Open-Source Software Practice

Lab 09. Web App

TA: **Jiwon Choi** (최지원, jasonchoi3@g.skku.edu)

Interactive Data Computing Lab (IDCLab)

College of Computing and Informatics,

Sungkyunkwan University

Goals

- Learn Front-end Toolkits
- Learn Browser-Event
- Develop and Publish SKKU-Todo App

Bootstrap

- Open-source Front-end Framework
- Maintain by Twitter -> Community
- Utility-First Framework
  - Set designs with class names.
- Pros
  - Easy to develop
  - Large community
- Cons
  - Every web app using Bootstrap looks same

New in v5.2 CSS variables, responsive offcanvas, new utilities, and more!

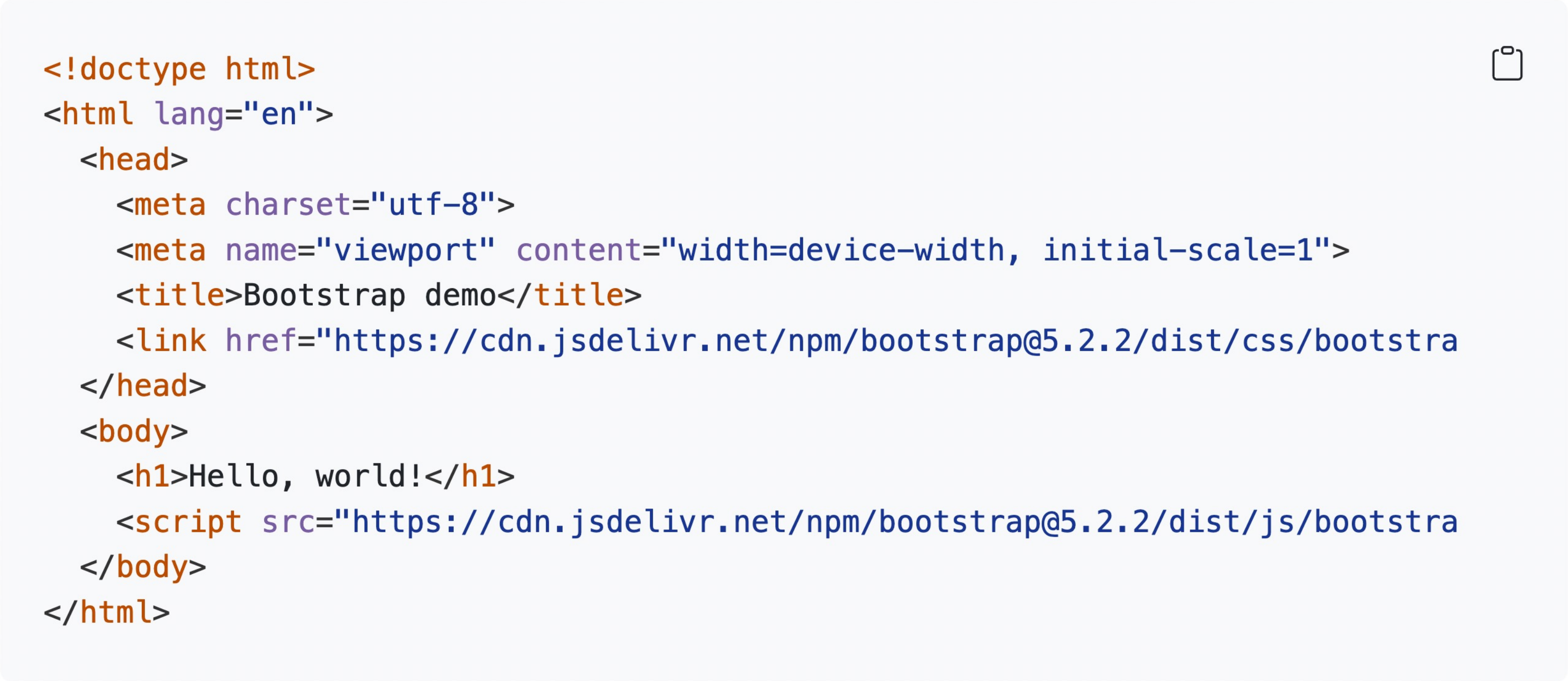
Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

```
$ npm i bootstrap@5.2.2
```

Start Bootstrap

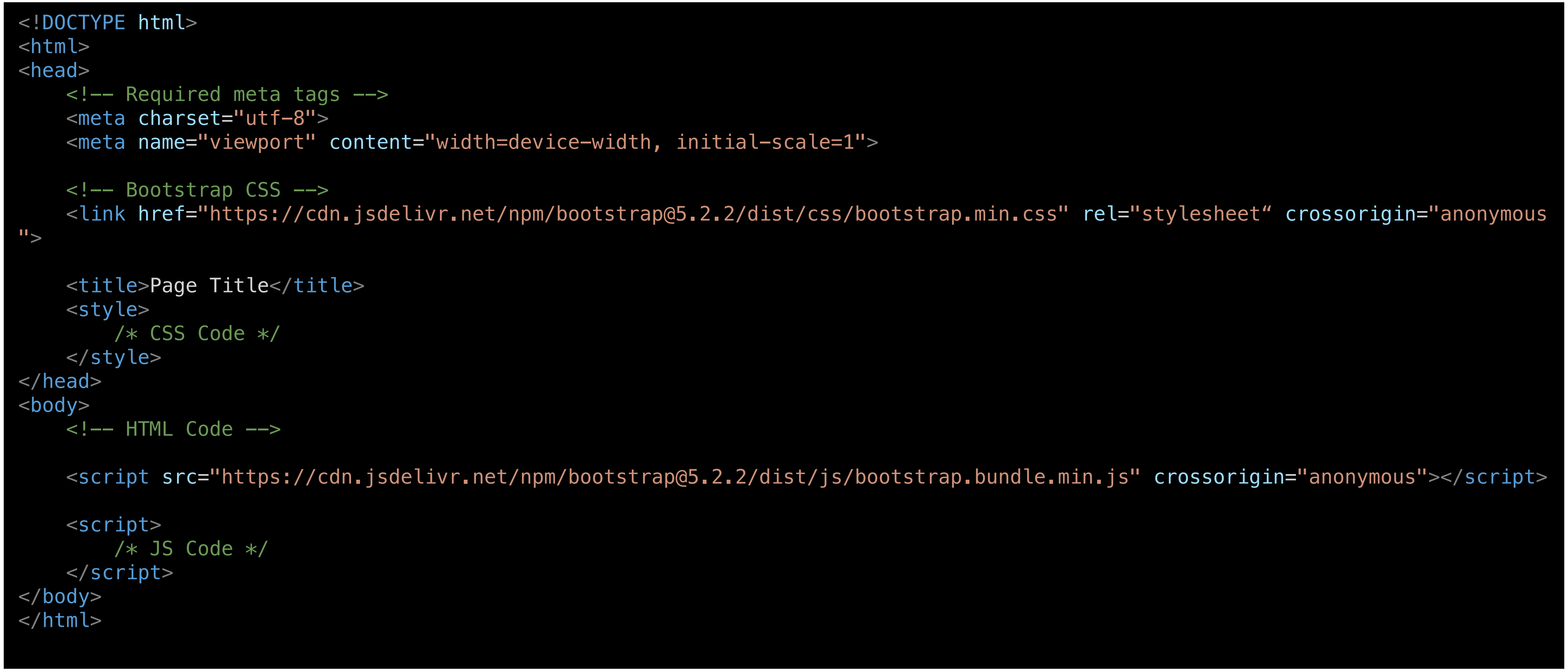
- Bootstrap Docs
  - https://getbootstrap.com/docs/5.2/getting-started/introduction/



```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstra
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstra
  </body>
</html>
```

