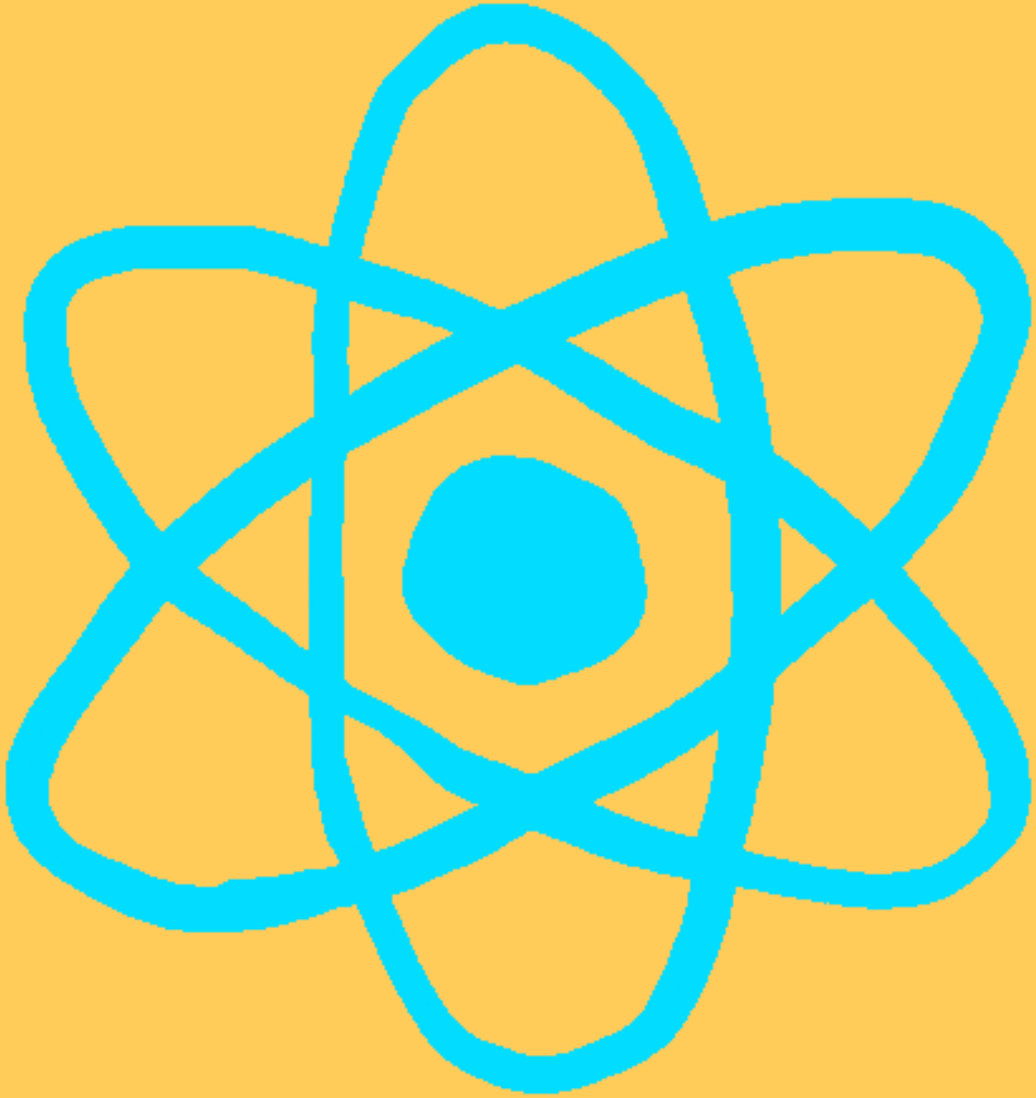
FRONTEND. REACT. ЛЕКЦІЯ 6

REACT FORMS, YUP, REACT-FORM-HOOK.

ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

BEER CS



ФОРМИ

ФОРМИ

- Базові форми

ФОРМИ

- Базові форми
- React Hook Form

Форми

- Базові форми
- React Hook Form
- Валідація з Yup

БАЗОВІ ФОРМИ

Керовані компоненти

```
import { useState } from 'react';

export function Form() {
        const [value, setValue] = useState(0);

        const handleSubmit = (event) => {
                console.log('submit value', value);
                event.preventDefault();
        }

        const handleChange = (event) => {
                setValue(event.target.value);
        }

        return (<form onsubmit="{handleSubmit}">
                <label>
                Ім'я:
                <input type="text" value="{value}" onchange="{handleChange}">
```

БАЗОВІ ФОРМИ

Керовані компоненти

```
5
6           const handleSubmit = (event) => {
7                   console.log('submit value', value);
8                   event.preventDefault();
9           }
10
11          const handleChange = (event) => {
12                  setValue(event.target.value);
13          }
14
15          return (<form onsubmit="{handleSubmit}">
16                  <label>
17                  Ім'я:
18                  <input type="text" value="{value}" onchange="{handleChange}">
19                  </label>
20                  <input type="submit" value="Надіслати">
21          </form>);
22  }
```

БАЗОВІ ФОРМИ

Керовані компоненти

```
import { useState } from 'react';

export function Form() {
        const [value, setValue] = useState(0);

        const handleSubmit = (event) => {
                console.log('submit value', value);
                event.preventDefault();
        }

        const handleChange = (event) => {
                setValue(event.target.value);
        }

        return (<form onsubmit="{handleSubmit}">
                <label>
                Ім'я:
                <input type="text" value="{value}" onchange="{handleChange}">
```

БАЗОВІ ФОРМИ

Керовані компоненти

```
export function Form() {
        const [value, setValue] = useState(0);

        const handleSubmit = (event) => {
                console.log('submit value', value);
                event.preventDefault();
        }

        const handleChange = (event) => {
                setValue(event.target.value);
        }

        return (<form onsubmit="{handleSubmit}">
                <label>
                Ім'я:
                <input type="text" value="{value}" onchange="{handleChange}">
                </label>
```

БАЗОВІ ФОРМИ

Керовані компоненти

```
import { useState } from 'react';

export function Form() {
        const [value, setValue] = useState(0);

        const handleSubmit = (event) => {
                console.log('submit value', value);
                event.preventDefault();
        }

        const handleChange = (event) => {
                setValue(event.target.value);
        }

        return (<form onsubmit="{handleSubmit}">
                <label>
                Ім'я:
                <input type="text" value="{value}" onchange="{handleChange}">
```

БАЗОВІ ФОРМИ

Керовані компоненти

```
1  import { useState } from 'react';
2
3  export function Form() {
4          const [value, setValue] = useState(0);
5
6          const handleSubmit = (event) => {
7                  console.log('submit value', value);
8                  event.preventDefault();
9          }
10
11         const handleChange = (event) => {
12                 setValue(event.target.value);
13         }
14
15         return (<form onsubmit="{handleSubmit}">
16                 <label>
17                 Ім'я:
18                 <input type="text" value="{value}" onchange="{handleChange}">
```

КЕРОВАНІ КОМПОНЕНТИ

КЕРОВАНІ КОМПОНЕНТИ

- Єдине джерело правди

КЕРОВАНІ КОМПОНЕНТИ

- Єдине джерело правди
- Зберігають данні в стейті

КЕРОВАНІ КОМПОНЕНТИ

- Єдине джерело правди
- Зберігають данні в стейті
- Єдиний спосіб роботи

КЕРОВАНІ КОМПОНЕНТИ

- Єдине джерело правди
- Зберігають данні в стейті
- Єдиний спосіб роботи
- Зміна стану ЛИШЕ через onChange

