

```
1 module Solution exposing (main)
2
3 import Browser
4 import Html exposing (..)
5 import Html.Attributes exposing (..)
6 import Html.Events exposing (..)
7
8
9
10 --- MODEL ---
11
12
13 type alias Model =
14     { inputText : String
15     , todos : List Todo
16     , nextId : Int
17     }
18
19
20 type alias Todo =
21     { text : String
22     , id : Int
23     , completed : Bool
24     }
25
26
27 initialModel : Model
28 initialModel =
29     { inputText = ""
30     , todos = []
31     , nextId = 1
32     }
33
34
35
36 --- UPDATE ---
37
38
39 type Msg
40     = TextUpdated String
41     | TodoAdded
42     | TodoDeleted Int
43     | CompleteToggled Int
44
45
46 update : Msg -> Model -> Model
47 update msg model =
48     case msg of
49         TextUpdated string ->
```

```

50         { model | inputText = string }
51
52     TodoAdded ->
53         { model
54           | inputText = ""
55           , todos = createTodo model.inputText model.
nextId :: model.todos
56           , nextId = model.nextId + 1
57         }
58
59     TodoDeleted deletedTodoId ->
60         { model
61           | todos =
62             List.filter
63               (\todo -> todo.id /= deletedTodoId)
64               model.todos
65         }
66
67     CompleteToggled completedTodoId ->
68         { model
69           | todos =
70             List.map
71               (\todo ->
72                 if todo.id == completedTodoId
73 then
74                     { todo | completed = not
75                       todo.completed }
76                 else
77                     todo
78               )
79               model.todos
80         }
81
82 createTodo : String -> Int -> Todo
83 createTodo string id =
84     { text = string
85       , id = id
86       , completed = False
87     }
88
89
90
91 --- VIEW ---
92
93
94 view : Model -> Html Msg
95 view model =

```

```

96     section [ class "todoapp" ]
97         [ header [ class "header" ]
98             [ h1 [] [ text "todos" ]
99                 , div [ class "new-todo-row" ]
100                     [ input
101                         [ class "new-todo"
102                             , placeholder "What needs to be done?"
103                             , autofocus True
104                             , autocomplete False
105                             , name "newTodo"
106                             , onInput TextUpdated
107                             , value model.inputText
108                         ]
109                     []
110                 , button
111                     [ class "add-todo"
112                         , onClick TodoAdded
113                     ]
114                     [ text "Add" ]
115                 ]
116         ]
117     , section [ class "main" ]
118         [ ul [ class "todo-list" ]
119             (List.map viewTodo model.todos)
120         ]
121     ]
122
123
124 viewTodo : Todo -> Html Msg
125 viewTodo todo =
126     li [ classList [ ( "completed", todo.completed ) ] ]
127         [ div [ class "view" ]
128             [ input [ class "toggle", type_ "checkbox",
129                 checked todo.completed, onClick (CompleteToggled todo.id)
130             ] []
131             , label [] [ text todo.text ]
132             , button [ class "destroy", onClick (
133                 TodoDeleted todo.id ) ] []
134         ]
135
136 --- INIT ---
137
138
139 main =
140     Browser.sandbox
141         { init = initialModel

```

```
142         , view = view
143         , update = update
144     }
145
146
147
148 -- Ekstraoppgave: Legg til følgende etter section [ class
   "main" ], og implementer funksjonene
149 {--
150 footer [ class "footer" ]
151     [ span [ class "todo-count" ]
152       [ strong [] [ text "x" ]
153         , text " items left"
154       ]
155     , ul [ class "filters" ]
156       [ li []
157         [ a [ class "selected" ] [ text "All" ]
158         ]
159       , li []
160         [ a [] [ text "Active" ]
161         ]
162       , li []
163         [ a [] [ text "Completed" ]
164         ]
165       ]
166     , button [ class "clear-completed" ] [ text "Clear
   completed (x)" ]
167   ]
168 --}
169 -- (a [] [ text "Active"] er egentlig en lenke, men man kan
   behandle dem som buttons med onClick)
170
```