Start Bootstrap

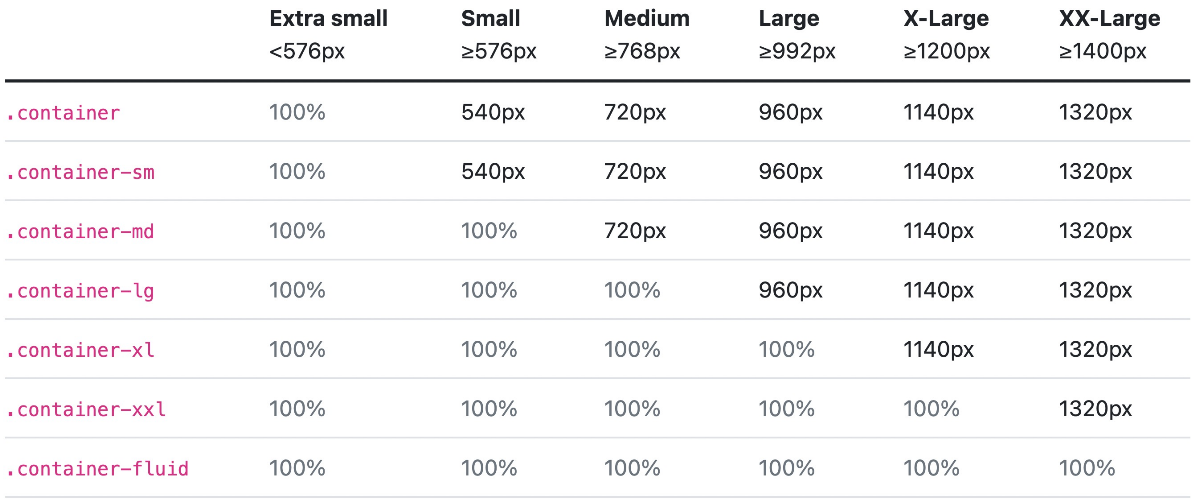
- https://getbootstrap.com/docs/5.2/examples/starter-template/



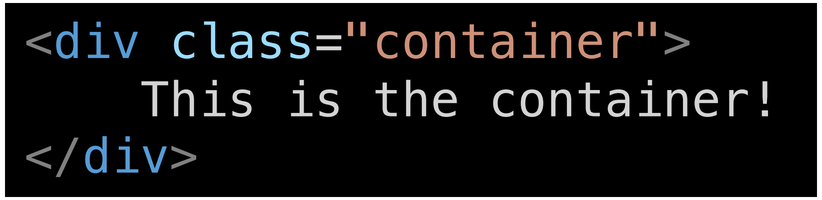
```
<!DOCTYPE html>
<html>
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet“ crossorigin="anonymous
">

    <title>Page Title</title>
    <style>
        /* CSS Code */
    </style>
</head>
<body>
    <!-- HTML Code -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>

    <script>
        /* JS Code */
    </script>
</body>
</html>
```

Adding Container

- Container
  - Containers are the most basic layout element in Bootstrap and are required when using our default grid system
  - Create ⟨div⟩ with class name “container”



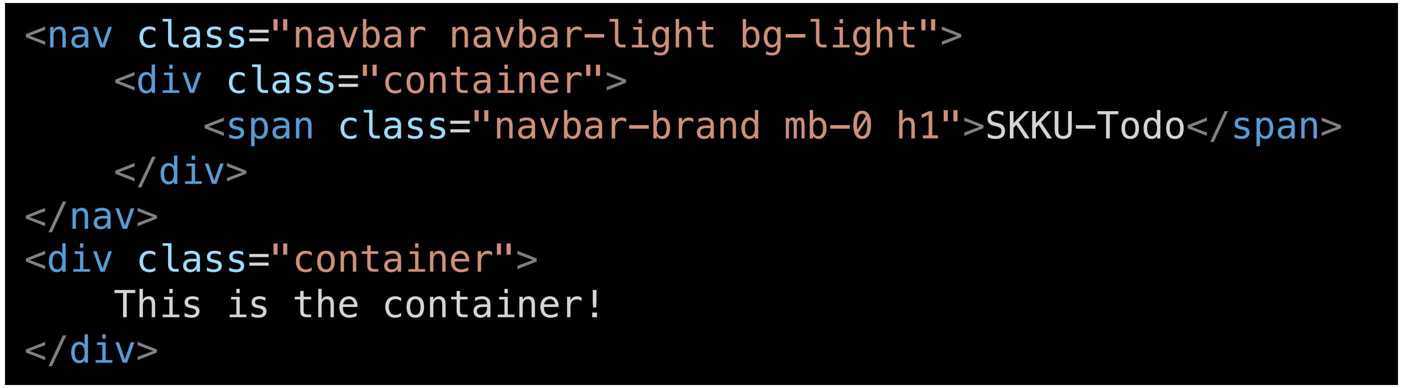
	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
`.container`	100%	540px	720px	960px	1140px	1320px
`.container-sm`	100%	540px	720px	960px	1140px	1320px
`.container-md`	100%	100%	720px	960px	1140px	1320px
`.container-lg`	100%	100%	100%	960px	1140px	1320px
`.container-xl`	100%	100%	100%	100%	1140px	1320px
`.container-xxl`	100%	100%	100%	100%	100%	1320px
`.container-fluid`	100%	100%	100%	100%	100%	100%



```html
<div class="container">
    This is the container!
</div>
```

