Proxy

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Proxy

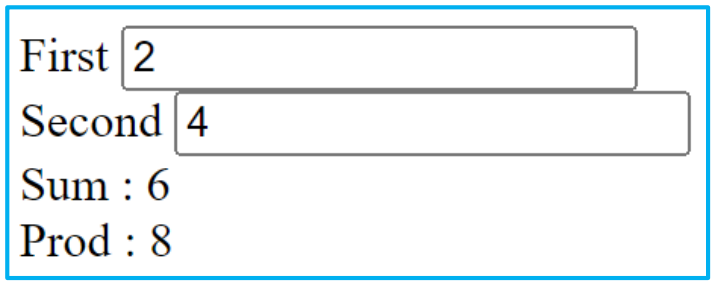
https://uk.javascript.info/proxy

Proxy

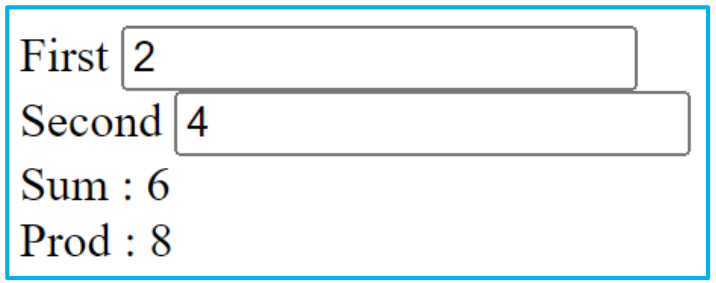
- Об’єкт Proxy обгортає інший об’єкт і перехоплює операції, такі як читання/запис властивостей та інші, за бажанням обробляючи їх самостійно, або прозоро дозволяючи об’єкту обробляти їх.
- Проксі використовуються в багатьох бібліотеках і деяких фреймворках браузера.

https://uk.javascript.info/proxy

Розглянемо приклад знаходження суми та добутку за введеними двома числами



Розглянемо приклад знаходження суми та добутку за введеними двома числами

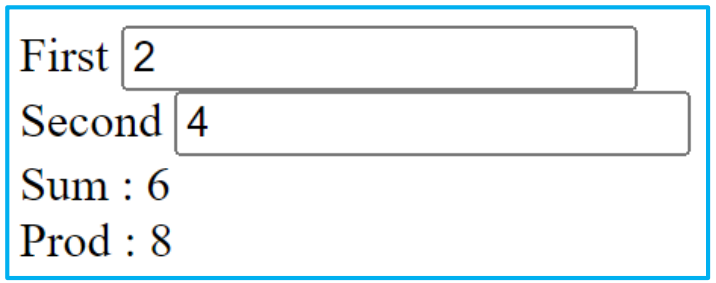


Використовуючи прості засоби HTML та JS

```
<div>
  <label>
    First
    <input type="number" id="num1" value="0" onchange="changeNum1()" />
  </label>
</div>
<div>
  <label>
    Second
    <input type="number" id="num2" value="0" onchange="changeNum2()" />
  </label>
</div>
<div>Sum : <span id="sum">0</span></div>
<div>Prod : <span id="prod">0</span></div>
```

```
<script>
  let data = {
    num1: 0,
    num2: 0,
  }
  function calcResults() {
    const sum = data.num1 + data.num2
    const prod = data.num1 * data.num2
    document.getElementById('sum').innerText = sum
    document.getElementById('prod').innerText = prod
  }
  function changeNum1() {
    let num1Val = parseInt(document.getElementById('num1').value)
    data.num1 = num1Val
    calcResults()
  }
  function changeNum2() {
    let num2Val = parseInt(document.getElementById('num2').value)
    data.num2 = num2Val
    calcResults()
  }
</script>
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами



Використовуючи прості засоби HTML та JS

