Easy TEI Publication with CETEIcean (/sɪˈtiːʃn/) 🐳

Raff Viglianti

Get this presentation: http://j.mp/EasyTEI (also in shared Drive folder)
Easy publication?

- Nothing’s ever easy, just less complex
- Minimal technical stack for running a TEI project
- No server-side component needed
- Isomorphic TEI - HTML: TEI in the Browser

Get this presentation: http://j.mp/EasyTEI (also in shared Drive folder)
CETEIcean: TEI as HTML

- **CETEIcean** 🐳 = HTML5 Custom Elements + TEI
  
  (/sɪˈtiːʃn/)  [https://github.com/TEIC/CETEIcean](https://github.com/TEIC/CETEIcean)

- Custom Elements look like `<prefix-elementname>`, so `<tei-text>`, for example.
  - Only supported in modern browsers, but no problem

- CETEIcean either loads a TEI XML document and converts it to Custom Elements or takes a pre-converted document and "upgrades" it.

Get this presentation: [http://j.mp/EasyTEI](http://j.mp/EasyTEI)
(also in shared Drive folder)
Motivations for CETEIcean

- Using web technologies for web publication
  - XSLT/XQuery are not sufficient, by themselves, to create a digital publication on the web: “front end” code (CSS, JavaScript) is always necessary
- Semantics
  - Shared text model between TEI and its publication
- Preservation
  - Server-side software can be omitted
- Getting started faster
Let’s try it out 🐳
Gearing up: what you’ll need

- **A TEI file** (feel free to use your own)
  - Need one? [Here’s one](#) (also in shared Drive folder)

- **A code editor** for HTML and JavaScript
  - *Recommended: Atom* (free)
  - Oxygen is ok, too
  - Many more!

- **A way to preview a simple website**
  - *Recommended: Atom* with plugin ‘atom-html-preview’
  - Experts: use a simple HTTP server with python, nodejs, etc.
Creating static sites for TEI documents

What is a static site?

A web page delivered to the user exactly as stored

What is it made of?

- HTML
- CSS
- JAVASCRIPT
- OTHER ASSETS
Creating static sites for TEI documents

What is a static site?
A web page delivered to the user exactly as stored

What is it made of?

- HTML
- CSS
- JAVASCRIPT
- IMAGES
- OTHER MEDIA
- TEI
Creating static sites for TEI documents

Create this folder structure for your static site

Copy your TEI file into data

MySite

css
data
js
index.html
Creating static sites for TEI documents

Download CETEI.clean from: (direct link) or http://j.mp/CETEI

Copy CETEI.js into js
Creating static sites for TEI documents

Copy CETEIcean.css into css

Get CETEIcean’s example CSS file ([direct link](https://goo.gl/kvwHPK)) or https://goo.gl/kvwHPK

MySite

- css
- data
- js
- index.html
<!doctype html>
<html lang="en">
<head>
  <title>My project</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="css/CETEIcean.css" />
  <script src="js/CETEI.js"></script>
</head>
<body>
  <div id="TEI"></div>
  <script>
    var CETEIcean = new CETEI()
    CETEIcean.getHTML5("data/YOUR_TEI.xml", function(data) {
      document.getElementById("TEI").appendChild(data)
    })
  </script>
</body>
</html>
<!doctype html>
<html lang="en">
<head>
  <title>My project</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="css/CETEIcean.css"/>
  <script src="js/CETEI.js"></script>
</head>
<body>
  <div id="TEI"></div>
  <script>
    var CETEIcean = new CETEI();
    CETEIcean.getHTML5("data/YOUR_TEI.xml", function(data) {
      document.getElementById("TEI").appendChild(data)
    })
  </script>
</body>
</html>
Index.html - explained

<!doctype html>
<html lang="en">
<head>
  <title>My project</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="css/CETEIcean.css"/>
  <script src="js/CETEI.js"></script>
</head>
<body>
  <div id="TEI"></div>
  <script>
    var CETEIcean = new CETEI()
    CETEIcean.getHTML5("data/YOUR_TEI.xml", function(data) {
      document.getElementById("TEI").appendChild(data)
    })
  </script>
</body>
</html>
Loading CETEIcon

```html
<script>
    var CETEIcon = new CETEI();
</script>
```
Loading CETEIcean

```html
<script>
  var CETEIcean = new CETEI()
  CETEIcean.getHTML5("data/YOUR_TEI.xml")
</script>

* As long as it’s on the same domain, or the site serving it allows cross-domain requests.
Loading CETEIcean

<script>
    var CETEIcean = new CETEI();
    CETEIcean.getHTML5("data/YOUR_TEI.xml", function(data) {
        
    })
</script>

Create a function (block of code) to do something with the TEI as Custom Elements
Loading CETEIcean

```javascript
var CETEIcean = new CETEI()
CETEIcean.getHTML5("data/YOUR_TEI.xml", function(data) {
  document.getElementById("TEI").appendChild(data)
})
</script>

Find `<div id="TEI">` and place the data there.
Adding some style
Styling with CSS

- Let’s add styles specific to your TEI document
- Create `style.css` and add it to `css`
- Link to it from the HTML:
  ```html
  <link rel="stylesheet" href="css/CETEIcean.css" />
  <link rel="stylesheet" href="css/style.css" />
  ```

Add this line under the link to CETEIcean’s example CSS
Writing new styles

- Basic CSS syntax

  selector  declaration
  div        { color: red;   text-align: center }

- CETEIcean adds a tei- prefix to TEI elements to distinguish them from HTML ones

  selector  declaration
  tei-div   { color: red;   text-align: center }
Styling with CSS

Let’s add a new style declaration for `tei-head` in `style.css`

```css
.tei-head {
  color: red;
}
```

Most editors will suggest CSS properties and values as you type.