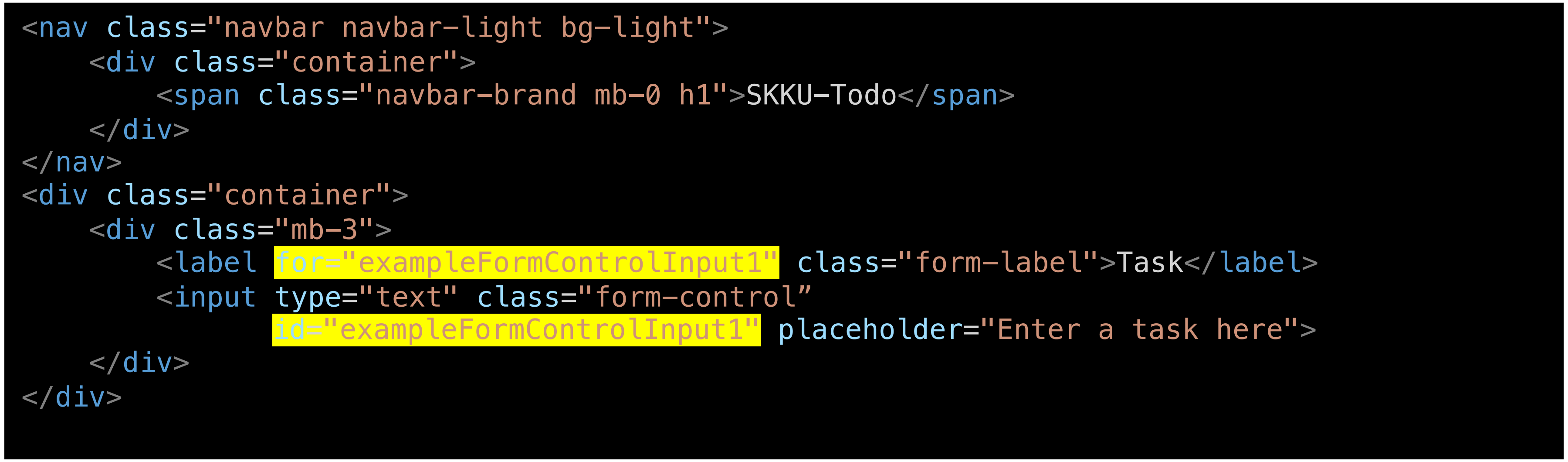
Adding Navbar

- Navbar
  - Section of a page whose purpose is to provide navigation links, either within the current document or to other documents.
- ⟨nav⟩ tag?
  - ⟨div⟩ tag with semantic meaning (for navigation)

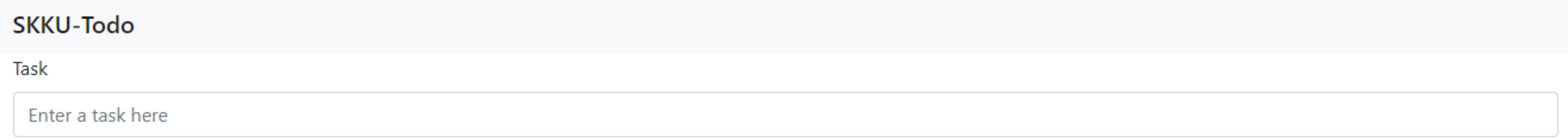


```
<nav class="navbar navbar-light bg-light">
    <div class="container">
        <span class="navbar-brand mb-0 h1">SKKU-Todo</span>
    </div>
</nav>
<div class="container">
    This is the container!
</div>
```

Adding Form

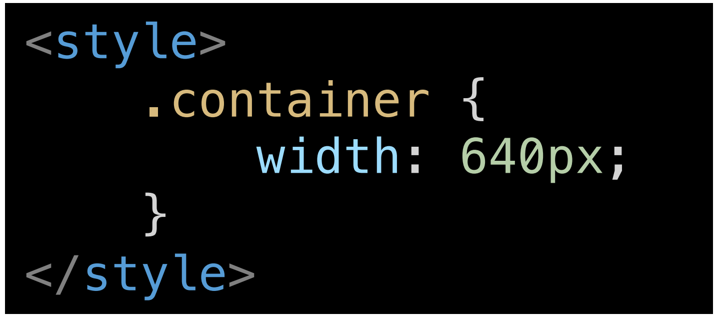


```
<nav class="navbar navbar-light bg-light">
    <div class="container">
        <span class="navbar-brand mb-0 h1">SKKU-Todo</span>
    </div>
</nav>
<div class="container">
    <div class="mb-3">
        <label for="exampleFormControlInput1" class="form-label">Task</label>
        <input type="text" class="form-control”
               id="exampleFormControlInput1" placeholder="Enter a task here">
    </div>
</div>
```



Adjusting CSS Property

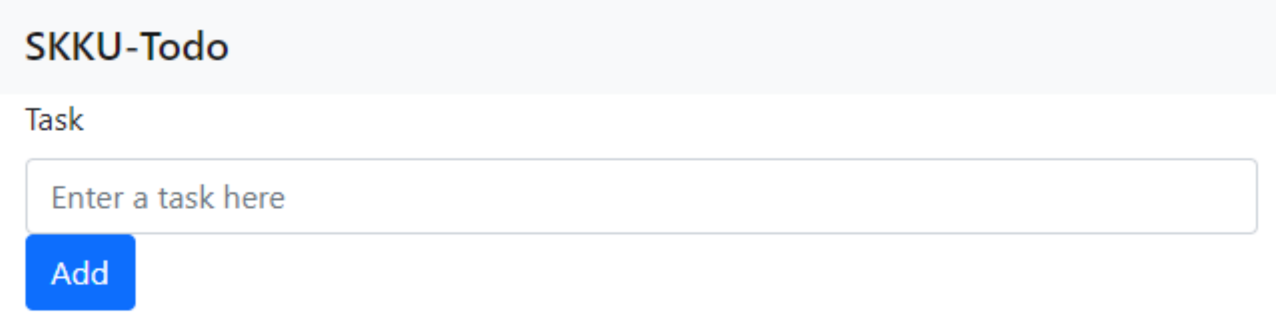
- You can adjust bootstrap component’s property using CSS