```html
<div>
  <label>
    First
    <input type="number" id="num1" value="0" onchange="changeNum1()" />
  </label>
</div>
<div>
  <label>
    Second
    <input type="number" id="num2" value="0" onchange="changeNum2()" />
  </label>
</div>
<div>Sum : <span id="sum">0</span></div>
<div>Prod : <span id="prod">0</span></div>
```

```html
<script>
  let data = {
    num1: 0,
    num2: 0,
  }
  function calcResults() {
    const sum = data.num1 + data.num2
    const prod = data.num1 * data.num2
    document.getElementById('sum').innerText = sum
    document.getElementById('prod').innerText = prod
  }
  function changeNum1() {
    let num1Val = parseInt(document.getElementById('num1').value)
    data.num1 = num1Val
    calcResults()
  }
  function changeNum2() {
    let num2Val = parseInt(document.getElementById('num2').value)
    data.num2 = num2Val
    calcResults()
  }
</script>
```

Використовуючи Proxy

```
let data = {
  num1: 0,
  num2: 0,
}
```

Використовуючи Proxy

Загортаємо у proxy об’єкт

```
let data = {
  num1: 0,
  num2: 0,
}

data = new Proxy(data, {
  get(target, prop) {
    return target[prop]
  },
  set(target, prop, val) {
    target[prop] = parseInt(val)
    calcResults()
  },
})
```

Використовуючи Proxy

Загортаємо у proxy об’єкт

```
let data = {
  num1: 0,
  num2: 0,
}

data = new Proxy(data, {
  get(target, prop) {
    return target[prop]
  },
  set(target, prop, val) {
    target[prop] = parseInt(val)
    calcResults()
  },
})
```

let val = data . num1

Використовуючи Proxy

Загортаємо у proxy об’єкт

```
let data = {
  num1: 0,
  num2: 0,
}

data = new Proxy(data, {
  get(target, prop) {
    return target[prop]
  },
  set(target, prop, val) {
    target[prop] = parseInt(val)
    calcResults()
  },
})
```

При зчитуванні значення викликається метод get

```
let val = data.num1
```

Використовуючи Proxy

Загортаємо у proxy об’єкт

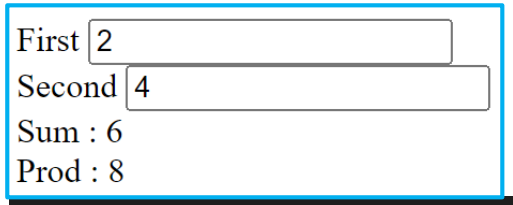
```
let data = {
  num1: 0,
  num2: 0,
}

data = new Proxy(data, {
  get(target, prop) {
    return target[prop]
  },
  set(target, prop, val) {
    target[prop] = parseInt(val)
    calcResults()
  },
})
```

При встановленні нового значення викликається метод set

```
data . num1 = 7
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами



Використовуючи Proxy

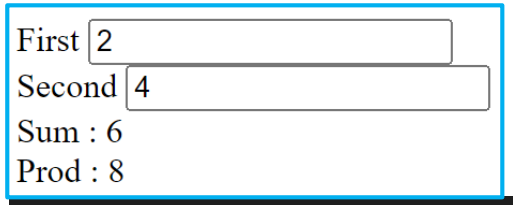
```
<div>
  <label>
    First
    <input
      type="number"
      id="num1"
      value="0"
      onchange="data.num1=this.value"
    />
  </label>
</div>
<div>
  <label>
    Second
    <input
      type="number"
      id="num2"
      value="0"
      onchange="data.num2=this.value"
    />
  </label>
</div>
<div>Sum : <span id="sum">0</span></div>
<div>Prod : <span id="prod">0</span></div>
```

```
<script>
  let data = {
    num1: 0,
    num2: 0,
  }

  data = new Proxy(data, {
    get(target, prop) {
      return target[prop]
    },
    set(target, prop, val) {
      target[prop] = parseInt(val)
      calcResults()
    },
  })

  function calcResults() {
    const sum = data.num1 + data.num2
    const prod = data.num1 * data.num2
    document.getElementById('sum').innerText = sum
    document.getElementById('prod').innerText = prod
  }
</script>
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами



Використовуючи Proxy

