="index" >
```



looping في الجمل directive دا : v-for
عند object أو array على
نعم element في بعض

data :

```
arr: [   ],
obj: { x: 1,
        y: 3
      },
      ↓ key
```

جاء
Just Rendering

```
<p v-for="(car, index) in arr"></p>
<p v-for="(o, key, index) in obj"></p>
```

: v-if , v-else-if , v-else
دول تأتي من directive
condition (عندما معين بـ) element خارج

```
<p v-if="s > 3"></p>
<p v-else-if=">"></p>
<p v-else></p>
```

يتحقق
conditional
Rendering

override دا الجمل directive دا : v-once
element دا view
(ex) <p v-once>{{x}}</p>
view دا update دا

Name

Tel.

Email

الـ v-once يطبق على الـ element
وامض ويعبر كـ يتحقق

نوفمبر ٢٠١٦

٢٣	١٥	١٧	٥	٥
٢٤	١٦	١٨	٦	٦
٢٥	١٧	١٩	٧	٧
٢٦	١٨	٢٠	٨	٨
٢٧	١٩	٢١	٩	٩

١) directive `v-show` لـ ظهور و إخفاء
الـ condition element معين.

`<P v-show="5>3"></P>`

٢) آية الفرق بين الـ `v-if` والـ `v-show`

الـ `v-if` يستخدم عادةً لـ ظهور وإخفاء
الـ element من الـ dom tree على معيار شرط معين.

الـ `v-show` يـ `re-render` element
من الـ dom tree.

إذا `v-show` يـ `re-render` element
و يـ `update` من الـ dom tree.

لو الـ element مستعملة هـ `re-render` من خـ `dom tree`
ويختلف أكثر من مرة بـ `data` ، يـ `re-render` element
الـ `v-show`. لـ `v-show` سـ `re-render` element
لـ `dom tree` كلـ `change` في الـ `data`.

٣) directive `v-bind`: كـ `bind` يعني ربط
underlying data مـ `html attributes` و بين
الـ `vue instance` و بين الـ `html attributes` .

(ex) `data:`

`url: 'http--'`,

}

`<a v-bind:href="url">`

`<a :href="url">`

↳ shorthand
`v-bind:`



١ صفر
١٤٣٨ هـ
٢٢ بابه

November

٢٠١٦

نوفمبر

Tuesday

الثلاثاء

event handling مثلاً نعمل على : V-ON ⑨

v-on:click = "m1()", m2()"

shorthand
v-on

event

handlers

@click = "m1()"

وقد نعمل على invoke للكثير من event handlers

v-on:click.right.prevent = "m1()"

directive event modifier(s) handler

لأن directive v-on من v-on يطلب Listen events والباقي events ونقدر نضيف
customizes v-on مع event modifier
behavior على event

<P @click.right.prevent

@copy.prevent

@paste.prevent > ... </P>

نمنع user من copy أيهه

Name

Tel.

Email

نوفمبر ٢٠١٦

الأحد	٥
الاثنين	٦
الثلاثاء	٧
الاربعاء	٨
الخميس	٩

اکتوبر ۲۰۱۶ ۳۱ October 2016

الاثنين Monday

الاثنين

٣١
31

v-on:click = "m1(\$event)"

@:click = "m1(\$event)"

الآن حمل سار event object اسے \$event
الآن بجھوں میں بیانا سے الگ

اڑاں نے جل حوالہ binding ①

→ :style = "{color: , font-size: }"

① object syntax

② multiple value

:style = "{border: ['1px', 'solid', 'red']}"

③ multiple objects

:style = "[{}, {}, {}]"

④ :class = "{ 'color': s>3, 'm-4': s<6 }"

⑤ class = "row" :class = "{}"

⑥ :class = "[{}, {}, ...]"

⑦ :class = "[s>3 ? { 'row': s<4 } : {}]"

:class = "[s>3 ? { 'row': 'flex', 'm-4': '' }, ...]"

:class = "['m-4', x==12? ' ', ' ', ' ', ...]"

October

2016 ٢٠١٦ أكتوبر

٢٩ محرم ١٤٣٨

بابه ٢٠

Sunday

٢٠١٦

30

two-way binding بirectional directive : v-model
 . we instance input element وال ما بين

value ل bind على ن بتاتعة ال <input> مع ال data

```
(ex) data: {
  name: ''
}
<input v-model="name" />
<p>{{name}}</p>
```

form input ل bind ما بين / يربط v-model
 . component's data وال

html attribute ل bind ما بين v-bind .
 . component's data وال

component's data ل {{ }} بعرض ال interpolation
 . component ل

Name _____

Tel. _____

Email _____

أكتوبر ٢٠١٦

٢٩	٢٧	٢٥	٢٤	٢٣	٢٢	٢١
٢٠	٢٢	١٩	٢٠	٢١	٢٢	٢٣
٢٣	٢١	١٧	١٨	١٩	٢٠	٢١

من empty attribute و v-cloak
mounting element ال يحصل على element
لكن هى مغيرة بشكل
كامل.

لأن فايرتك فتر نبذة من ال CSS عنا
يتحقق ال component مع عناصر
app في ال mount

```
<style>
```

```
[v-cloak]{}
```

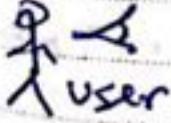
```
display:none;
```

```
}
```

```
</style>
```

```
<div v-cloak>{{ message }}</div>
```

يحل محله (v-cloak) بالـ (flash of uncompiled templates)
وهو لـ raw data user يستوفى الـ
وهو قبل ما يتم compliling
وهو لـ vue لـ component
وقتها عناصر ما فيـ mount





٢٧ محرم ١٤٣٨

١٨ بابه ١٧٧٢

October

Friday

2016 أكتوبر ٢٠١٦

جمعة

④ computed property

اسطرد ايه هو ال computed properties

هي عبارة عن `function` تكتب في `data` كـ `getter` و `setter`
على `Vue instance` ثانية موجودة من ال `data` `caching` (يُرجع لها بـ `getter`)
ال `data` ايجديك ديره يرجع لها بـ `setter` `dependencies` (غيرها).
يعني من متغير إلى تلك `dependencies` يتغير.

نستخدم ال `computed property` لـ `副作用` نعمل
عندية صافية `data` قبل ما نعرض على ال UI

اـ ال `data property` هو `computed property` ال
ـ `data properties` (عنوانها نعمل `getter` بـ `value` `builder` من `data` قدرة)
ـ `data` (يمكن نكتب `data` بـ `getter` و `setter`)

ـ ال `recompute` (يمكن نكتب `recompute` الـ `getter` و `setter`).
ـ `update` (يمكن نكتب `update` الـ `getter` و `setter`).
ـ الـ `dependencies` (يمكن نكتب `dependencies` الـ `getter` و `setter`).
ـ `caching` (يمكن نكتب `caching` الـ `getter` و `setter`).
ـ `(recompute)` (يمكن نكتب `(recompute)` الـ `getter` و `setter`).

ـ الـ `method` (يمكن نكتب `method` الـ `getter` و `setter`).
ـ الـ `recompute` (يمكن نكتب `recompute` الـ `getter` و `setter`).
ـ الـ `update` (يمكن نكتب `update` الـ `getter` و `setter`).
ـ الـ `caching` (يمكن نكتب `caching` الـ `getter` و `setter`).
ـ الـ `(recompute)` (يمكن نكتب `(recompute)` الـ `getter` و `setter`).

Name _____

Tel. _____

Email _____

أكتوبر ٢٠١٦

٢٠١٦ أكتوبر