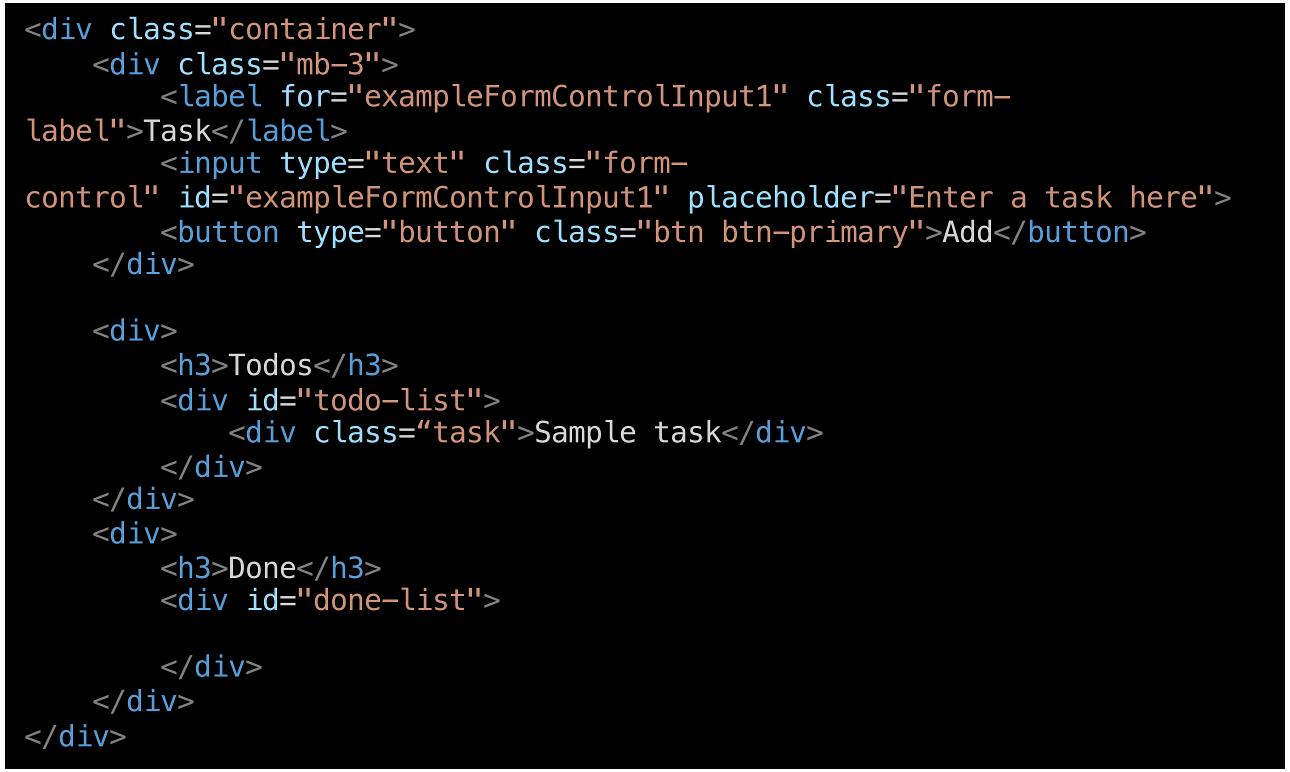
```
<style>
    .container {
        width: 640px;
    }
</style>
```

Adding Button

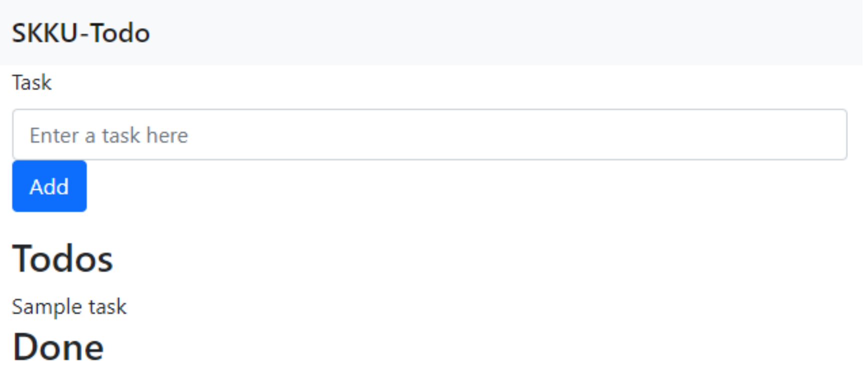
```
<div class="container">
    <div class="mb-3">
        <label for="exampleFormControlInput1" class="form-label">Task</label>
        <input type="text" class="form-control" id="exampleFormControlInput1" placeholder="Enter a task here">
        <button type="button" class="btn btn-primary">Add</button>
    </div>
</div>
```



Adding Lists



```
<div class="container">
    <div class="mb-3">
        <label for="exampleFormControlInput1" class="form-
label">Task</label>
        <input type="text" class="form-
control" id="exampleFormControlInput1" placeholder="Enter a task here">
        <button type="button" class="btn btn-primary">Add</button>
    </div>

    <div>
        <h3>Todos</h3>
        <div id="todo-list">
            <div class="task">Sample task</div>
        </div>
    </div>
    <div>
        <h3>Done</h3>
        <div id="done-list">

        </div>
    </div>
</div>
```



SKKU-Todo

Task

Enter a task here

Add

Todos

Sample task

Done

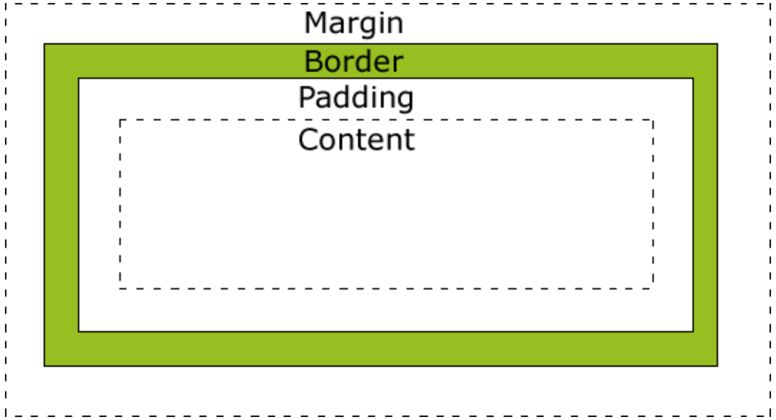
Styling the Interface

- Flexbox
  - Placing children object side-by-side
  - Class name “d-flex” makes 〈div〉 to flexbox.
  - “align-items-center”
  - “flex-grow-1”