КЕРОВАНІ КОМПОНЕНТИ

Textarea

```
export function Form() {
        return (<textarea value="{value}" onchange="{handleChange}"></textarea>
        );
}
```

КЕРОВАНІ КОМПОНЕНТИ

Select

```
export function Form() {
        return ( <select value="{value}" onchange="{handleChange}">
                <option value="грейпфрут">Грейпфрут</option>
                <option value="лайм">Лайм</option>
                <option value="кокос">Кокос</option>
                <option value="манго">Манго</option>
          </select>);
}
```

НЕКЕРОВАНІ КОМПОНЕНТИ

```
import { useRef } from 'react';

export function FormUncontrol() {
        const handleSubmit = (event) => {
                console.log('submit value', input.current.value);
                event.preventDefault();
        }

        const input = useRef();

        return (<form onsubmit="{handleSubmit}">
                <label>
                Ім'я:
                <input type="text" ref="{input}">
                </label>
                <input type="submit" value="Надіслати">
        </form>);
}
```

НЕКЕРОВАНІ КОМПОНЕНТИ

```
1  import { useRef } from 'react';
2
3  export function FormUncontrol() {
4          const handleSubmit = (event) => {
5                  console.log('submit value', input.current.value);
6                  event.preventDefault();
7          }
8
9          const input = useRef();
10
11         return (<form onsubmit="{handleSubmit}">
12                 <label>
13                 Ім'я:
14                 <input type="text" ref="{input}">
15                 </label>
16                 <input type="submit" value="Надіслати">
17         </form>);
18 }
```

НЕКЕРОВАНІ КОМПОНЕНТИ

```
1  import { useRef } from 'react';
2
3  export function FormUncontrol() {
4          const handleSubmit = (event) => {
5                  console.log('submit value', input.current.value);
6                  event.preventDefault();
7          }
8
9          const input = useRef();
10
11         return (<form onsubmit="{handleSubmit}">
12                 <label>
13                 Ім'я:
14                 <input type="text" ref="{input}">
15                 </label>
16                 <input type="submit" value="Надіслати">
17         </form>);
18 }
```

НЕКЕРОВАНІ КОМПОНЕНТИ

```
import { useRef } from 'react';

export function FormUncontrol() {
        const handleSubmit = (event) => {
                console.log('submit value', input.current.value);
                event.preventDefault();
        }

        const input = useRef();

        return (<form onsubmit="{handleSubmit}">
                <label>
                Ім'я:
                <input type="text" ref="{input}">
                </label>
                <input type="submit" value="Надіслати">
        </form>);
}
```

НЕКЕРОВАНІ КОМПОНЕНТИ

НЕКЕРОВАНІ КОМПОНЕНТИ

- Ненайкраще рішення

НЕКЕРОВАНІ КОМПОНЕНТИ

- Ненайкраще рішення
- Використовуються з типом файл

А В ЧОМУ ПРОБЛЕМА?

А В ЧОМУ ПРОБЛЕМА?

- Складні форми

А В ЧОМУ ПРОБЛЕМА?

- Складні форми
- Повторюваність

А в чому проблема?

- Складні форми
- Повторюваність
- Валідація

БІБЛІОТЕКИ

БІБЛІОТЕКИ

- React final form

БІБЛІОТЕКИ

- React final form
- yup

БІБЛІОТЕКИ

- React final form
- yup
- react-hook-form

БІБЛІОТЕКИ

- React final form
- yup
- react-hook-form
- Formik

REACT-HOOK-FORM

https://www.npmjs.com/package/react-hook-form

REACT-HOOK-FORM

npm i react-hook-form

REACT-HOOK-FORM

```
import React from 'react';
import { useForm } from 'react-hook-form';

export function ReactHookForm() {
const { register, handleSubmit, formState: { errors } } = useForm();
const onSubmit = data => console.log(data);
console.log(errors);

return (
        <form onsubmit="{handleSubmit(onSubmit)}">
        <input type="text" placeholder="First name" {...register("first="" name",=
        <input type="text" placeholder="Last name" {...register("last="" name",=""
        <input type="text" placeholder="Email" {...register("email",="" {required:
        <input type="tel" placeholder="Mobile number" {...register("mobile="" numb
        <select {...register("title",="" {="" required:="" true="" })}="">
                <option value="Mr">Mr</option>
                <option value="Mrs">Mrs</option>
                <option value="Miss">Miss</option>
```