```
(ex) new Vue({
  el: "#app",
  data: {
    fname: 'Hera',
    lname: 'Anir'
  },
  computed: {
    fullname() {
      return this.fname + ' ' + this.lname;
    }
  }
});
```

نبيل حسني مبارك من طلبة
جامعة طنطا

```
<div>
  <p>{{ fullname }}</p>
</div>
```

لو حصلت على change في UI لـ rerender
فسيتم recalculate caching لـ recompute
نفس المثودات العاديـة.

(الخلاصة)

ـ data function هي computed property
ـ computed property هي ثانية ، الـ data property
ـ يستعمل Vue في caching فاكـر
ـ يستعمل caching في recompute / reinvokـe / recall
ـ إلا لو الـ data الـ data يـغـير اـنتـغيرـاتـ

الفرق بين الـ method والـ computed إـنـه يـستـغـلـهـ

ـ الـ Vue

٢٦
26

١- اخراج ال binding (Q)

- ١- one way binding ①
- ٢- two way binding ②

صوّاب اصنا ندخل حامٍ ما بين binding الـ " " .

الـ one way هو اصنا ندخل الـ data من الـ component's view لـ component's data .

الـ two way هو اصنا ندخل الـ data من الـ component's view وـ component's data .

form-input elements مع الـ v-model الـ two-way binding .

```
<input type="number" v-model="count" />
<p>{{ count }}</p>
```

```
data: {
  count: 0,
}
```

v-model.lazy = "count"
 directive modifier
 binds to the data .

Name

Tel.

Email

اكتوبر ٢٠١٦

٩) إيه حوال directive modifiers

دَسْ عباري عن keywords يتضيّف أو يتغير آل directive.

١٠) إيه أهمية ال v-once

ابتغل `element render` واصره `v-once` تأكّي `update` بجهة `freeze`.

١١) إيه الفرق بين ال `computed` وال `method`

١٢) إيه الفرق بين `call` كل مانعاد `method` وكل `view rerender` من ال `view`.

١٣) إيه الفرق بين `computed` و `method` في `data` (dependencias) `view` (caching) `data` (first render) `data` (second render).

١٤) إيه الفرق بين `methods` و `computed` لكن منه يُستعمل `invoke` إلا ما يُحتمل مع `update` على `data` الـ `methods` يستخدمون إلـ `data`.

٢٤
24

٢٢ محرم ١٤٢٨
١٤ باب ١٣٢

October

Monday

٢٠١٦ ٢٠١٦
أكتوبر

اثنين

(٩) لست بمعنى الـ computed property بـ computed method ويسير

لدت الـ computed property بـ كلها نفس شكل الـ method لكن اسخدم امر زى اسخنام الـ property ، ازاي ستحتى الـ property كـ يعن من بنعمل call لكن بنعمل computing

ex computed: {
 method:
 m1(){
 return 'hi';
 }
 },

 } } m1() X
 } } m1() L

(١٠) ازاي نعمل computed property

• method كل طريقة بـ ① ملـ object أو ② ملـ object جىتوه . getter > setter مع

الفرق إنـ الـ getter والـ setter ناخدـ value منـ الـ view ونعمل validate قبل ما نـ setـ الـ property وـ value

Name

Tel.

Email

أكتوبر ٢٠١٦

٢٩	٢٧	٢٥	٢٤	A	١
٢٨	٢٦	٢٤	٢٣	B	٢
٢٩	٢٧	٢٥	٢٤	C	٣
٢٩	٢٨	٢٦	٢٤	D	٤
٢٩	٢٨	٢٧	٢٤	E	٥

```

(ex) computer: {
    getMessage() {
        },
        getMessage2: {
            get() {
                return this.message.toUpperCase();
            },
            set(value) {
                // validation
                this.message = value;
            }
        }
    }
}

```

(ماحوظة) الـ computed set تغيره قبل من يدخل value والـ set (value) للذرة.

```
set(newValue, oldValue); }
```

(ماحوظ) وينظر vue data () track كل الم diretives للذرة data درجة تغير من الـ data

```

new Vue({
    el: '#app',
    data: {
        count: 0
    }
}); 
```

vue الـ data الـ tracking مستقر تغادر

٢٠١٦ أكتوبر 2016

October

Saturday

٢١ محرم ١٤٣٨

١٢ بابي ١٧٣٣

22

⑤ watch property -

الخاصية `watch` هي إحدى `features` في `Vue` التي تتيحنا
مراقبة التغييرات على الأك بـ `data` property من حيث
أو `computed` property معين بعد كل تغيير في `logic` معين
كل ما يحصل بالتغيير من الأك بـ `data` property.

- الـ `methods` يحتوى على `watch property`.
- كل `method` يراقب `data property` معين.
- لازم أن `method` يكون نفس حجم للتغير الذي يراقبه.
- `parameter` `watch method by default` عبارة عن القيمة بتاته أو `data` الذي يراقب.

(ex)

```
data: {
  message: 'Hello',
},
watch: {
  message(newValue) {
    console.log(newValue);
  }
},
```

Name

Tel.

Email

أكتوبر ٢٠١٦

⑥ Components :

الـ component هو جزء من الصيغة بتلبة المروجع ، نقدر content بـ reuse .

الفكرة من الـ component أن إتنا نقطع الموجع بجانبنا للاستراد وكل جزء نقدر زكيته من أي مكان آخر من سره بنفس الكورس .

أهم مزايا الـ component هي reusability وسهولة تنظيم الـ UI بية .

Q) أزاي نعمل Vue component

من الـ class Vue.extend يفتح الـ component .

الـ component يتطلب 2-parameters ، الأول هو الـ options الـ second هو الـ template .

```
Vue.component('navbar', {
  ① template: `<nav> ${} </nav>`, // output يتألف من الـ view . view
  ② data() {
    return {
      name: 'Hello',
    }
  },
  ③ methods: { } // business logic
}); // component
```



١٤٣٨ محرم ١٩
١٧٧٣ بابه

OCTOBER

Thursday

الخميس

Q) لــ data property بــ vue instance عبارة عن عبارة function ، إنها object ، إنها component عبارة عن object بــ ترجع object {

البيــ دــ الصــرــيــقــةــ الــ كــ الــ Vue instance مــ نــ يــ تــولــ مــ نــ يــلــ الــ "component instance"

الــ Vue instance دــ وــ عــبــارــةــ عــنــ نــســخــةــ وــاـمــحــهــ مــعــرــىــ عــنــ فــيــرــهــاـمــســتــكــرــ ،ــ بــالــتــاكــ الــ dataــ الــ Vueــ فــيــلــ هــســكــوــنــ خــاصــةــ يــســلــ .ــ

ــ لــ ماــ الــ Vue componentــ نــقــرــ نــعــلــ مــنــ أــكــلــ مــنــ نــســخــةــ كــيــ الــ applicationــ وــ مــعــشــانــ الــ Vueــ يــعــلــ نــســخــةــ مــنــ الــ componentــ الــ لــازــمــ كــلــ نــســخــةــ يــرــكــوــنــ لــيــلــ الــ copyــ الــ اــذــاصــ بــيــلــ مــنــ الــ component's dataــ مــحــاـنــ نــيــزــهــ مــنــ بــعــقــيــ .ــ

ــ بــالــتــاكــ مــرــىــ الــ componentــ ســبــارــىــ عــنــ dataــ خــلــيــنــاـ الــ componentــ بــرــجــعــهــ objectــ عــلــىــ تــائــىــ كــلــ مــاـ الــ Vueــ يــعــلــ مــنــ instanceــ مــنــ componentــ بــرــجــعــهــ dataــ functionــ جــىــىــ مــنــ الــ dataــ الــ Vueــ .ــ

Q) اــلــ اــنــ اــنــعــلــ مــعــ الــ componentــ مــنــ instanceــ مــعــ الــ html tagــ مــعــ الــ <navbar></navbar>

Name

Tel.

Email

كتوبر ٢٠١٦

١٩) ابرهاد props
هي طريقة لنقل الـ data من parent component إلى child component.

- ١) نعمل خطوتين للتعامل مع props:
- ١-) أول خطوة نجتازها "component selector"
- ٢-) ثانية خطوة سنتقبع في child component

