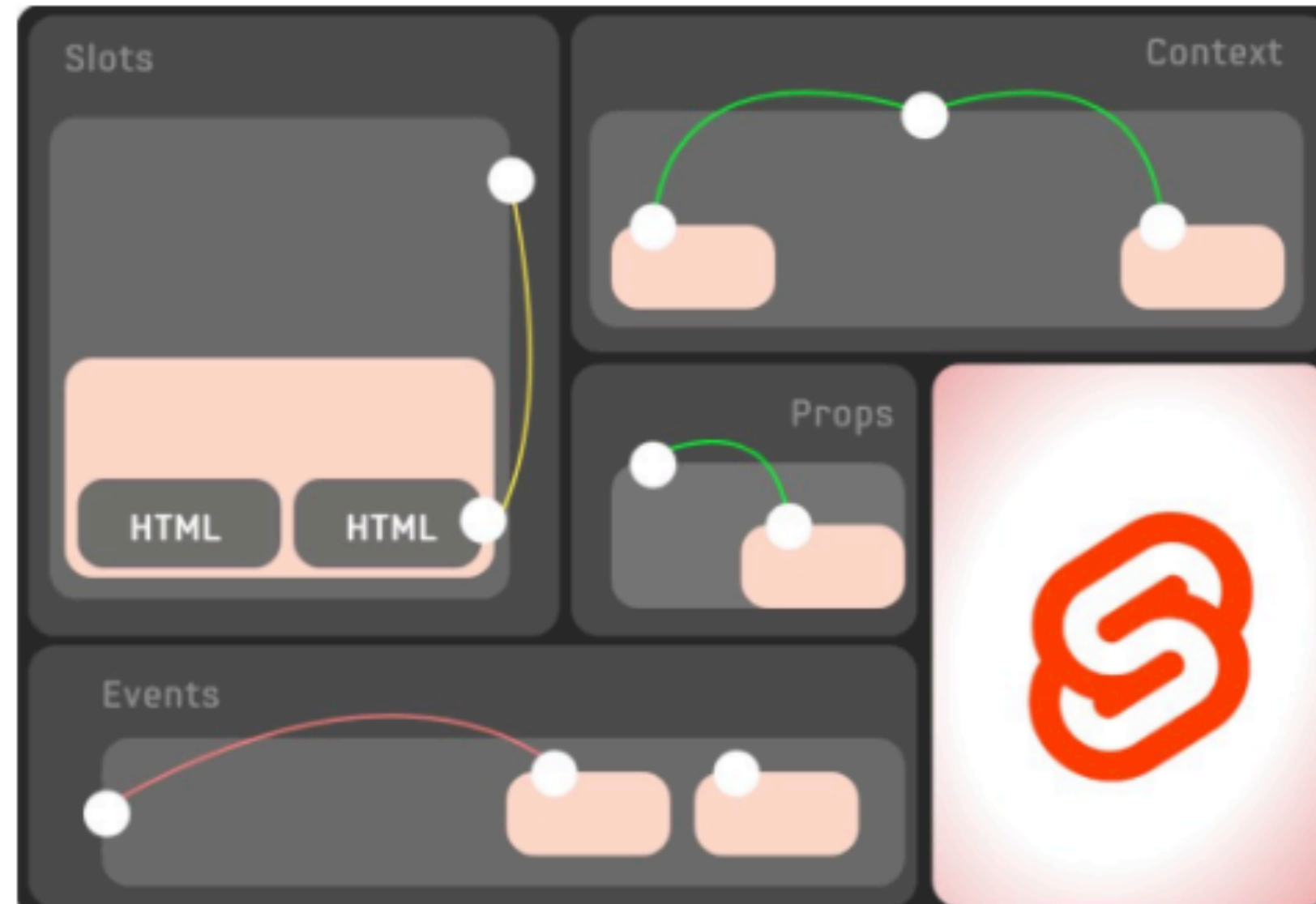


Svelte Components

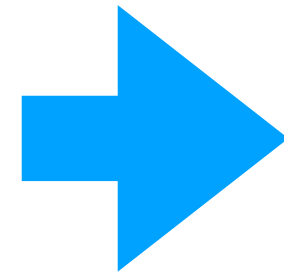


Creating and assembling
Svelte Components

Svelte Components



Factor out into
(reusable)
Svelte
components



Simple Todo List

Fun things to do

What should I do?

