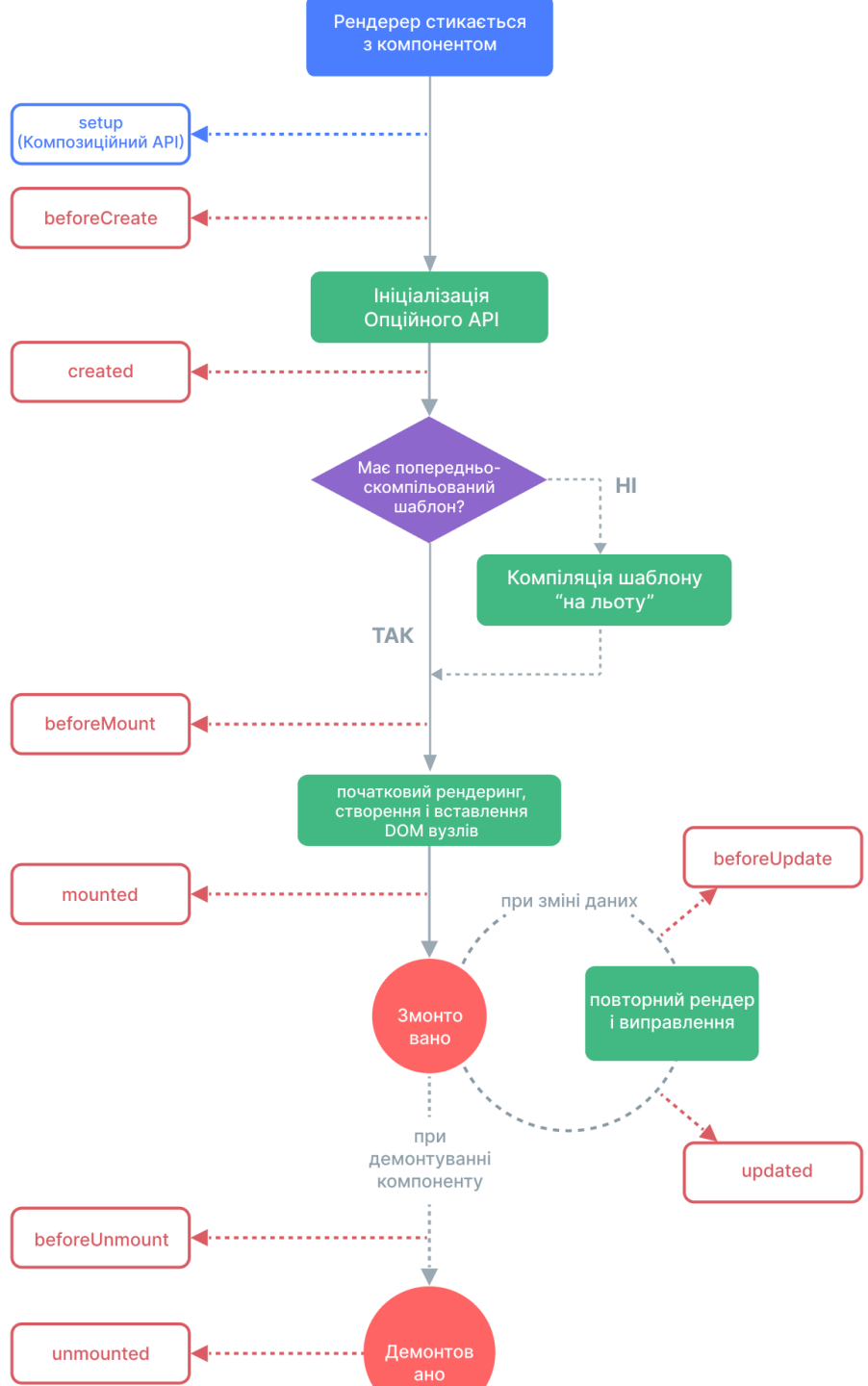
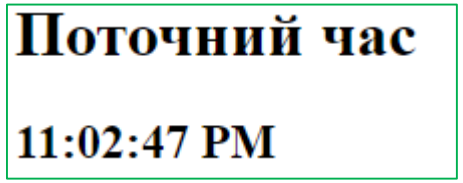
Хуки життєвого циклу