```
new Vue({  
  el: "#app",  
  data: { posts: [{}, {}, {}] }  
});
```

```
Vue.component('card', {  
  template: '

~<div>~, ~</div>

',  
  data: function(){  
    return {}  
  },  
  props: ['posts']  
});
```

نستقبل props من parent component من خلال child

① <card :posts="posts"></card>



١٧ محرم ١٤٣٨

October

٢٠١٦ أكتوبر

٨ باب ١٧٧٢

Tuesday

ثلاثاء

Q) أي الفرق بين الـ local component وـ global component.

الفرق الأساسي ما بينهم هو أنهما في application فـ local component هي خدمه من أى مكان وـ global component هي خدمه من أى مكان وطريقة ابتدأته.

الـ local component ينبع عن باستخداً him من `Vue.component()` وـ `Vue.component()` هي المـ application التي هي لـ `Vue.component()` لأنها هي المـ application التي هي لـ `Vue.component()`.
ex: `Vue.component('c1', { template: '<p></p>' })`

`Vue.component('c1', { template: '<p></p>' })` | global component

`new Vue({ el: '#app1' })` | target element
الـ `el` mount in الـ `app1`

Name _____

Tel. _____

Email _____

٢٠١٦ أكتوبر

يتبع local component الـ scope داخلاً بـ **local component** (١) ومن غيره نستخدم **Vue component** (٢) معنون بالـ **local component** (٣) متعرف فيه.

الـ **components option** (٤) الـ **local component** (٥) **Vue component** (٦) خارج الـ **local component**.

(ex)

```
var c1 = {  
  template: '<div></div>',  
};
```

```
var parentComponent = new Vue({  
  el: '#app1',  
  components: {  
    'my-local-component': c1,  
  },  
});
```

(ما يحضر)

١- الـ **mount** (٧) الـ **elements** (٨) الـ **template** (٩) الـ **target element** (١٠).

٢- الـ **override** (١١) الـ **template property** (١٢) الـ **target element** (١٣).

اكتوبر ٢٠١٦

October

Sunday

الأحد

١٥ محرم ١٤٣٨

٦ باب ١٩٧٣

١٦
16

الأحد

⑦ Referencing with Refs :-

الـ dom node هي طريقة سهلة لـRefs لـ element .
 الحقيقي بـ target معنـ بـ كل مـ target ، بالـ target
 نقدر نـ target لـ element .

" اذـ نـ target لـ element " .

١. ما خطوه ؟
 ① اول خطوه هو إخراج لـ element لـ reference .
 الـ reference يـ target dom node وـ target علىـ element .
 بـ target اـ صـنا مـ target .

special attribute `ref` يـ target reference .
`ref = "blt"` يـ target Vue .

٢. ثـ خطوه منعـ لـ component لـ target لـ component .
 ref لـ read لـ element .
 \$refs هوـ builtin property لـ component .

① `<input ref="blt" />`

② `this.$refs.blt`

Name

dom element كـ target

Tel.

Email

ـ اـ target لـ element

ـ اـ target لـ element

ـ اـ target لـ element

أكتوبر ٢٠١٦

٢٩	٢٢	١٥	A	١
٢٨	٢١	١٤	T	٢
٢٧	٢٠	١٣	T	٣
٢٦	١٩	١٢	T	٤
٢٥	١٨	١١	T	٥
٢٤	١٧	١٠	T	٦

⑧ Lifecycle hooks:-

methods عباري عن دلائل vue lifecycle hooks .
كل method يتبع تسلقانياً من مرحلة . Vue component محيطة من دورة حيّات الـ

- ↳ Vue component الـ يمر بـ
- ↳ Vue component الـ دورة حيّات

(أولاً) كلـ دورة حيّات hooks ، هيـ المراحل الـ memory ـ يمر بـ الـ instance ـ علىـ ما توصلناـ الـ data وـ تتظـرـ حـ

ـ كلـ مرحلة / hook / logic مـ يـ بـuilt-in methods كلـ مرحلة .

(ثانية) الـ instance / المراحل الـ يـ بـuilt-in الـ

① beforeCreate ()

③ beforeUpdate ()

created ()

updated ()

② beforeMount ()

④ beforeUnmount ()

mounted ()

unmounted ()

١٤
14

دورة مرحلة الابتكاف بين الـ
initialization vue instance
يجب إتبني لـ instance لكن من غير
value لـ .

beforeCreate()

جبر ستكون
properties ازدياد

٣) المرحلة دعو vue اتعمل
خلاص يبن او properties بتوحة اخطفهم

created()

value او assign
properties او

compiling vue مسارات ← template
template: ' ← template
ان element دوار template
الـ mountin root element عن او
لـ .

compiling
template

```
<div id="app">  
  template  
</div>
```

قبل ما يدخل او dom node
الـ template يدخل قبل ما يركب
target element او template
او root element
باتبعنا الاك دوا

beforeMount()

قبل ما يركب
target elements

سته او dom node [اتبني و اترکب
<div id="app"> target element عن او

mounted()

سته تركب كل
حاجه من مكانته

Name

Labanomand

Tel.

Email

أكتوبر ٢٠١٦

٢٩	٣٠	٣١	١	٢
٣	٤	٥	٦	٧
٨	٩	١٠	١١	١٢
١١	١٢	١٣	١٤	١٥

October 2016 ٢٠١٦ أكتوبر

Thursday الخميس

١٢ محرم
١٤٣٧
الخميس

١٣
13

data property run لو صدر تغيير من الـ dom tree وقبل ما التديل يسمع من الـ virtual dom يين التغيير سمع من الـ beforeUpdate()

بعد التغيير سمع من الـ updated

template قبل ما الـ dom node يتغير الـ dom tree يسمع من الـ beforeUnmount()

بعد الـ dom node انتزح / انسحال من الـ dom tree unmouted()

هناك مثل `logic` معين في مرحلة مختلفة من مرافق `Vue instance` يتابع امرها `hook method` `override` على `target` `element`.