Add Todo

Things yet do

Task	Date	
go for a run	25/3/2022, 11:29:11	<button>delete</button>
go for a cycle	25/3/2022, 11:29:22	<button>delete</button>

Things done

Task	Date
go for a walk	25/3/2022, 11:29:16

Svelte Components

Simple Todo List

Fun things to do

What should I do?

go for a cycle

Add Todo

Things yet do

Task

Date

go for a run

25/3/2022, 11:29:11

delete

go for a cycle

25/3/2022, 11:29:22

delete

Things done

Task

Date

go for a walk

25/3/2022, 11:29:16



Title.svelte



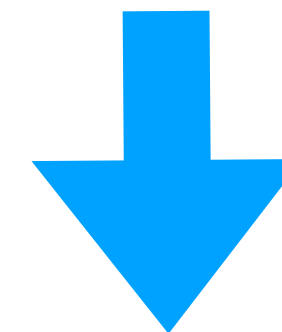
AddTodoForm.svelte



TodoList.svelte



App.svelte



Title.svelte (1)

Simple Todo List

Fun things to do

```
<div class="box has-text-centered">
  <div class="title"> Simple Todo List</div>
  <div class="subtitle">Fun things to do</div>
</div>
```

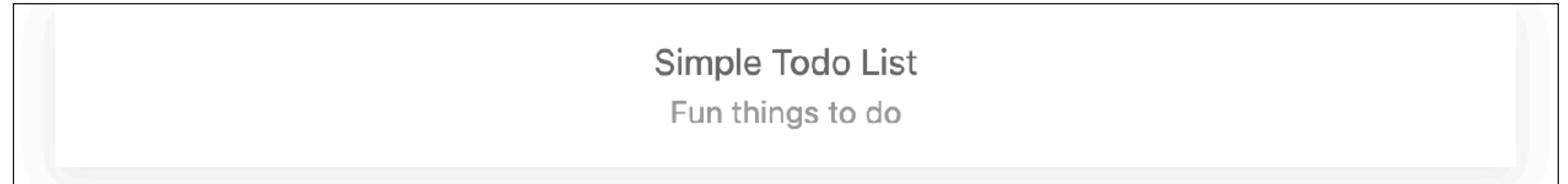
src/App.svelte

```
import Title from "../Title.svelte"
```

```
<div class="container">
  <Title />
  ...
</div>
```

- Simplest possible component
- No parameters or shared features of any kind
- Simply import and use the component as new type of element

Title.svelte (2)



```
<script>
  export let title;
  export let subTitle;
</script>
<div class="box has-text-centered">
  <div class="title"> {title} </div>
  <div class="subtitle"> {subTitle} </div>
</div>
```

- Export the properties we expect to receive parameters for

- Use these properties in the UX, enclosed in {}

- Pass properties to the component

```
import Title from "../Title.svelte"
```

```
<Title title="Simple Todo List" subTitle="Fun Things to do"/>
```


TodoList.svelte (1)

```
<script>
  export let todoItems = [];
</script>

<div class="section box">
  <div class="title is-6">Things yet do</div>
  <table class="table is-fullwidth">
    <thead>
      <th>Task</th>
      <th>Date</th>
      <th></th>
    </thead>
    <tbody>
      {#each todoItems as todo}
        <tr>
          <td> {todo.text} </td>
          <td> {todo.date} </td>
        </tr>
      {/each}
    </tbody>
  </table>
</div>
```

Items To Do :

TASK	DATE
Learn Svelte	23/3/2021, 13:34:55
Learn React	23/3/2021, 13:35:03

- Export the property we expect to receive
- Use this property in the UX, enclosed in {}
- Pass values for property when using the component
- However, the title and hard coded