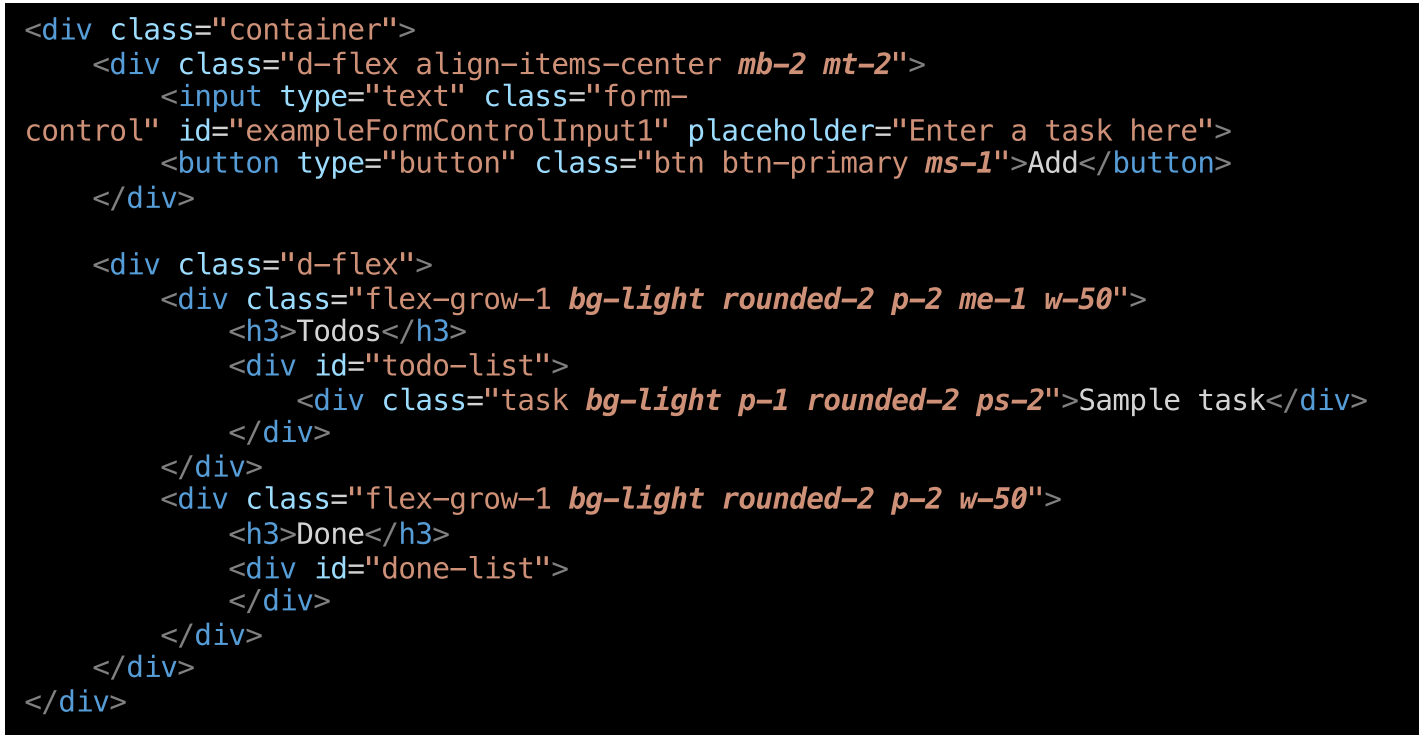
```
<div class="container">
    <div class="d-flex align-items-center mb-3">
        <label for="exampleFormControlInput1" class="form-label">Task</label>
        <input type="text" class="form-
control" id="exampleFormControlInput1" placeholder="Enter a task here">
        <button type="button" class="btn btn-primary">Add</button>
    </div>

    <div class="d-flex">
        <div class="flex-grow-1">
            <h3>Todos</h3>
            <div id="todo-list">
                <div class="task">Sample task</div>
            </div>
        </div>
        <div class="flex-grow-1">
            <h3>Done</h3>
            <div id="done-list">
            </div>
        </div>
    </div>
</div>
```

Styling the Interface

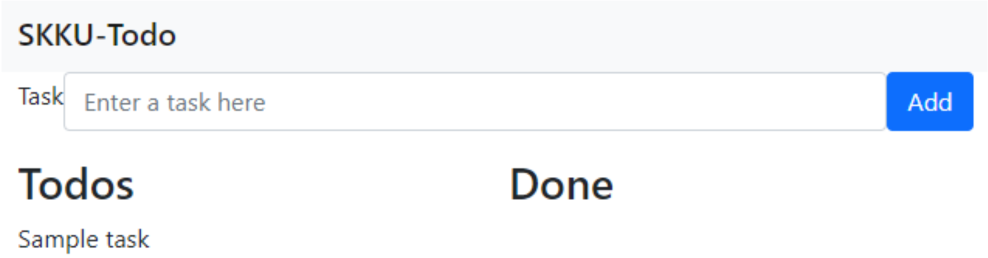
- **Utility Classes**
  - useful to control the appearance of tags without adding CSS rules
  - {property}{sides}-{size}
  - property: m (margin), p (padding)
  - sides: t (top), b (bottom), s (start or left), e (end or right), x (left and right), y (top and bottom)
  - size: integer from 0 to 5 where 1 = 0.25rem

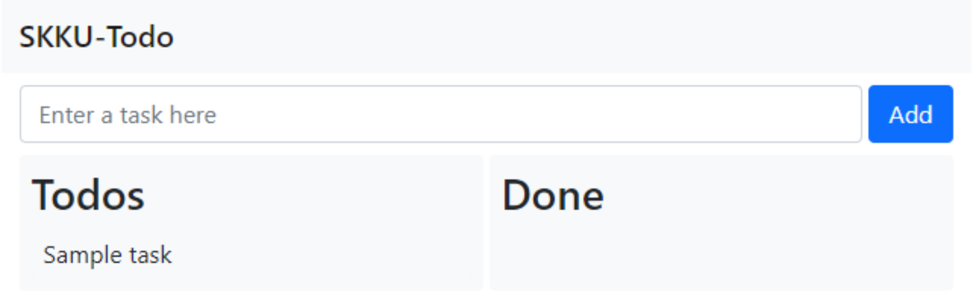


Styling the Interface



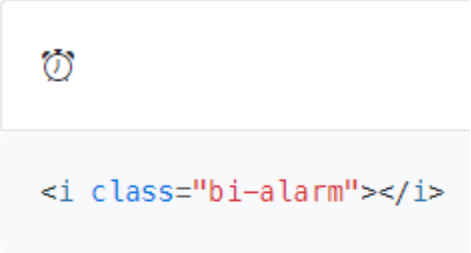
```
<div class="container">
    <div class="d-flex align-items-center mb-2 mt-2">
        <input type="text" class="form-
control" id="exampleFormControlInput1" placeholder="Enter a task here">
        <button type="button" class="btn btn-primary ms-1">Add</button>
    </div>

    <div class="d-flex">
        <div class="flex-grow-1 bg-light rounded-2 p-2 me-1 w-50">
            <h3>Todos</h3>
            <div id="todo-list">
                <div class="task bg-light p-1 rounded-2 ps-2">Sample task</div>
            </div>
        </div>
        <div class="flex-grow-1 bg-light rounded-2 p-2 w-50">
            <h3>Done</h3>
            <div id="done-list">
            </div>
        </div>
    </div>
</div>
```