```
const app = Vue.createApp({
  data() {
    return {};
  },
  methods: {},
  computed: {},
  components: {},
  template: '',
  beforeCreate() {},
  created() {},
});
```

`app.mount('#app')` → target element ← المـ mount is call

١٢
12

١١ محرم ١٤٣٨
٢٠١٦
١٧٢٢
Wednesday

October

٢٠١٦ أكتوبر

الأربعاء

ذكرنا حلوٰ لـ `test` (وتتابع)
الـ `data` / المراحل بـ `instance` `hooks`
ـ `lifecycle hooks` من `alert()` (ex)

ما هو لهم

`beforeCreate() { alert('---'),`
`created() { alert('---'),`

ومنها

ـ `concise methods` أو
ـ `methods` مختصرة للإستخدام `syntax`
ـ `object literals` `ES6` `syntex` `concise method` اذ

`const obj1 = {`

`m1: function() { }},` و `m1()`

`}`

ـ `ES6` قبل

ـ `ES6` بعد

ـ `concise method`

ـ `state` يعني حاصلـ افضل فـ `أنا` `ما` `واتابعـ`ـ `لكـ`
ـ `يحصل علىـ / أنا حوفـ` `حالـ`ـ `لـ`ـ `أـ`

Name

Tel.

Email

أكتوبر ٢٠١٦

٢٠١٦

١٠ محرم ١٤٣٨
١٢٢٢ ١ باب

October 2016 ٢٠١٦ أكتوبر
Tuesday Tuesday اشلاقاء

١١
11

⑨ deal with Form in Vue :-

اول خطوة : نجعوز ال form component

SignupForm.vue

```
<template> <script> <style>  
  <form> export default {  
    </form> } </script> </style>  
</template>
```

ثانية خطوة : register id : خطيء ال register id
النحو : في App component الـ nested components

App.vue

```
<template> ③ مسند علیه  
  <SignupForm />  
</template> ④ مسند علیه  
 export default {  
   name: 'App',  
   components: { SignupForm }, register ③ مسند علیه  
   App component الـ  
 }  
</script>
```



١٠ محرم ١٤٢٨

٢٠١٦ توت ٢٠١٣

October

Monday

اكتوبر ٢٠١٦

الاثنين

10

ثالث خطاوه :

- template من ال form inputs المجهز من الدايركتيف data property ملحوظة input مع كل component data حمله data property
- عيّنات تستقبل قيمة ال data الذي تم تحديده
- عيّنات values track التي تم تحديدها، هنا نستخدم two-way data binding + v-model directive
- يحصل على ال object من form-data حيث نغير بعده مع ال API

```

<template>
  <form>
    <input v-model="post.email" />
    <input v-model="post.password" />
    <input type="submit" @click="handleSubmit" />
  </form>
</template>
<script>
  export default {
    data() {
      return {
        post: { email: '', password: '' }
      }
    }
  }
</script>
  
```

Name _____

Tel. _____

Email _____

اكتوبر ٢٠١٦

٢٩	٣٠	١	٢	٣	٤	٥
٢٨	٣١	٢	٣	٤	٥	٧
٢٧	٣٢	٣	٤	٥	٦	٨
٢٦	٣٣	٦	٧	٨	٩	١٠
٢٥	٣٤	٧	٨	٩	١٠	١١
٢٤	٣٥	٨	٩	١٠	١١	١٢

October 2016 ٢٠١٦ أكتوبر

Sunday

٨ محرم ١٤٣٥
٢٩ تون

الأحد

٩
٩

٤- رابع خطوة ، submit زار ال submit ونجحته ال Validation

formdata.error خاصية بـ data property لـ ①
.error عبارة عن flag لوجود ال data property ②