В процесі створення та функціонування компонентів Vue він проходить певні етапи (по аналогії з елементами, для яких можуть наставати певні події) і при їх настанні викликаються відповідні функції обробники (їх називають хуками життєвого циклу). Описуться після ***methods***

Хук життєвого циклу	Опис
beforeCreate	Викликається під час ***ініціалізації екземпляра*** (перед обробкою інших параметрів, таких як data() або computed)
created	Викликається після того, як ***екземпляр завершив обробку всіх параметрів, пов'язаних зі станом*** (реактивні дані, обчислювані властивості, методи та спостерігачі)
beforeMount	Викликається безпосередньо ***перед монтуванням компонента*** (вузли DOM ще не створено, він збирається вперше виконати свій ефект рендерингу DOM)
mounted	Викликається ***після монтування компонента*** (коли є потреба доступу до відрендереного DOM компонента)
beforeUpdate	Викликається безпосередньо перед тим, як ***компонент збирається оновити*** своє дерево DOM через реактивну зміну стану (для доступу до стану DOM до того, як Vue оновить DOM)
updated	Викликається після того, як ***компонент оновив своє дерево*** DOM через реактивну зміну стану
beforeUnmount	Викликається ***безпосередньо перед демонтуванням*** екземпляра компонента (екземпляр компонента все ще має повну функціональність)
unmounted	Викликається після того, як ***компонент було демонтовано*** (Використовуйте цей хук, щоб очистити створені вручну побічні ефекти, такі як таймери, слухачі подій DOM або підключення до сервера)



Приклад. **Після створення** запустити таймер для відображення поточного часу



Приклад. **Після створення** запустити таймер для відображення поточного часу

Використаємо хук життєвого цилку ***created***

```
<div id="app">
    <h1>Поточний час</h1>
    <h2>{{currentTime}}</h2>
</div>
```

Поточний час

11:02:47 PM

Приклад. **Після створення** запустити таймер для відображення поточного часу

Використтаємо хук життєвого цилку ***created***

```
<div id="app">
    <h1>Поточний час</h1>
    <h2>{{currentTime}}</h2>
</div>
```

Поточний час

11:02:47 PM

```
const app = createApp({
    data() {
        return {
            currentTime: null,
        }
    },

    methods: {
        setCurrentTime() {
            this.currentTime = new Date().toLocaleTimeString()
        },
    },

    created() {
        setInterval(() => {
            this.setCurrentTime()
        }, 1000)
    },
}).mount('#app')
```

Приклад. **Після створення** зробити запит на сервер для завантаження даних

```html
<div id="app">
  <div v-if="isLoading">... Завнтаження ...</div>
  <div v-else v-for="item in resData">
    <hr />
    <h1>{{item.attributes.titles.en}}</h1>
    <img :src="item.attributes.posterImage.medium" />
    <div>{{item.attributes.description}}</div>
  </div>
</div>
```

```js
const app = createApp({
  data() {
    return {
      resData: null,
      isLoading: true,
    }
  },

  methods: {
    loadData() {
      fetch(
        'https://kitsu.io/api/edge/anime?page[limit]=5&page[offset]=0'
      )
        .then((response) => response.json())
        .then((resData) => {
          console.log(resData)
          this.resData = resData.data
          this.isLoading = false
        })
    },
  },

  created() {
    this.loadData()
  },
}).mount('#app')
```

Приклад. На сайті робиться перевірка віку користувача. **Після завантаження сайту** встановити фокус у поле введення віку.