```
<div>
  <label>
    First
    <input
      type="number"
      id="num1"
      value="0"
      onchange="data.num1=this.value"
    />
  </label>
</div>
<div>
  <label>
    Second
    <input
      type="number"
      id="num2"
      value="0"
      onchange="data.num2=this.value"
    />
  </label>
</div>
<div>Sum : <span id="sum">0</span></div>
<div>Prod : <span id="prod">0</span></div>
```

Змінюючи поле ***data.num1*** викликається ***set***

```
<script>
  let data = {
    num1: 0,
    num2: 0,
  }

  data = new Proxy(data, {
    get(target, prop) {
      return target[prop]
    },
    set(target, prop, val) {
      target[prop] = parseInt(val)
      calcResults()
    },
  })

  function calcResults() {
    const sum = data.num1 + data.num2
    const prod = data.num1 * data.num2
    document.getElementById('sum').innerText = sum
    document.getElementById('prod').innerText = prod
  }
</script>
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Proxy

```
<div>
  <label>
    First
    <input
      type="number"
      id="num1"
      value="0"
      onchange="data.num1=this.value"
    />
  </label>
</div>
<div>
  <label>
    Second
    <input
      type="number"
      id="num2"
      value="0"
      onchange="data.num2=this.value"
    />
  </label>
</div>
<div>Sum : <span id="sum">0</span></div>
<div>Prod : <span id="prod">0</span></div>
```

```
<script>
  let data = {
    num1: 0,
    num2: 0,
  }

  data = new Proxy(data, {
    get(target, prop) {
      return target[prop]
    },
    set(target, prop, val) {
      target[prop] = parseInt(val)
      calcResults()
    },
  })

  function calcResults() {
    const sum = data.num1 + data.num2
    const prod = data.num1 * data.num2
    document.getElementById('sum').innerText = sum
    document.getElementById('prod').innerText = prod
  }
</script>
```

При зміні поля *data.num1* викликається *calcResult ( )*

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Proxy

```
<div>
  <label>
    First
    <input
      type="number"
      id="num1"
      value="0"
      onchange="data.num1=this.value"
    />
  </label>
</div>
<div>
  <label>
    Second
    <input
      type="number"
      id="num2"
      value="0"
      onchange="data.num2=this.value"
    />
  </label>
</div>
<div>Sum : <span id="sum">0</span></div>
<div>Prod : <span id="prod">0</span></div>
```

```
<script>
  let data = {
    num1: 0,
    num2: 0,
  }

  data = new Proxy(data, {
    get(target, prop) {
      return target[prop]
    },
    set(target, prop, val) {
      target[prop] = parseInt(val)
      calcResults()
    },
  })

  function calcResults() {
    const sum = data.num1 + data.num2
    const prod = data.num1 * data.num2
    document.getElementById('sum').innerText = sum
    document.getElementById('prod').innerText = prod
  }
</script>
```

Отже об'єкт data є реактивним, тобто при зміні його полів виконуємо деякі потрібні операції

При зміні поля *data.num1* викликається *calcResult ( )*

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Vue

```
<script>
  const { createApp } = Vue

  const app = createApp({
    data() {
      return {
        num1: 0,
        num2: 0,
      }
    },
    methods: {
      onClear() {
        this.num1 = 0
        this.num2 = 0
      },
    },
  })

  app.mount('#myApp')
</script>
```

Описуємо моделі даних

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Vue

```
<div id="myApp">
    <div>
        <label>
            First
            <input type="number" id="num1" v-model="num1" />
        </label>
    </div>
    <div>
        <label>
            Second
            <input type="number" id="num2" v-model="num2" />
        </label>
    </div>
    <div>Sum : <span id="sum"> {{ num1+num2 }} </span></div>
    <div>Prod : <span id="prod">{{ num1*num2 }}</span></div>
    <button @click="onClear">Clear</button>
</div>
```