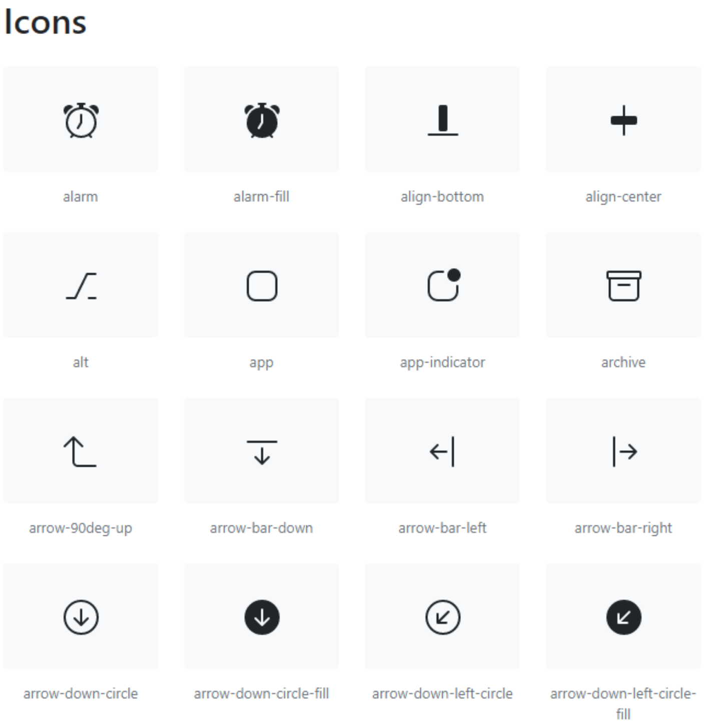


Icons

- Bootstrap-Icons
  - Bootstrap provides several pre-defined icons.
  - https://icons.getbootstrap.com



```
<i class="bi-alarm"></i>
```

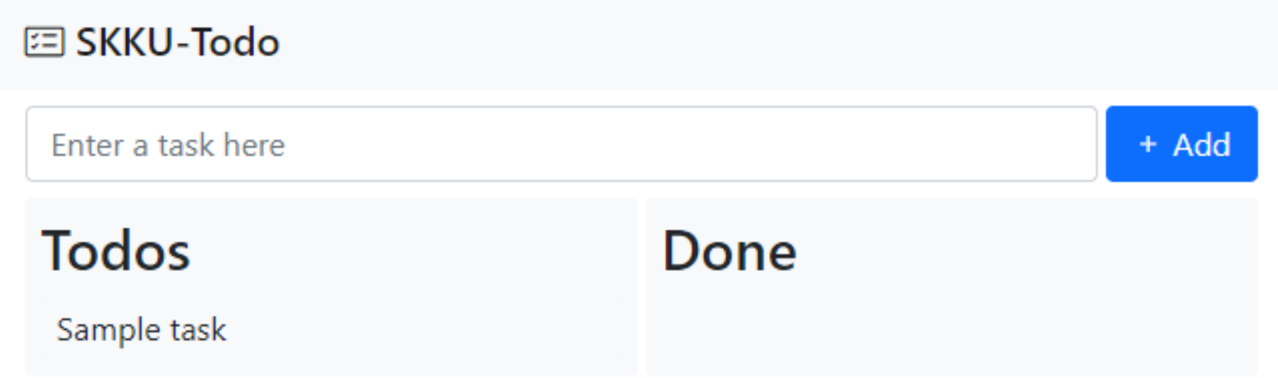


```
<!-- Bootstrap CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap... >
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.1/font/bootstrap-icons.css">
```

Icons

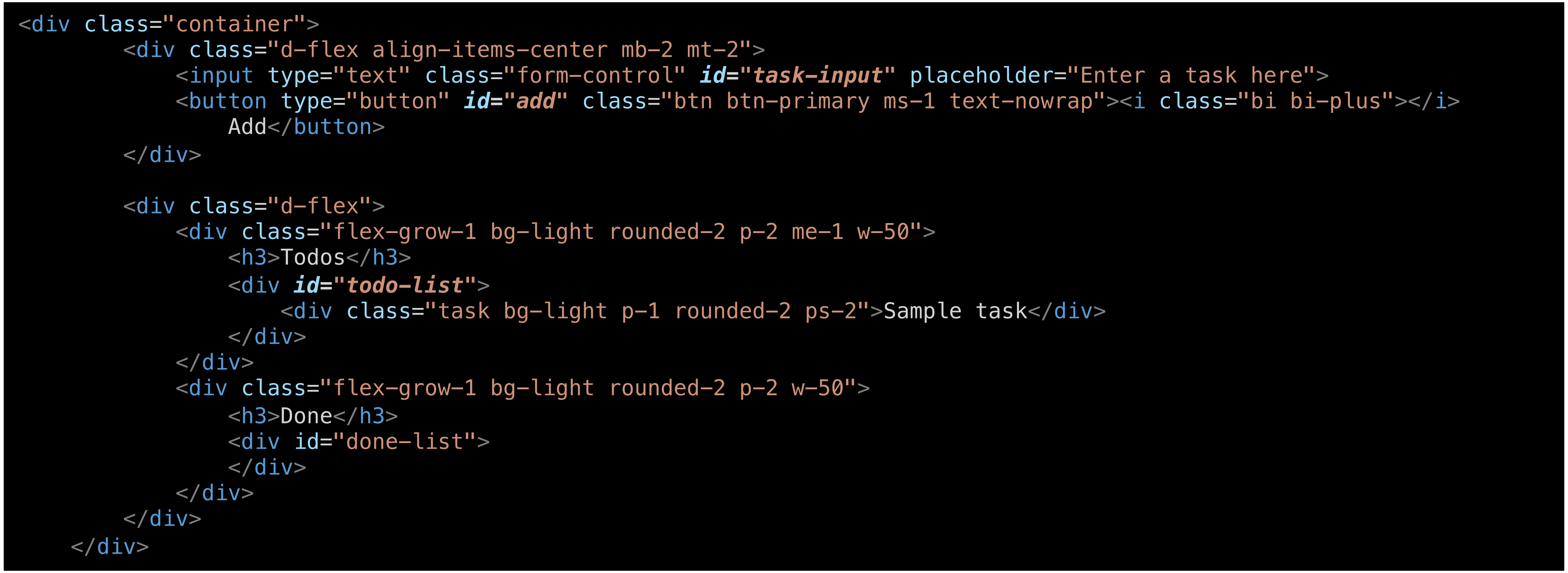
- Let's add some icons.

```
<nav class="navbar navbar-light bg-light">
    <div class="container">
        <span class="navbar-brand mb-0 h1"><i class="bi bi-card-checklist"></i> SKKU-Todo</span>
    </div>
</nav>
<div class="container">
    <div class="d-flex align-items-center mb-2 mt-2">
        <input type="text" class="form-control" id="exampleFormControlInput1" placeholder="Enter a task here">
        <button type="button" class="btn btn-primary ms-1 text-nowrap"><i class="bi bi-plus"></i> Add</button>
    </div>
```



Set id

- Set id
  - id must be unique.