src/App.svelte

```
import TodoList from "../TodoList.svelte";
```

And then use it:

```
<TodoList items="{todoItems}"/>
```

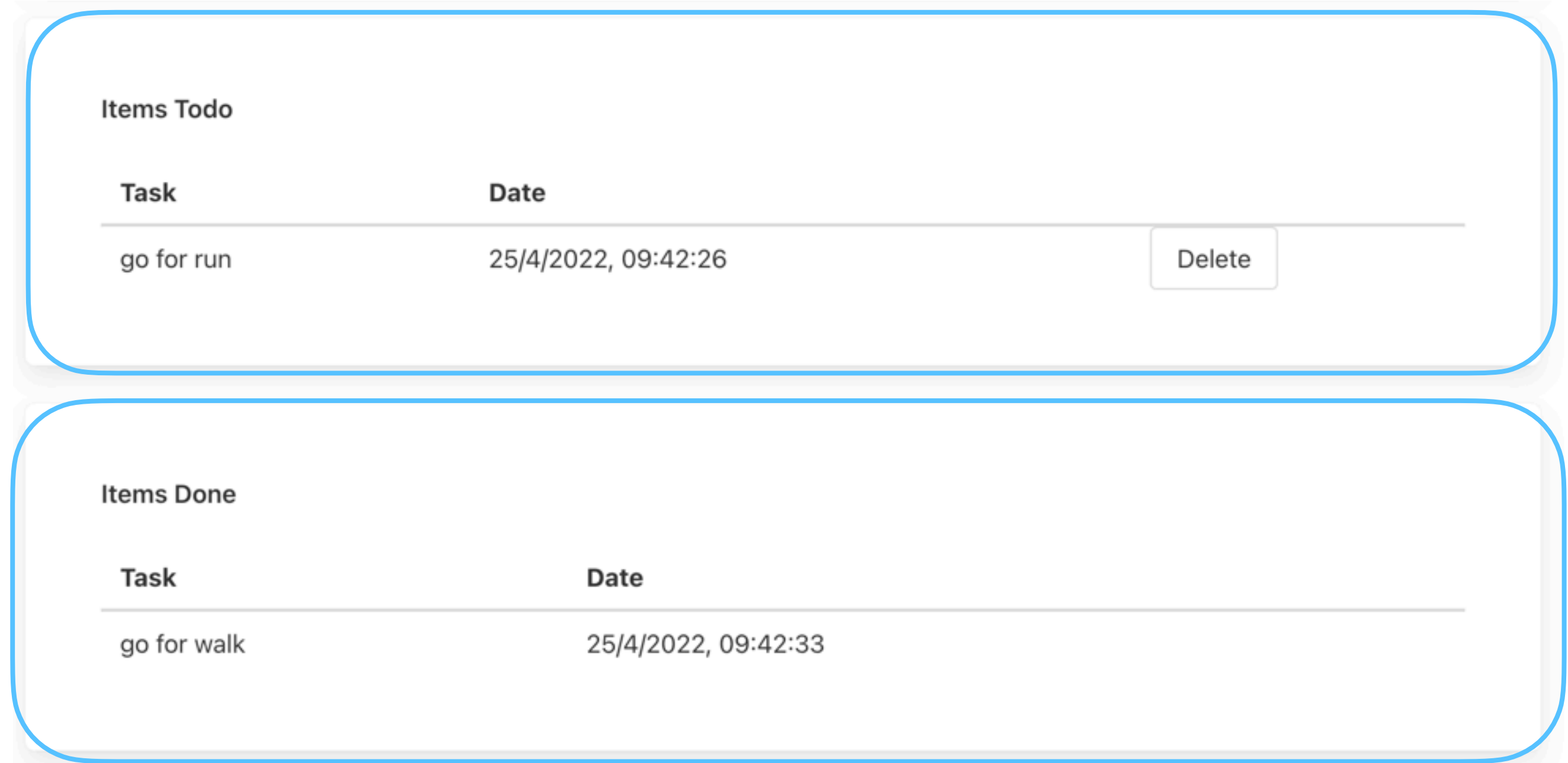
TodoList.svelte (2)

```
<script>
  export let caption;
  export let items = [];
</script>

<div class="section box">
  <div class="title is-6">{caption}</div>
  <table class="table is-fullwidth">
    <thead>
      <th>Task</th>
      <th>Date</th>
      <th></th>
    </thead>
    <tbody>
      {#each items as todo}
        <tr>
          <td> {todo.text} </td>
          <td> {todo.date}</td>
          </tr>
      {/each}
    </tbody>
  </table>
</div>
```

src/App.svelte

```
<TodoList caption="Items Todo" items="{todoItems}"/>
<TodoList caption="Items Done" items="{doneItems}"/>
```



Task	Date	
go for run	25/4/2022, 09:42:26	Delete


Task	Date
go for walk	25/4/2022, 09:42:33

- Also expect & use *caption*
- Use TodoList twice - passing different captions + list