```
<script>
export default {
  data() {
    return {
      post: {},
      errors: {
        email: '',
        password: ''
      },
      is_valid: true,
    }
  },
  methods: {
    handleSubmit() {
      // validation
    }
  }
}
```

العامل الـ v-model للتعامل مع الـ form

Part ②



٢٠١٦ محرم

١٤٣٨ قتوت

٢٨

October

Saturday

2016

٢٠١٦

أكتوبر

السبت

- 1- Vue CLI.
- 2- Share data b/w components.
- 3- slots.
- 4- dynamic component.
- 5- How to consume API.
- 6- routing
- 7- Vuex (state management)

Name _____

Tel. _____

Email _____

أكتوبر ٢٠١٦

٢٩ ٢٨ ٢٧ ٢٦ ٢٥ ٢٤ ٢٣ ٢٢ ٢١ ٢٠ ١٩ ١٨ ١٧ ١٦ ١٥ ١٤ ١٣ ١٢ ١١ ١٠ ٩ ٨ ٧ ٦ ٥ ٤ ٣ ٢ ١

① The Vue CLI :

① اية هو الـ Command line interface operating system الذي نتكلم معه على سطح المكتب مع GUI التي يأخذ اوامر تحضير وينفذ بستخدام اوامر مرئية كذلك.

② اية هو الـ Vue CLI اداة command line tool بحسب هل علينا دعم او بناء مشروع vue . من حيث انها بتوفر مجموعة اوامر وكل امر ينفذ task معين من المشروع.

- مثلاً) ١- بناء الـ boilerplate او الدباجة بتاعة ال Vue app
- ٢- بين افتح component
- ٣- تحويل لايديك المشروع على الـ production
- ٤- تشغيل المشروع على live server
- ٥- وهذا

الخاصية الـ Vue CLI هي tools development tools التي تساعدنا اشارة بناء مشروع vue application ، مبروك اية بيوفر لنا وقت من خلال رته بين الـ boilerplate او الدباجة الى خاتصالها من غير ما نكتبه بنفسنا .

يعلن دو سطر امر واحد ← هيستاد vue application بـ كل .
 سطر امر واحد ← هو امر vue app .
 ← هو لايديه build او production بـ اول app .
 ← هو تشغيل component بالدباجة بتاعته .

أكتوبر ٢٠١٦

October

Thursday

الخميس

٥ محرم ١٤٣٨

٢٢ توت ١٩٧٢

٦

يمكن تثبيت Vue CLI أوامر الـ Vue CLI عارضه هنا،
لما زلت تنزل الـ Vue CLI عارضه هنا.

»npm i -g @vue/cli«

»vue --version«

① To create new Vue project.

»vue create myProject«

- manually select feature

- Babel, choose Vue version, linter

أو يمكن تختار ببس وجد choose Vue version

كذلك تثبيت الى انه مازالت لا تحتاجه.

- 3.x

- In package.json

- n

الاختيارات الى اصنا نختار دينه بنصيحت الـ presets
نتحقق تختار عادة باسم معيون يمكن بعد كده نختار
لا presets الى اصنا علينا ذلك بل ما كل مرة نعمل
نغير الـ options دينه من شائى.

الـ preset دينه مجموعة الاختيارات الى اصنا
بنختارها.

Name

Tel.

Email

أكتوبر ٢٠١٦

السبت	٣	٤	٥	٦	٧	٨
الأحد	٨	٩	١٠	١١	١٢	١٣
الإثنين	١٣	١٤	١٥	١٦	١٧	١٨
الثلاثاء	١٨	١٩	٢٠	٢١	٢٢	٢٣
الأربعاء	٢٣	٢٤	٢٥	٢٦	٢٧	٢٨

الـ file باستهار (.vue) ينتمي إلى single file component يتكون من three main parts أو الـ component أي parts التي يحتوي على الـ data logic and presentation. هنا: الـ HTML and CSS and JS.

الـ SFC هو file يحتوي على الـ three parts متواجدة في optional Vue component أي الـ three parts هي data logic and presentation. يمكن حذف HTML from the file and it will still work. بالتأكيد any file باستهار (.vue) يعتبر Vue component without data and style.

أى ملف باستهار (.vue) يعتبر Vue component وبينما الملفاته باسم single file component (SFC) وهو الملف الذي يحتوي على الـ three parts of the component.

Vue instance + option يحتوي على Vue instance وته عبارة عن object من single file component all the components can use the same local variable or function. all the components can use the same global variable.

ملحق الـ App.vue ينتمي إلى base component أو parent component أو wrapper component.

لا component can use the same variable in two instances unless the component is reused in two places. كل component has its own data and functions and methods. all the data and functions are private to the component. data object will not be available outside the component.

الـ function في component هو data member
يتوجه object بالاتك الى data member
invoke يطلب data member الى الخادم بالـ component
وهو يرجع الى data member الخادم . invoke

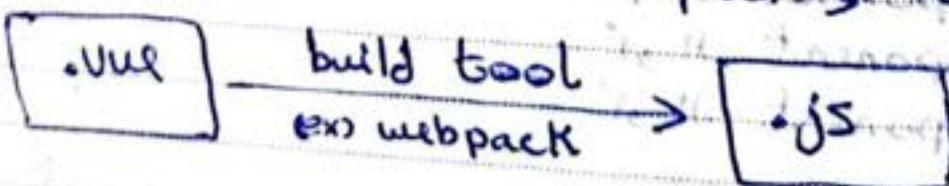
٦١. instance من component يخزن بـ instance

بيان نخلي كل instance يخزن لـ data member
component بـ instance بـ data member بتاتره
تبعد function وليس object

الـ SFC هو vue file كل اجزاء الـ component

الـ compile أو build في app كل ملفات الـ SFC متتحول للبنية
عادية .

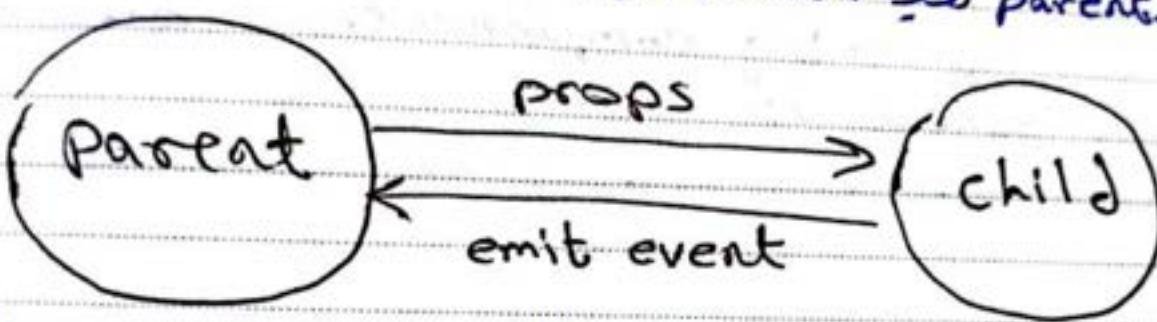
المؤول عن عملية تحويل ملف vue الى client-side tools في vite أو
webpack أو vue-cli



Name _____ Tel. _____ Email _____
أكتوبر ٢٠١٦
ملف الـ index.html والـ main.js مين قربيت
صنيع

لـ parent component من data .
 مثـان يـدـعـت بـین خـدمـة child component

لـ child component من data .
 مثـان يـدـعـت بـین خـدمـة parent component
 باـضـعـار عـبـرـهـاـ منـ custoـmـ eventـ سـنـ اـلـ child
 والـ لـisـtenـ هـيـ parentـ علىـهـ



مـثـان نـتـحـلـ dataـ مـاـبـينـ مـاـلـقـةـ
 بـعـدـ vuexـ تـحـلـ state managementـ

ملـفـ الـ r~ootـ هوـ main.jsـ فـيـ entry pointـ الـ k~eyـ

الـ h~ost~ كلـ الـ appـ تـحـلـ كلـ الـ global vue instanceـ

كـلـ الـ Reactـ Angularـ Vueـ dzـ frameworksـ

كـلـهمـ بـيـتـعـلـواـ بـارـ dzـ moduleـ dzـ modul~eـ الـ ak~ow~

لـ exportـ وـ importـ بـيـنـ themـ بـيـنـ modulesـ

مـثـان دـخـلـ compo~nentـ جـداـ

اـولـ صـاحـبـهـ : بـيـنـ allـ importـ

ثـانـ ~ـ : بـيـنـ compo~nentـ allـ importـ

ثـالـثـ ~ـ : بـيـنـ compo~nentـ arrayـ or registerـ

ثـالـثـ ~ـ : بـيـنـ templateـ لـ componentـ

- ① import
- ② register
- ③ ...



١ محرم ١٤٢٨
٢٢ سوتو ١٩٧٢

October

Sunday

أكتوبر ٢٠١٦

الأحد

عشان نخسر الـ local style يعني
الـ empty attribute هنستعمل اسلمة scoped

<style scoped>

الـ style اللي هنا scoped

</style> - مثال component اللي هو خارج
فقط.

الـ component اللي بنادي بيته الـ custom html tag

tag component بنفس معنیه الـ

<my-component>

Name

Tel.

Email

أكتوبر ٢٠١٦

السبت ١٥ أكتوبر ٢٠١٦

② sharing data b/w components

نوعية نوادر من العلاقات بين components وبعضاً

- child → parent ① ملائمة
- parent → child ② ملائمة

② child → parent من data

لهم تخدم props
الـ html attribute هي props
الـ props not member يحتوى على vue instance
بخط فيه اسماء الـ props الى احنا مسنيبل.

الخطوات كالتالي:

① في parent component:
child tag component يحتوى على prop
ونساورة بالقيمة الى ما يريده شاركها.

② من child component:
ما هو props member من parent
property

① <child blz="hamza" />
② props: ["blz"],

أول ما يخرج من child
data() { return { blz: "hamza" } }

ما يكتب هنا
behind the scene

}},



٢٩ ذوالحجّة ١٤٣٧

٢٠ توت ١٣٣٣

September

Friday

2016 ٢٠١٦

سبتمبر

الجمعة

هل يمكن نقل props الى member property (Q)
يُنفس تم الـ props الى member property (A)

لا ، لأن الـ props vue instance هي تم عرضها
من الـ data property من create . يُنفس member property باسم .

(ex.)

```
data(){
  return{
    blz: ''
  },
  props: ['blz'],
}
```

behind
the
scene

```
data(){
  return{
    blz: '',
    blz: ''
  }
}
```

- html attribute بـ props على child
- قيمة الـ parent ينبعها من الـ props
- html attribute props معاملة parent من child
- data property الـ child

هـما خصـوـتين بـ props : ابـعـت و ابـتـام .

① <child p1="" p2="">

② props: ['p1', 'p2'],

Name

Tel.

Email

سبتمبر ٢٠١٦

parent الـ data من الـ props

٢٨	٢٩	٣٠	١
٢٩	٣٠	١	٢
٣٠	١	٢	٣
٣١	٢	٣	٤
٣١	٢	٣	٤

٢٨ ذوالحجّة ١٤٣٧
١٩ سوت ١٤٣٦

September 2016 ٢٠١٦

Thursday

سبتمبر
الخميس

٢٩
29

②) ازای نسبت child من data من ای

کل مانند parent لـ child الـ data الـ events

method پر تین event ای raise کیا شان ڈی
parameter تائیں 2 parameter پر emit() ایسے
دو optional اخیزیات!

emit = method child component ای
custom event ای

① methods:
sendData() {
event data ای
this.emit('bit', data);
event ای emit / raise کیا

② event listener جو parent component ای
custom event ای

③ <child @bit = "handleBit" />
methods:
handleBit(data){
} {
} {

③ call is child component ای
custom event ای emit =

④ <button @click = "sendData" > <button>



٢٧ ذوالحجjah ١٤٣٧

١٨ توت ١٤٣٩

September

Wednesday

2016

سبتمبر
الأربعاء

③ slots

الـ slot هي طريقة بــ ســ مــ حلــ لــ نــ بــعــتــ من الـ parents للـ child.

مــ يــ هــ الـ slot إــ يــ إــ صــ نــ حــ دــ نــ عــلــ reusuable component وــ تــ غــيرــ مــ زــ مــ الـ content بــ تــ اــعــةــ مــ عــلــ ماــ يــ عــدــلــ الـ parents ذاتــ نــســةــ component.

لو الـ slot أوــ slot component يــ حــتــوــ عــلــ compo~ent المــ صــرفــ الـ html content place~holders لــ الـ parents component من الـ parents component.

الـ slot اــعــرــكــ اــكــنــةــ اــلـ custom content لــ placeholder.

بنــظــلــ الـ slot يــ بــارــخــاـمــ لــ الـ slot element.

لــ الـ slot يــ رــنــدــرــ compo~ent فيــ الـ slot المــ حــتــوــ عــلــ content وــ يــحــطــ مــكــانــهــ الـ child component من الـ parents component.

الـ child component الـ child يــ حــتــوــ عــلــ الـ content المــ حــتــوــ عــلــ closing tag وــ الـ open tag.

<child>

→

الـ child يــ حــتــوــ عــلــ content المــ حــتــوــ عــلــ slot element.

Name _____

Tel. _____

Email _____

سبتمبر ٢٠١٦

٢١	٢٢	٢٣	٢٤	٢٥	٢٦	٢٧
الأحد	الاثنين	الثلاثاء	الخميس	الإثنين	الثلاثاء	الأربعاء
٢٩	٣٠	٣١	٢	٣	٤	٥
الجمعة	السبت	الجمعة	السبت	الجمعة	السبت	الجمعة
٣٠	١	٢	٣	٤	٥	٦

من placeholder / slot من مثل ذلك من يحتوي على child component أو slot كل ذلك في المقام الأول على child component من نوع named slot.

