# HTML

The following code defines datatypes for representing structured (X)HTML markup.

```
data Attr = Attr String String
  deriving (Eq,Show)

data HtmlElement
  = HtmlString String
  | HtmlTag String [Attr] HtmlElements
  deriving (Eq, Show)

type HtmlElements = [HtmlElement]
```

A piece of HTML code is either plain text `HtmlString` or is a tagged node `HtmlTag` with attributes. In case of a node, other elements can be nested under it. The following HTML code

```
<a href="https://www.kuleuven.be/kuleuven/">
KU Leuven
</a>
```

is represented by the following value:

```
example :: HtmlElement
example = HtmlTag "a"
                  [Attr "href" "https://www.kuleuven.be/kuleuven/"]
                  [HtmlString "KU Leuven"]
```

We can group all types that can be rendered as HTML in a type class:

```
class HTML a where
  toHtml :: a -> HtmlElement
```

- Write an `HTML` instance that creates an anchor for the following datatype

  ```
  data Link = Link
                String  -- Link target.
                String  -- Text to show.
  ```

- Encode the following unordered HTML list as an `HtmlElement`:

  ```
  <ul>
  <li>Apples</li>
  <li>Bananas</li>
  <li>Oranges</li>
  </ul>
  ```

- Write an `HTML` instance for Haskell lists using unordered HTML lists.
- Model datatypes for an address book by defining a type `AddressBook` (and as many other data types you need). You should store at least the following information about your contacts:
  - First and last name.
  - A list of email addresses.
  - For each email address you should store if it is a work or private email address.
- Define an example address book `exampleAddressBook :: AddressBook` with at least two entries.
- Define `HTML` instances for the types of your address book. This exercise can become **very long** if you only construct values of `HtmlElement` directly. Try to abstract over recurring code.

**NOTE:** For this exercise there is only little coverage from our testing framework (E-Systant), since this exercise involves a user-defined data type, not known to the system. Hence, be extra careful when defining these data types.