TodoList.svelte (3)

```
<script>
  export let caption;
  export let items = [];
  export let deleteHandler = null;
</script>

<div class="section box">
  <div class="title is-6">{caption}</div>
  <table class="table is-fullwidth">
    <thead>
      <th>Task</th>
      <th>Date</th>
      <th></th>
    </thead>
    <tbody>
      {#each items as todo}
        <tr>
          <td> {todo.text} </td>
          <td> {todo.date}</td>
          {#if deleteHandler}
            <button on:click={deleteHandler(todo.id)} class="button">Delete</button>
          {/if}
        </tr>
      {/each}
    </tbody>
  </table>
</div>
```



```
function deleteTodo(id) {
  const found = todoItems.findIndex((todo) => todo.id == id);
  const done = todoItems[found];
  todoItems.splice(found, 1);
  todoItems = [...todoItems];
  doneItems.push(done);
  doneItems = [...doneItems];
}
```

src/App.svelte

```
<TodoList caption="Items Todo" items="{todoItems}" deleteHandler="{deleteTodo}"/>
```

Items To Do :

TASK	DATE
Learn Svelte	23/3/2021, 13:34:55
Learn React	23/3/2021, 13:35:03

DELETE

DELETE

- Support delete on Todo list only (not Done list)
- If statement in list
- Pass *deleteTodo* to component
- Notice that deleteHandler expects the id of the Todo to delete



AddTodoForm.svelte

```
<script>
  export let addTodo;
  let todoText;
</script>

<div class="field is-horizontal">
  <div class="field-label is-normal">
    <label for="todo" class="label">What should I do?</label>
  </div>
  <div class="field-body">
    <div class="field">
      <p class="control">
        <input bind:value={todoText} id="todo" class="input" type="text" placeholder="Type something...">
      </p>
    </div>
    <button on:click={addTodo(todoText)} class="button">Add Todo</button>
  </div>
</div>
```

src/App.svelte

```
import AddTodoForm from "../AddTodoForm.svelte";
```

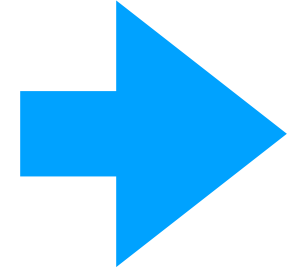
```
function addTodo(todoText) {
  const todo = {
    text: todoText,
    date: new Date().toLocaleString("en-IE"),
    id: uuidv4()
  };
  todoItems.push(todo);
  todoItems = [...todoItems];
  todoText = "";
}
```

```
<AddTodoForm addTodo="{addTodo}" />
```

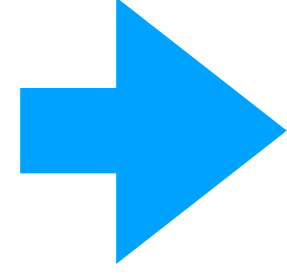
- Keep local copy of todo text
- Pass this to the addTodo function

What should I do? go for walk Add Todo

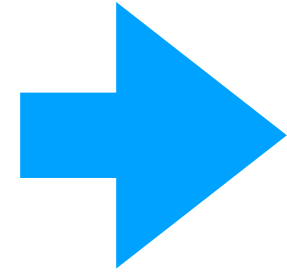
Import components



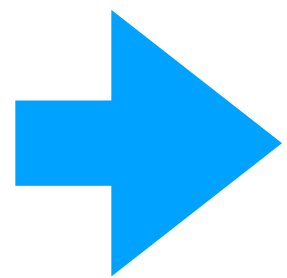
Local 'state'



Methods to update state.