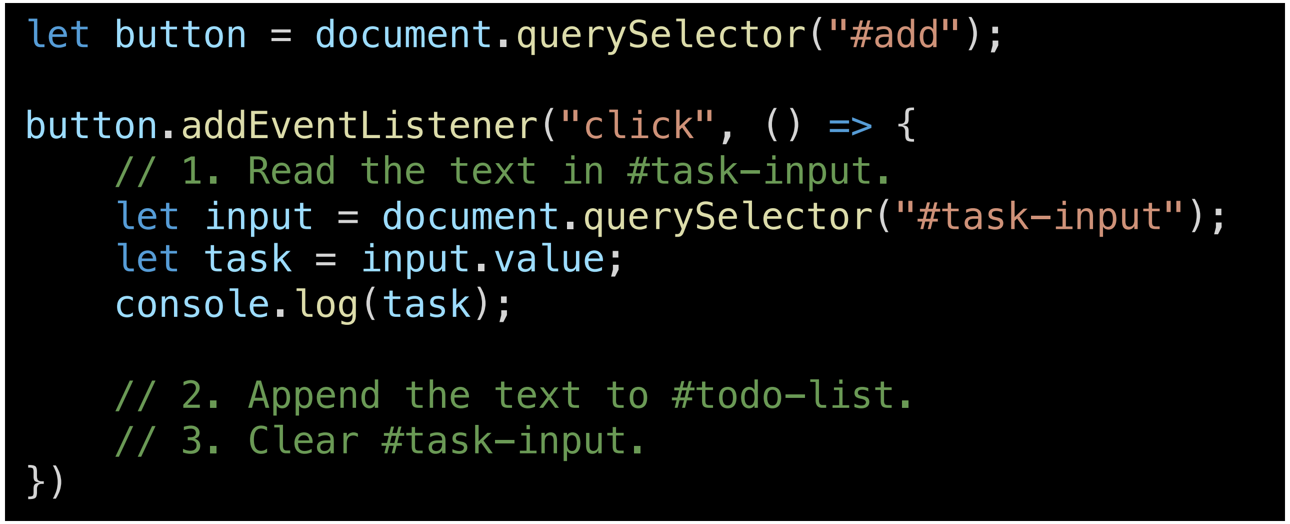


```
<div class="container">
        <div class="d-flex align-items-center mb-2 mt-2">
            <input type="text" class="form-control" id="task-input" placeholder="Enter a task here">
            <button type="button" id="add" class="btn btn-primary ms-1 text-nowrap"><i class="bi bi-plus"></i>
                Add</button>
        </div>

        <div class="d-flex">
            <div class="flex-grow-1 bg-light rounded-2 p-2 me-1 w-50">
                <h3>Todos</h3>
                <div id="todo-list">
                    <div class="task bg-light p-1 rounded-2 ps-2">Sample task</div>
                </div>
            </div>
            <div class="flex-grow-1 bg-light rounded-2 p-2 w-50">
                <h3>Done</h3>
                <div id="done-list">
                </div>
            </div>
        </div>
    </div>
```

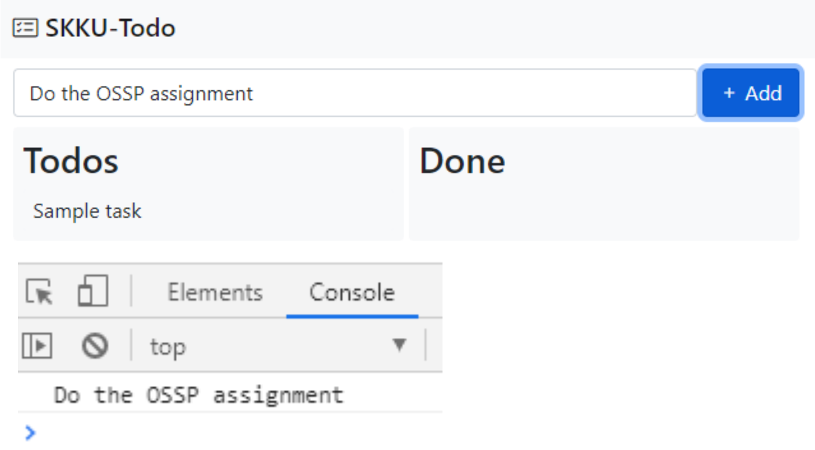
Event?

- Event
  - JavaScript object that indicates something happens in browser
    - Click
    - Change
    - Page loaded ...
- Event Handler
  - Event handler is a function that is called whenever the corresponding event occurs

Getting the Input



```
let button = document.querySelector("#add");

button.addEventListener("click", () => {
    // 1. Read the text in #task-input.
    let input = document.querySelector("#task-input");
    let task = input.value;
    console.log(task);

    // 2. Append the text to #todo-list.
    // 3. Clear #task-input.
})
```