```
<slot name="bit"></slot>
<template v-slot:bit>
</template>
```

من غير slot هو default slot أو named slot يحتوي على ذلك من component ولكن ميزة هذا هو أنه يمكنه بمحض اختياره إدخال أي محتوى.

```
<template>           child component (ex)
  <slot name="top"></slot>
  <slot></slot>
<template>
```

: parent component

```
<child>
  <template v-slot:top>
    <p>hello</p>
  <template>
    <p>default</p>
  </child>
```

٢٦
26

٢٥ ذوالحجّة ١٤٣٧

١٦ توت ١٩٢٢

September

Monday

٢٠١٦ سبتمبر

الاثنين

(١) الفرق بين `slot` والـ `{props}`

الدشنين هما طرق لارسال data من `parent component` الى `child component` ، لكن الفرق

الا `slot` يستخدمه متى نريد من `props` الـ `data` الى `child` ، والـ `child` يقرر ياخذ `data` او `logic` معين .

الـ `child` ياخذ `data` من `props` او `slot` .

الـ `child` لا ياخذ `content` من `slot` الا `placeholder` كـ `content` يستخدم static content من غير خدمة من `logic` معين .

Name _____

Tel. _____

Email _____

سبتمبر ٢٠١٦

٢٤	٢٥	٢٦	٢٧
الاثنين	الثلاثاء	الجمعة	السبت
٢٥	٢٦	٢٧	٢٨
٢٦	٢٧	٢٨	٢٩
٢٧	٢٨	٢٩	٣٠

④ dynamic component :-

ـ ١- switch : library يسمى dynamic component .
ـ ٢- ما بين " آكثر من condition " في component

ـ ٣- بنحو dynamic component functionality .
ـ ٤- لو سنتا آكتر من component لفهم منه البرو.
ـ ٥- لكن كل واحد له UI مختلف .

ـ ٦- لو ما بين dynamic component ما بين آكتر من condition .

ـ ٧- علشان نفتح dynamic component الـ <component> element .
ـ ٨- منفتح الـ <component> element علشان placeholder .

ـ ٩- عندها is attribute الـ & bind .
ـ ١٠- render .

<component :is="targetComponent" >

```
data() {
  return {
    targetComponent: 'componentA',
  }
}
```

ـ ١١- قيمة الـ component is attribute اسم الـ component .
ـ ١٢- ما بين نظرة .

٢٤
24

٢٢ ذوالحجّة ١٤٣٧

١٤ قوت ١٩٢٢

September

Saturday

2016 ٢٠١٦

سبتمبر

السبت

(المظاہر)

١. الـ `switch` رـ `components` ما يـ `use` لـ `switch` عـ `variables` لـ `switch` مـ `import` مـ `Registers` بـ `use` كـ `state` مـ `Registers` مـ `Components` member لـ `Components`

٢. `component` وـ `dynamic component` وـ `runtime component` مـ `different` اـ `between` اـ `component`

٣. `component` مـ `component` مـ `switch` مـ `create` اـ `component` وـ `unmount` فـ `switch` اـ `component` اـ `component` اـ `component`

٤. مـ `keep-alive` مـ `component` مـ `keep-alive` مـ `component` مـ `element` مـ `wrap` مـ `unmount` مـ `keep-alive` مـ `element`

٥.  `keep-alive` مـ `memory` مـ `2-hooks` بـ `hook` جـ `parent` مـ `child` مـ `parent` بـ `parent` مـ `child` مـ `parent` مـ `child`

`activated()`

`deactivated()`

Name

Tel.

Email

سبتمبر ٢٠١٦

٢٢	٢٣	٢٤	٢٥
٢٣	٢٤	٢٥	٢٦
٢٤	٢٥	٢٦	٢٧
٢٥	٢٦	٢٧	٢٨
٢٦	٢٧	٢٨	٢٩
٢٧	٢٨	٢٩	٣٠
٢٨	٢٩	٣٠	٣١
٢٩	٣٠	٣١	١
٣٠	٣١	١	٢

أولاً الـ `switch` مابين `componentA` و `componentB` باستخراج `keep-alive`؟

(١) بدون استخراج الـ `keep-alive` او `create` `componentB`، يمكن ان . invoke `componentB` `created()` hook

ثانياً: `unmount` `componentA` ، يمكن ان . invoke `componentA` `unmounted()` hook

(٢) باستخراج الـ `keep-alive` او لا، فـ `componentA` / `componentB` `create` `componentB` . `deactivated()` `activated()` و `activated()` `deactivated()` .

ثانياً: `create` `componentB` لأول مرة فقط.

. `deactivated()` `componentA` : ثالثاً،

. `activated()` `componentB` ، اولاً،

مزدوجاً الـ `keep-alive` بـ `componentA` . `destroy` `componentA` `state` `memory` . `unmount` `componentB` `caching` و `destroy` `componentB`



٢١ ذوالحججة ١٤٣٧

١٢ توت ١٣٣٣

September

Thursday

2016

٢٠١٦

سبتمبر
٢٠١٦

٥ Deal with API

للحصول على البيانات من API نستخدم axios.js 3rd party library

→ get data from API

methods: {

```
async loadposts() {
  let allposts = await fetch('url');
  let allpostsasjson = await allposts.json();
  this.posts = allpostsasjson;
}
```

→ post / send post request

methods: {

```
addpost(data) {
  fetch('url', {
    method: 'POST',
    header: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  }).then(resp => resp.json())
  .then(data =>
}
}
```

Name

API لـ fetch في المقدمة
mounted lifecycle hook

Email

سبتمبر ٢٠١٦

٢٢	٢٣	٢٤	٢٥	٢٦	٢٧	٢٨
٢٩	٣٠	٣١	١	٢	٣	٤
٥	٦	٧	٨	٩	١٠	١١
١٢	١٣	١٤	١٥	١٦	١٧	١٨
٢٩	٣٠	١	٢	٣	٤	٥

○ json server

هو مسار من software أو tool يُستخدم
لتحقيق REST API وهم طلب json عندهم من المحرر.

الهدف من json server هو إنشاء API
لما ينطوي على front-end test.

(ex)

أمثلة على data endpoint وتصفح API من db.json