REGISTER FIELDS

https://react-hook-form.com/get-started#Registerfields

Register fields

One of the key concepts in React Hook Form is to **`register`** your component into the hook. This will make its value available for both the form validation and submission.

Note: Each field is **required** to have a `name` as a key for the registration process.

Menu

- </> Quick start
- </> React Web Video Tutorial
- </> Register fields
- </> Apply validation
- </> Integrating an existing form
- </> Integrating with UI libraries
- </> Integrating Controlled Inputs

JS TS CODESANDBOX JS

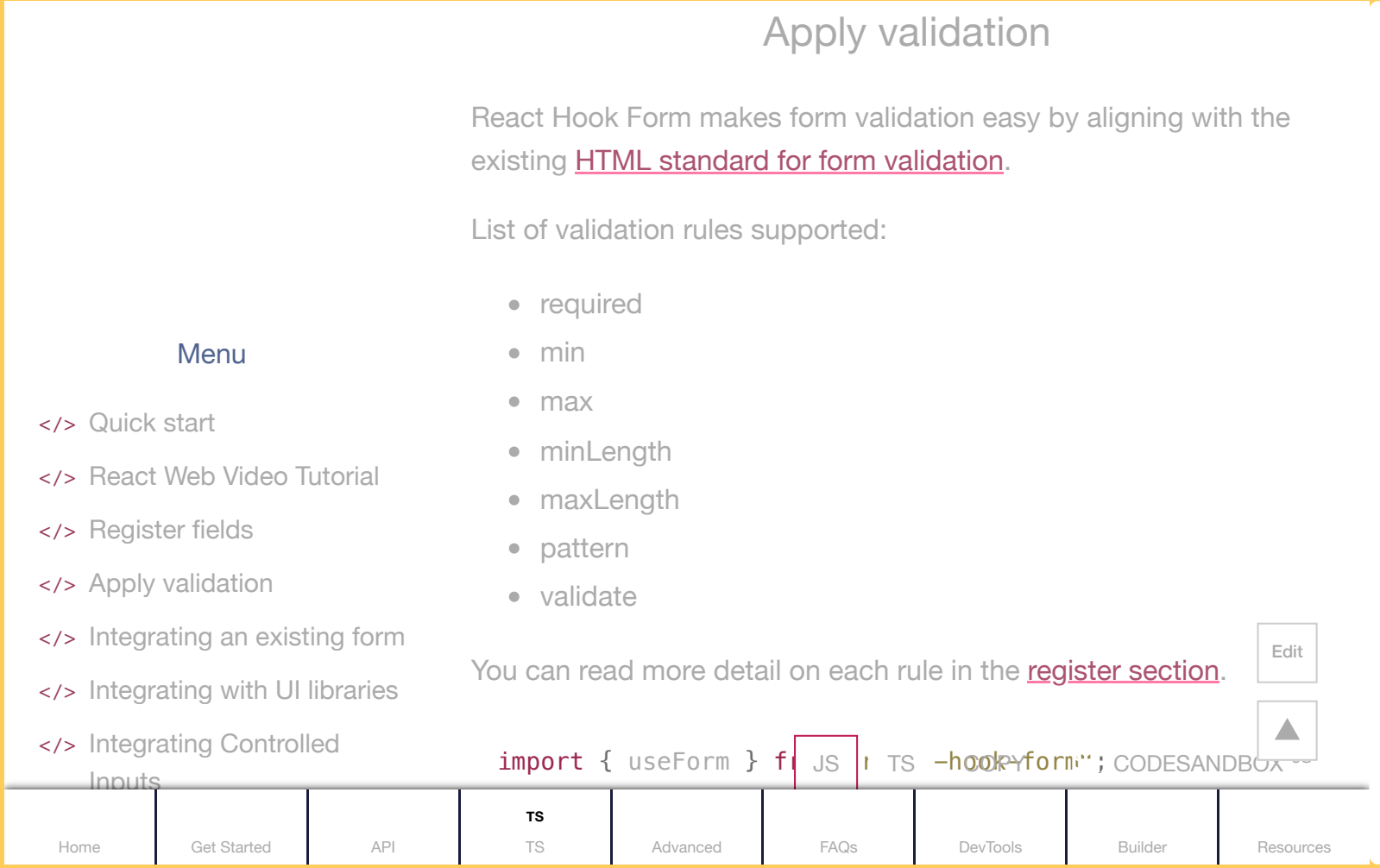
```
import { useForm } from "react-hook-form";

export default function App() {
  const { register, handleSubmit } = useForm();
  const onSubmit = data => console.log(data);

  return (
    <form onSubmit={handleSubmit(onSubmit)}>
      <input {...register("firstName")} />
      <select {...register("gender")}>
        <option value="female">female</option>
```