SKKU-Todo

Do the OSSP assignment

Todos

Sample task

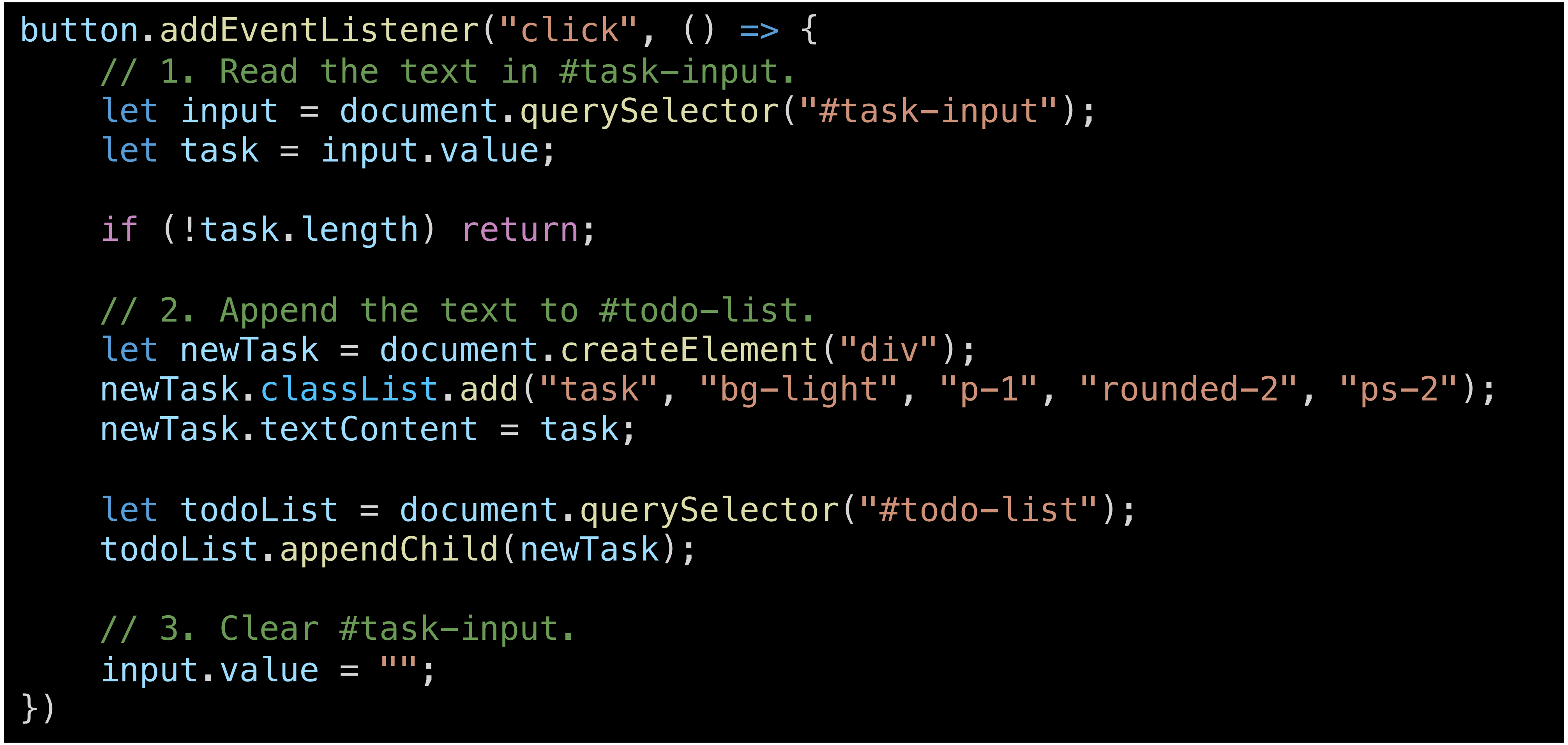
Done

```
Do the OSSP assignment
```

Creating an Element

- **document.createElement(tagName)**
  - Create HTMLObjectElement with tag name
- **HTMLObjectElement.appendChild(HTMLObjectElement)**
  - Append child to selected tag

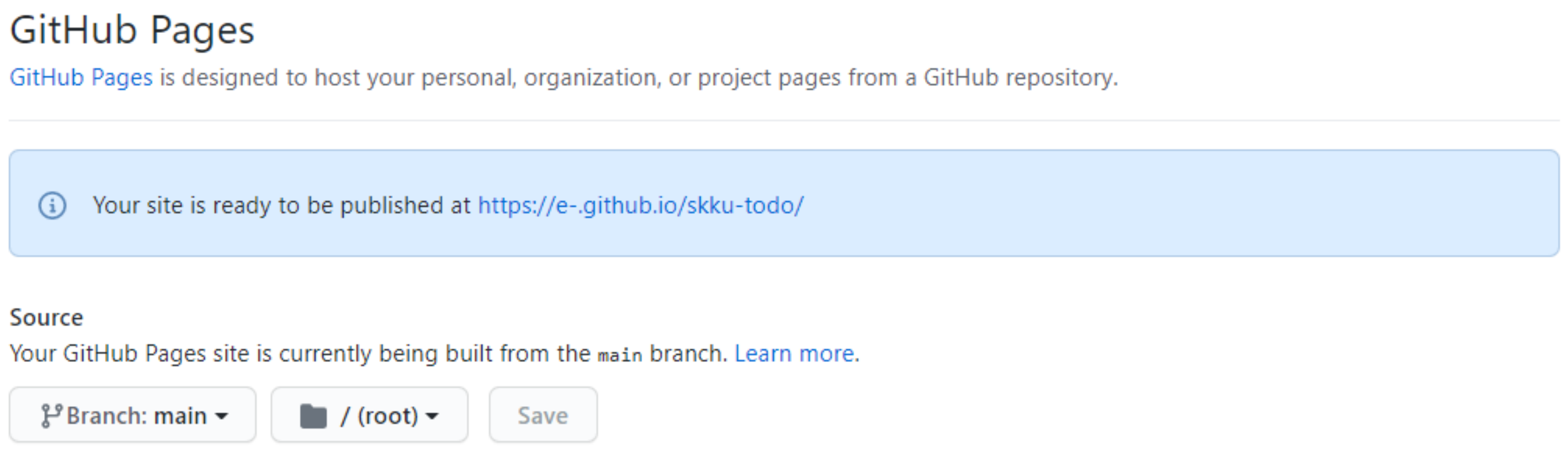
Creating an Element



```
button.addEventListener("click", () => {
    // 1. Read the text in #task-input.
    let input = document.querySelector("#task-input");
    let task = input.value;

    if (!task.length) return;

    // 2. Append the text to #todo-list.
    let newTask = document.createElement("div");
    newTask.classList.add("task", "bg-light", "p-1", "rounded-2", "ps-2");
    newTask.textContent = task;

    let todoList = document.querySelector("#todo-list");
    todoList.appendChild(newTask);

    // 3. Clear #task-input.
    input.value = "";
})
```

Let's Publish

- Create a Github repository
- Rename html file to "index.html" (mandatory)
- Commit the HTML file to the repository
- Go to repository settings and change source and save



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

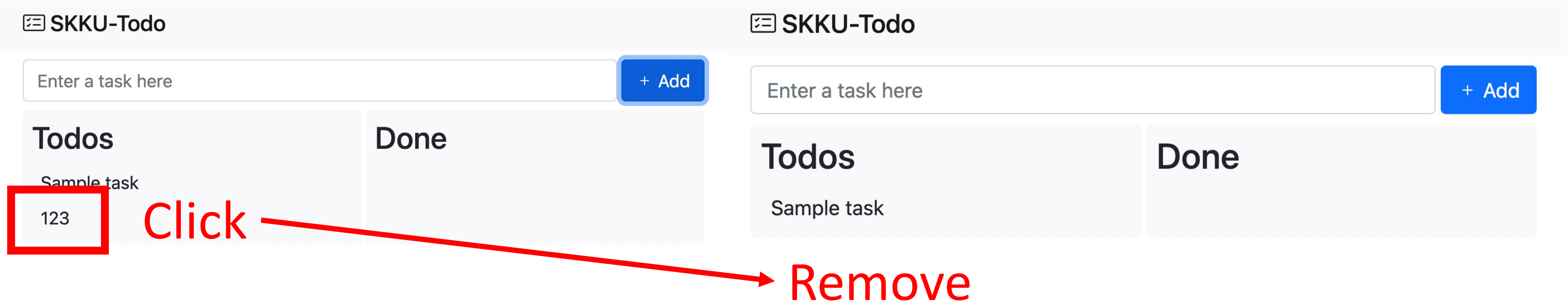
ⓘ Your site is ready to be published at https://e-.github.io/skku-todo/

Source

Your GitHub Pages site is currently being built from the `main` branch. Learn more.

Summary Quiz

- Let’s remove todo when todo is clicked
  - Hint: Use .remove() method and EventListener



Click

Remove