```
{
  "posts": [
    {
      "id": 1,
      "title": "blf",
      "body": "..."
    }
  ]
}
```

↓
endpoint of the API

json-server -l install لـ

» npm i json-server <

أمثلة على json server execution API وهم

» json-server --watch db.json لـ
مسار الملف

<http://localhost:3000/posts>

<http://localhost:3000/posts/1>



١٩ ذوالحججة ١٤٣٧

١٠ تقوت ١٤٣٧

September

Tuesday

2016

٢٠١٦

سبتمبر
الثلاثاء

ابدأ بـ API للـ calling فيه فعل فيه hook
 dom nodes hook يمكنه كون المكالمات
 browser في dom tree الـ استهلاوا وانتظروا حتى الـ

mounted () {

fetch('http://localhost/posts')

.then(res => res.json())

.then(data => this.allposts = data)

.then(err => console.log(err.message));

}

db.json

```
{
  'posts': [
    {
      'id': 1,
      'title': 'Post 1',
      'content': 'Content of Post 1'
    },
    {
      'id': 2,
      'title': 'Post 2',
      'content': 'Content of Post 2'
    }
  ]
}
```

← http://localhost/posts

← http://localhost/posts/1

Name _____

Tel. _____

Email _____

سبتمبر ٢٠١٦

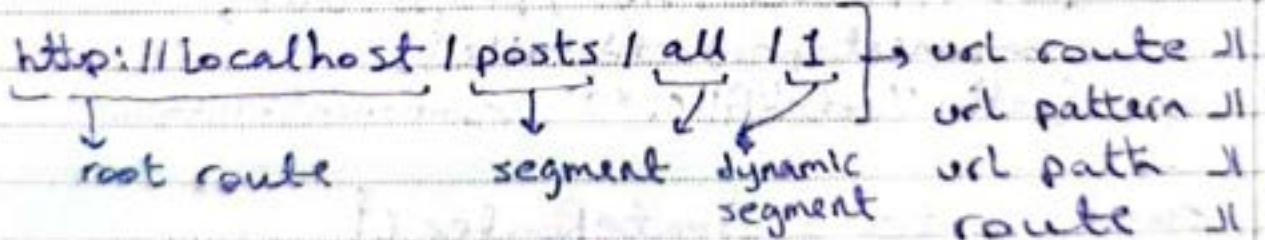
٢٣	٢٤	٢٥
٢٤	٢٥	٢٦
٢٥	٢٦	٢٧

⑥ Vue Router

route هو الـ URL في address bar وهو Link components هو الـ routes التي تنقل ما بين الـ UI (url bar) و url links (او routes) لـ UI.

ذكرنا routing في زمن ذكرنا dynamic component، يفضل نستخدم dynamic component مع المترافق الصغير، لكن لو اراد جعل scale كبير يفضل نستخدم router.

root route هي الـ URL في base url كل segment بعد الـ slash في URL، asterisk * الـ alternation not found page url أي طاجة منكبة خـ الـ URL.



كل component له route كـ global route في routes array، route وـ match، match لـ default， match لـ hash， match لـ history.

hashes route ①: route || history route ②

Vue يستخدم كل دفعات history لـ hasheds، createWebHistory()

١٨
18

١٧ ذوالحجjah ١٤٣٥

٢٠١٦ توت ٨

September

Sunday

2016 ٢٠١٦

سبتمبر
سبتمبر

{ vue routers ١٨ جوليل انيطوات اسما (Q)

1- > npm i vue-router@4.0.5

2- in main.js

```
import {createWebHistory, createRouter}
from 'vue-router';
```

3- in main.js

give route to all components it import jst

4- const routes = [

```
[{path: '/', component: home},
{path: '/about', component: about},
{path: '/products/:id', props: true,
component: productdetails},
{path: '/catchAll(.*)', component: NotFound}];
```

const router = createRouter({

