

# Adding Form Inputs and listening inputs

## Adding Form Inputs

1. div will hold all inputs.
2. We are having three input fields Title, amount and date using date picker.
3. we can set min and max values these are just default HTML attributes for input elements which control how this element can be used.
4. A button to submit the form.

NOTE : we have created 'ExpenseForm' as another component and will call it in 'NewExpense' that is another component and in last will combine all the component in App component.

```
// ExpenseForm
return (
  <form >
    <div className="new-expense__controls">
      <div className="new-expense__control">
        <label>Title</label>
        <input type="text"
          />
      </div>
      <div className="new-expense__control">
        <label>Amount</label>
        <input
          type="number"
          min="0.01"
          step="0.01"
        />
      </div>
      <div className="new-expense__control">
        <label>Date</label>
        <input
          type="date"
          min="2019-01-01"
          max="2022-12-31"
        />
      </div>
    </div>
    <div className="new-expense__actions">
      <button type="submit">Add Expense</button>
    </div>
  </form>
);
```

Note: In next component will add the expenseform using import function.

```
//NewExpense component
return(<div className="new-expense">
  <ExpenseForm />
</div>);
```

Note : In last will add the 'NewExpense' into 'App' component

```
// App Component
return (
  <div>
    <h2>Let's get started!</h2>
    <NewExpenses></NewExpenses>
    <Expenses items={expense} />
  </div>
);
```

## listening inputs

1. Listening means on every keystroke, we wanna get the value the user entered and store that somewhere.
2. Gathering user input.
3. We need to add listeners to listen, for every keystroke or for every change of these inputs. So, on the input by adding a prop starts with **"on"**.
4. We can use **onInput** and **onChange** to get the values.
5. But the advantage of the **onChange** event uses same event for all inputs types.

Note: **onChange={titleChangeHandler}** here titleChangeHandler function will store the value whenever title will change.

```
const ExpenseForm = () => {
  const titleChangeHandler = (event) => { // titleChangeHandler will store the
    value whenever title will change. "default event object".
    console.log(event.target.value);
  }

  return (
    <form>
      <div className="new-expense__controls">
        <div className="new-expense__control">
          <label>Title</label>
          <input type="text"
            value={entitledTitle}
            onChange={titleChangeHandler} /> // added listener using onChange.
        </div>
      </div>
    </form>
  );
};
```

```
)  
}
```