Feel free to style any other TEI element that may be in your document!

Just prefix it with `tei-`

E.g.: `tei-del` or `tei-persName`
Styling with CSS

Let’s add a new style declaration for tei-supplied in style.css

tei-head {
  color: red;
}

A pseudoselector

tei-head:before {
  content: "["
}

tei-head:after {
  content: ""]"
}
Play Time!

Work in groups or individually

Prompts:

- Add more CSS! Here’s a cheat sheet
- Switch to another TEI file you know, if you like.

Get this presentation: http://j.mp/EasyTEI
Publishing your static site on GitHub
Publishing your code to GitHub

- If you haven’t already, create a GitHub account on github.com
- Create a new repository for your site

Owner
raffazizzi

Repository name
myTEIsite

Great repository names are short and memorable. Need inspiration? How about upgraded-happiness?
Publishing your code to GitHub

- Upload your files (if you can, without their container folder)
- Click on “uploading an existing file”

Quick setup — if you’ve done this kind of thing before

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

- To update your files later, simply re-upload them.
Publishing your site with GitHub Pages

- Go to Settings
- Turn on GitHub pages

GitHub Pages

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

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

- None
- Save

Select source

- master branch
  Use the master branch for GitHub Pages.

Learn more.
Publishing your site with GitHub Pages

- Wait a little bit
- Then visit your website!

https://USERNAME.github.io/PROJECT
Behaviors
CETEIcean behaviors

- What if we want more than what CSS can provide?
- CETEIcean elements can be configured with their own behaviors.
CETEIcean behaviors

- What if we want more than what CSS can provide?
- CETEIcean elements can be configured with their own **behaviors**.

JS object, registered via `CETEI.addBehaviors()` **Before** loading the TEI document.
CETEIcean behaviors

- Prepend and append
  - Just like CSS :before and :after, but actually changes the text and makes the added content selectable

```javascript
var CETEIcean = new CETEI();
CETEIcean.addBehaviors({
  handlers: {
    "head": [ "[", "]" ]
  }
})
```

Note that the selector “head” has no tei- prefix in this case.
CETEIcean behaviors

- More complex operations.* Link a persName to its @ref

```javascript
CETEIcean.addBehaviors({
    handlers: {
        "persName": function(el) {
            var a = document.createElement("a")
            a.setAttribute("href", el.getAttribute("ref"))
            a.innerHTML = el.innerHTML;
            return a
        }
    }
})
```

* This is when things turn a bit more like XSLT, as we’re modifying the data. However, the persName TEI element remains intact. The aim is to get the text model closer to the user.

Rainer Rilke

https://viaf.org/viaf/46897170/
CETEIcean behaviors

- Full guide on CETEIcean behaviors (by Hugh Cayless):

  https://github.com/TEIC/CETEIcean/wiki/Anatomy-of-a-behaviors-object
Play Time!

Prompts:

- Keep styling your TEI file
- Build other HTML pages for your site
- Add simple JavaScript interaction (see next slides / ask me how)

Work in groups or individually

Get this presentation: http://j.mp/EasyTEI
User interaction

Typically, user interaction is created in JavaScript even with XSLT-based workflows.

- Use JavaScript to dynamically change the CSS of TEI elements
- We’ll see how to:
  - hide some stuff with CSS
  - create a “click” event
  - toggle visibility
- Our case study: subst, add, and del

There are many ways of adding interaction to a web page. We’re just taking a quick dip!
User interaction

Hide del in style.css, but only when its container is subst

```css
tei-subst > tei-del {
  display: none;
  text-decoration: line-through;
}
```

Give add some color and a pointer cursor

```css
tei-subst > tei-add {
  color: blue;
  cursor: pointer;
}
```
User interaction

Now add interaction via JavaScript

```
<script>
CETEIcean.getHTML5("data/example.xml", function(data) {
    document.getElementById("TEI").appendChild(data)

    var substitutions = document.querySelectorAll("tei-subst")
    substitutions.forEach(function(subst){
        subst.querySelector("tei-add").addEventListener("click", function(){
            var del = subst.querySelector("tei-del")
            if (window.getComputedStyle(del).getPropertyValue("display") == "none") {
                del.style.display = "inline"
            } else {
                del.style.display = "none"
            }
        })
    })
})
</script>
```

Place user interaction code after adding the data to HTML

This can be hard! Feel free to copy and paste.

A number of JavaScript libraries such as JQuery can make this operation simpler.