Edit

Home | Get Started | API | TS | Advanced | FAQs | DevTools | Builder | Resources

VALIDATION

https://react-hook-form.com/get-started#Applyvalidation



Apply validation

React Hook Form makes form validation easy by aligning with the existing HTML standard for form validation.

List of validation rules supported:

- required
- min
- max
- minLength
- maxLength
- pattern
- validate

You can read more detail on each rule in the register section.

Edit

Menu

</> Quick start
</> React Web Video Tutorial
</> Register fields
</> Apply validation
</> Integrating an existing form
</> Integrating with UI libraries
</> Integrating Controlled Inputs

import { useForm } f JS TS -hook-form"; CODESANDBOX

Home | Get Started | API | TS | Advanced | FAQs | DevTools | Builder | Resources

HANDLE ERRORS

https://react-hook-form.com/get-started#Handleerrors

Handle errors

React Hook Form provides an `errors` object to show you the errors in the form. `errors`' type will return given validation constraints. The following example showcases a required validation rule.

JS TS CODESANDBOX JS

```
import { useForm } f -hook-form";

export default function App() {
  const { register, formState: { errors }, handleSubmit
  const onSubmit = (data) => console.log(data);

  return (
    <form onSubmit={handleSubmit(onSubmit)}>
      <input
        {...register("firstName", { required: true })}
        aria-invalid={errors.firstName ? "true" : "fa
      />
      {errors.firstName?.type === 'required' && <p ro
```

Edit

Menu

- </> Quick start
- </> React Web Video Tutorial
- </> Register fields
- </> Apply validation
- </> Integrating an existing form
- </> Integrating with UI libraries
- </> Integrating Controlled Inputs

Home | Get Started | API | TS | Advanced | FAQs | DevTools | Builder | Resources

ВАЛІДАЦІЯ

ВАЛІДАЦІЯ

- Асинхронна валідація

ВАЛІДАЦІЯ

- Асинхронна валідація
- Кастомна валідація

ВАЛІДАЦІЯ

- Асинхронна валідація
- Кастомна валідація
- Вкладені об’єкти

CODE EXAMPLES

https://github.com/react-hook-form/react-hook-form/tree/master/examples

YUP

YUP

- Гнучна кастомна валідація

YUP

- Гнучна кастомна валідація
- Велика кількість хелперів

YUP

- Гнучна кастомна валідація
- Велика кількість хелперів
- Перевикористання валідації

YUP

https://github.com/jquense/yup

WO? WO?