Приклад. На сайті робиться перевірка віку користувача. **Після завантаження сайту** встановити фокус у поле введення віку.

Використтаємо хук життєвого цилку ***mounted***

```
<div id="app">
  <div>
    <label>
      Введіть вік :
      <input ref="age" type="number" v-model="age" />
    </label>
  </div>

  <button @click="onStart">Go</button>

  <div v-if="message" class="error">{{message}}</div>
</div>
```

Введіть вік :

Go

Приклад. На сайті робиться перевірка віку користувача. **Після завантаження сайту** встановити фокус у поле введення віку.

Використаємо хук життєвого цилку ***mounted***

```
<div id="app">
  <div>
    <label>
      Введіть вік :
      <input ref="age" type="number" v-model="age" />
    </label>
  </div>

  <button @click="onStart">Go</button>

  <div v-if="message" class="error">{{message}}</div>
</div>
```

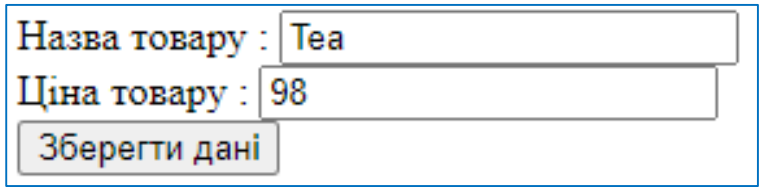
```
const app = createApp({
  data() {
    return {
      age: null,
      message: null,
    }
  },

  methods: {
    onStart() {
      if (!this.age) {
        this.$refs.age.focus()
        this.message = 'Введіть вік'
      } else {
        if (this.age < 18) this.message = 'Вхід заборонено'
        else this.message = 'Вхід дозволено'
      }
    },
  },

  mounted() {
    this.$refs.age.focus()
  },
}).mount('#app')
```

Введіть вік :

Go

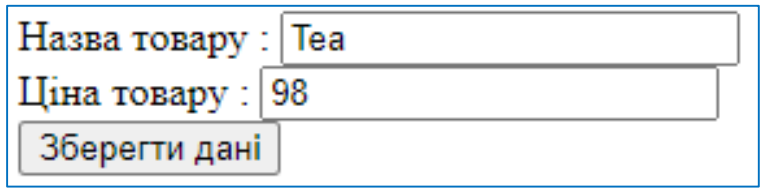
Приклад. Зберігаємо та продовжуємо редагувати дані. На сайті користувач вводить назву товару та кількість. Перед виході з сайту зберегти дані у localSrorage. А при заході на сайт – відновлювати дані з localSrorage.

```
<div id="app">
    <div>
        <label>
            Назва товару :
            <input type="text" v-model="product.title" />
        </label>
    </div>
    <div>
        <label>
            Ціна товару :
            <input type="number" v-model="product.price" />
        </label>
    </div>
    <button @click="saveData">Зберегти дані</button>
</div>
```



Приклад. Зберігаємо та продовжуємо редагувати дані. На сайті користувач вводить назву товару та кількість. Перед виході з сайту зберегти дані у localSrorage. А при заході на сайт – відновлювати дані з localSrorage.

```html
<div id="app">
  <div>
    <label>
      Назва товару :
      <input type="text" v-model="product.title" />
    </label>
  </div>
  <div>
    <label>
      Ціна товару :
      <input type="number" v-model="product.price" />
    </label>
  </div>
  <button @click="saveData">Зберегти дані</button>
</div>
```




```js
const app = createApp({
  data() {
    return {
      product: {
        title: null,
        price: null,
      },
    }
  },

  methods: {
    saveData() {
      localStorage.setItem('product', JSON.stringify(this.product))
    },
    loadData() {
      console.log('loadData')
      if (localStorage.getItem('product'))
        this.product = JSON.parse(localStorage.getItem('product'))
    },
  },

  mounted() {
    this.loadData()
  },
}).mount('#app')
```