Поєднуємо моделі даних з елементами розмітки

```
<script>
    const { createApp } = Vue

    const app = createApp({
        data() {
            return {
                num1: 0,
                num2: 0,
            }
        },
        methods: {
            onClear() {
                this.num1 = 0
                this.num2 = 0
            },
        },
    })

    app.mount('#myApp')
</script>
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Vue

Моделі даних є рективними

```
<div id="myApp">
    <div>
        <label>
            First
            <input type="number" id="num1" v-model="num1" />
        </label>
    </div>
    <div>
        <label>
            Second
            <input type="number" id="num2" v-model="num2" />
        </label>
    </div>
    <div>Sum : <span id="sum"> {{ num1+num2 }} </span></div>
    <div>Prod : <span id="prod">{{ num1*num2 }}</span></div>
    <button @click="onClear">Clear</button>
</div>
```

Поєднуємо моделі даних з елементами розмітки

```
<script>
    const { createApp } = Vue

    const app = createApp({
        data() {
            return {
                num1: 0,
                num2: 0,
            }
        },
        methods: {
            onClear() {
                this.num1 = 0
                this.num2 = 0
            },
        },
    })

    app.mount('#myApp')
</script>
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Vue

Моделі даних є рективними

```
<div id="myApp">
    <div>
        <label>
            First
            <input type="number" id="num1" v-model="num1" />
        </label>
    </div>
    <div>
        <label>
            Second
            <input type="number" id="num2" v-model="num2" />
        </label>
    </div>
    <div>Sum : <span id="sum"> {{ num1+num2 }} </span></div>
    <div>Prod : <span id="prod">{{ num1*num2 }}</span></div>
    <button @click="onClear">Clear</button>
</div>
```

1.При зміні значення у input значння передається у об'єкт

```
<script>
    const { createApp } = Vue

    const app = createApp({
        data() {
            return {
                num1: 0,
                num2: 0,
            }
        },
        methods: {
            onClear() {
                this.num1 = 0
                this.num2 = 0
            },
        },
    })

    app.mount('#myApp')
</script>
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Vue

Моделі даних є рективними

```
<div id="myApp">
    <div>
        <label>
            First
            <input type="number" id="num1" v-model="num1" />
        </label>
    </div>
    <div>
        <label>
            Second
            <input type="number" id="num2" v-model="num2" />
        </label>
    </div>
    <div>Sum : <span id="sum"> {{ num1+num2 }} </span></div>
    <div>Prod : <span id="prod">{{ num1*num2 }}</span></div>
    <button @click="onClear">Clear</button>
</div>
```

2.При зміні значення поля num1 його значння передається у input

```
<script>
    const { createApp } = Vue

    const app = createApp({
        data() {
            return {
                num1: 0,
                num2: 0,
            }
        },
        methods: {
            onClear() {
                this.num1 = 0
                this.num2 = 0
            },
        },
    })

    app.mount('#myApp')
</script>
```

Розглянемо приклад знаходження суми та добутку за введеними двома числами

First 2
Second 4
Sum : 6
Prod : 8

Використовуючи Vue

Моделі даних є рективними

```
<div id="myApp">
    <div>
        <label>
            First
            <input type="number" id="num1" v-model="num1" />
        </label>
    </div>
    <div>
        <label>
            Second
            <input typ
        </label>
    </div>
    <div>Sum : <span id="sum">{{ num1+num2 }}</span></div>
    <div>Prod : <span id="prod">{{ num1*num2 }}</span></div>
    <button @click="onClear">Clear</button>
</div>
```

3.При зміні моделей даних (num1 або num2) оновлюються інші залежні елементи розмітки

```
<script>
    const { createApp } = Vue

    const app = createApp({
        data() {
            return {
                num1: 0,
                num2: 0,
            }
        },
        methods: {
            onClear() {
                this.num1 = 0
                this.num2 = 0
            },
        },
    })

    app.mount('#myApp')
</script>
```