```
history: createWebHistory(),
routes});
```

5- createApp(App).use(router).mount("#app");

Name

Tel

Email

الـ path ١٨ ارسف اسما routes ١٨ jst
component ١٨ routes ١٨ jst

سبتمبر ٢٠١٦

١٢	١٣	١٤	١٥	١٦
١٣	١٤	١٥	١٦	١٧
١٤	١٥	١٦	١٧	١٨
١٥	١٦	١٧	١٨	١٩
١٦	١٧	١٨	١٩	٢٠

سبتمبر ٢٠١٦ September 2016

السبت Saturday

١٦ ذو الحجة ١٤٣٥
٧ قرطبة ١٤٣٥

سبتمبر

٢٠١٦

السبت

١٧
17

٦- in base component (App.vue)

- navigate إلى المكان الذي هو router-link
- placeholder الذي هو router-view

<template>

<router-link to="/about">about</router-link>
≡

<router-view></router-view>

</template>

٧- \$route.params.id → dynamic segment
this.\$route.go(-1) → history url
this.\$route.push('/') → programmatically navigate

{path: '/', redirect: '/home'}

{path: '/home', component: home}

١٦
16

١٥ ذوالحجjah ١٤٣٧

٦ توت ١٩٧٣

September

Friday

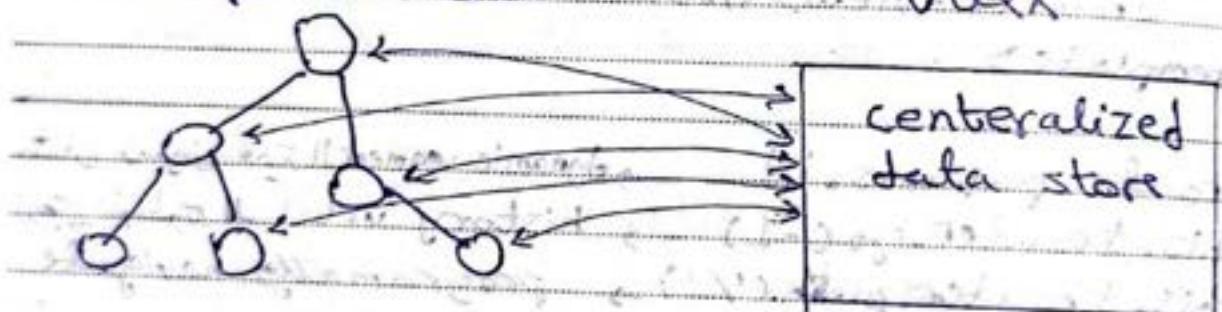
2016 ٢٠١٦

سبتمبر
الموسم

⑦ Vuex

أولاًVuex هي معمارية عن state management pattern بين كل الأركان. global state. يتحقق ذلك من خلال مكتبةVuex التي توفر events أو props ودون عن طريقVuex centralized store. الموزع центральный store الذي يحتفظ ببيانات application data لـ all the components.

Component tree vs. Vuex



فيVuex، data state لا ينبع من local state، بل من centralized data store. render job. وينبع local state من composition function.

عندما نختار local state ما بين components، communication ما بينهم وبين store من خلال props down events up! إن إلينا نعمل على الطريقة ذات مسافة جهاز (local) كبيرة، states من props والevents! إننا يمكن نجح كل local store ببساطة مكان واحد في store / الموزع الرئيسي.

Name _____

Tel. _____

Email _____

سبتمبر ٢٠١٦

T1	T2	T3	T4
T5	T6	T7	T8
T9	T10	T11	T12
T13	T14	T15	T16
T17	T18	T19	T20
T21	T22	T23	T24
T25	T26	T27	T28
T29	T30	T31	T32
T33	T34	T35	T36
T37	T38	T39	T40

١٨ - الـ vuex هو خزينة مشاركة للوصول إلى state لأى component tree أو component

١٩ - الـ vuex زرعة في vue instance data يعني أن data في component هي reactive ، يعني لو تغير data واحد على فيه يتأثر جميع components الذي يستخدم data زرعة هم بالتدريجية وبشكل تلقائي مما تتقبل العقيدة الجميلة.

٢٠ - لو كنا نريد أن نعمل على state في الـ component ، فيجب أن ندخلها في vuex store بالـ computed pattern مع التعامل مع التغيير من الـ state.

٢١ - بنية الـ vuex زرعة الـ vuex

١ - ما يندر على vue instance داخل vuex store ، data member يحتوى على vue instance والـ reactive state member يحتوى على vuex store ، والـ action member يحتوى على vue instance والـ mutation member يحتوى على vuex store

٢ - actions member يحتوى على vuex store والـ mutations member يحتوى على vue instance والـ getters member يحتوى على vuex store

٣ -mutations member يحتوى على vuex store وذلك عن طريق إرسال طلب التغیر على state وـ state يحول إلى mutations call ، بعد ذلك يحول إلى actions call ، state أو update

الغرض إن الـ vuex store يحتوى على vue instance وذلك عن طريق إرسال طلب التغیر على state وـ state يحول إلى mutations call ، بعد ذلك يحول إلى actions call ، state أو update

١٤
14

١٤٣٧ ذوالحجjah ١٤٣٧

١٩٧٧ : ثوب

September

Wednesday

2016 ٢٠١٦

سبتمبر

الأربعاء

Vue

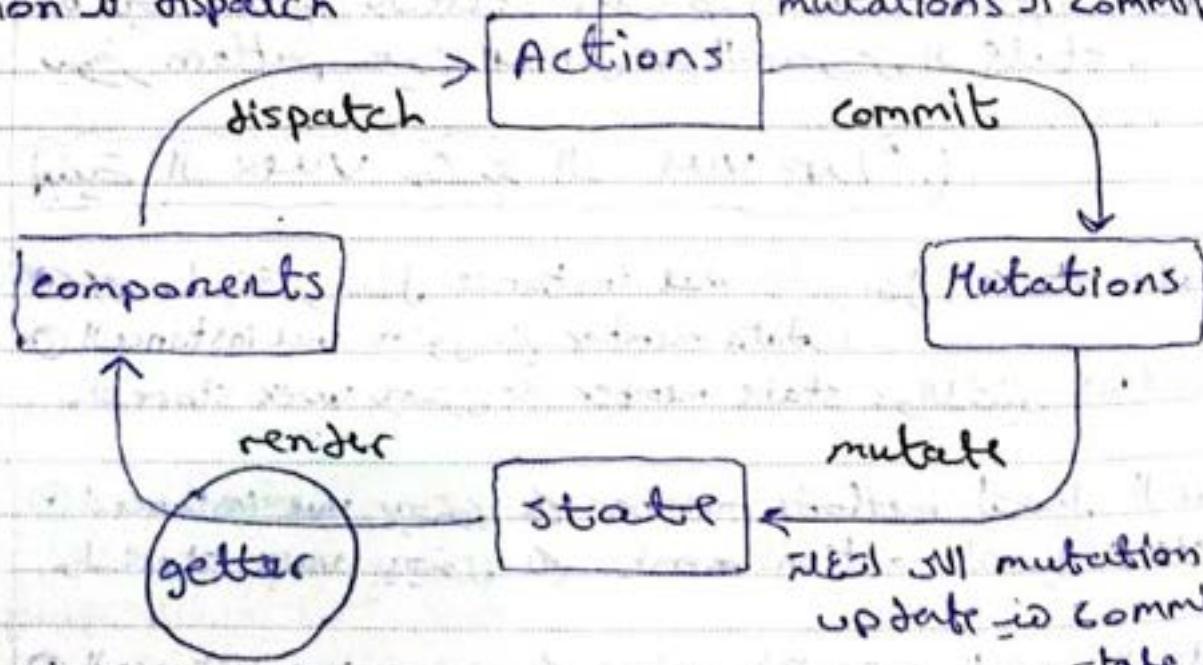
```
const app = new Vue({
  data: { },
  methods: { },
  computed: { }
});
```

Vuex

```
const store = new Vuex.Store({
  state: { },
  actions: { },
  getters: { },
  mutations: { }
});
```

عن طريق component
action || dispatch

dispatch || action
mutations || commit



getter || سترچر
state || موجل

Name _____ Tel. _____ Email _____

سبتمبر ٢٠١٦

٢٢	٢٣	٢٤	٢٥	٢٦
٢٣	٢٤	٢٥	٢٦	٢٧
٢٤	٢٥	٢٦	٢٧	٢٨
٢٥	٢٦	٢٧	٢٨	٢٩
٢٦	٢٧	٢٨	٢٩	٣٠

؟ VueX مع المقدرات للتعامل مع القيم

1. $\Rightarrow \text{npm i -v vuex@4.0.0} \Leftarrow \text{https://github.com/vuexjs/vuex}$

2. create your store (.mystore.js)

import {createStore} from 'vuex'

const store = createStore({

state: {

counter: 0

},

getters: {

counter(state){

return state.counter;

}

},

mutations: {

increment(state, payload){

state.counter += payload.n;

}

},

actions: {

increment(commit){

commit('increment');

}

});

export default store;

١٢
12

١١ ذوالحجّة ١٤٣٧

٢ تقویت ١٤٣٧

September

Monday

2016

٢٠١٦

3- in main.js

```
import store from 'store.js';
createApp(App).use(store).mount('#app');
```

4- in your component:

لـ `this.$store.state` لـ `this.$store.getters` الـ `increment` `state` `getters`
 $\rightarrow this.$store.getters.counter;$

commit `this.$store.commit('increment')` لـ `mutation`
 $\rightarrow this.$store.commit('increment', { data: 1 })$ `data` دو فریز

action `this.$store.dispatch('increment')`
 $\rightarrow this.$store.dispatch('increment')$

الـ `action` و الـ `mutation` بـ `Vuex`

الـ `action` \rightarrow عمل لكن :
 يـ `synchronous code` لـ `mutation` \rightarrow لا
 يـ `synchronous` ممكن بـ `mutation` \rightarrow action \rightarrow
 asynchronous logic \rightarrow يمكن لـ `mutation` لـ `asynchronous`
 action \rightarrow يـ `action`

Name

Tel.

Email

سبتمبر ٢٠١٦

٢٢	٢٣	٢٤	٢٥
٢٣	٢٤	٢٥	٢٦
٢٤	٢٥	٢٦	٢٧
٢٥	٢٦	٢٧	٢٨
٢٦	٢٧	٢٨	٢٩
٢٧	٢٨	٢٩	٣٠

September 2016 سبتمبر ٢٠١٦

Sunday الأحد

سبتمبر

الاحد

11

١٠ الجمعة
٣٣٣
١٣٣

إيرج الخزن من Adexunex

الخزن منه تغيرات حول التفاعل مع الـ global state application state بتاته او component store هو مكان واحد ينبع منها كل الـ props او events لـ store .