UX



```
<script>
  import Title from "../Title.svelte"
  import AddTodoForm from "../AddTodoForm.svelte";
  import TodoList from "../TodoList.svelte";
  import { v4 as uuidv4 } from 'uuid';

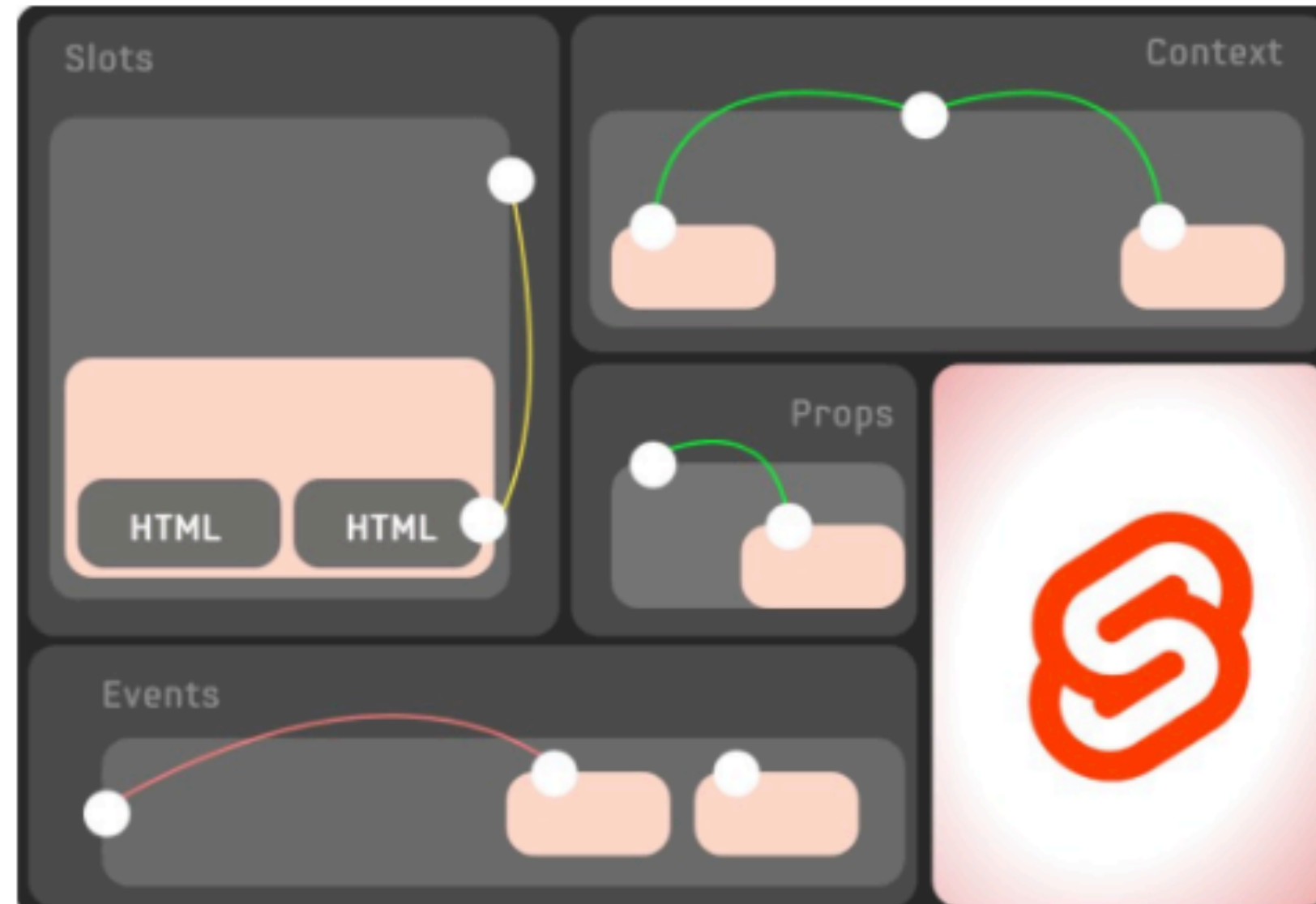
  let todoText = "";
  let todoItems = [];
  let doneItems = [];

  function addTodo(todoText) {
    const todo = {
      text: todoText,
      date: new Date().toLocaleString("en-IE"),
      id: uuidv4()
    };
    todoItems.push(todo);
    todoItems = [...todoItems];
    todoText = "";
  }

  function deleteTodo(id) {
    const found = todoItems.findIndex((todo) => todo.id == id);
    const done = todoItems[found];
    todoItems.splice(found, 1);
    todoItems = [...todoItems];
    doneItems.push(done);
    doneItems = [...doneItems];
  }
</script>

<div class="container">
  <Title title="Simple Todo List" subTitle="Fun Things to do"/>
  <div class="section box">
    <AddTodoForm addTodo="{addTodo}" />
  </div>
  <TodoList caption="Items Todo" items="{todoItems}" deleteHandler="{deleteTodo}"/>
  <TodoList caption="Items Done" items="{doneItems}"/>
</div>
```

Svelte Components



Creating and assembling
Svelte Components