

Easy TEI Publication with CETEIcean (/sɪ'tiːʃn/)

Raff Viglianti

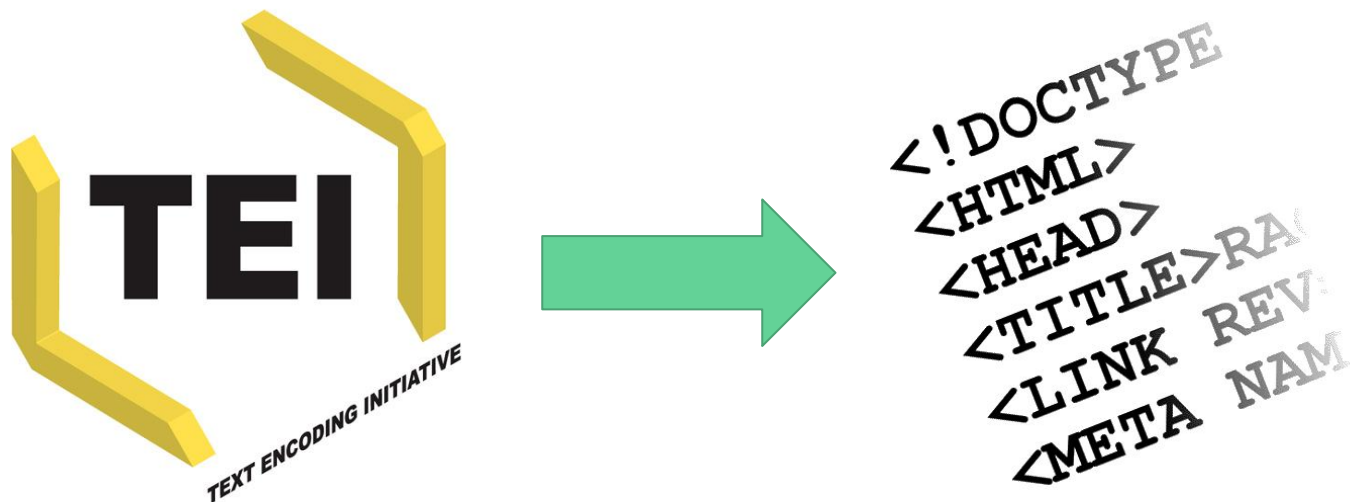
rviglian@umd.edu

Get this presentation:
<http://j.mp/EasyTEI>
(also in shared Drive folder)

Easy publication?


Get this presentation:
<http://j.mp/EasyTEI>
(also in shared Drive folder)

- 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



CETElcean: TEI as HTML

Get this presentation:
<http://j.mp/EasyTEI>
(also in shared Drive folder)

- **CETElcean**  = HTML5 Custom Elements + TEI
(/sɪ'tiːʃn/) <https://github.com/TEIC/CETElcean>
- Custom Elements look like <prefix-elementname>, so <tei-text>, for example.
 - Only supported in modern browsers, but no problem
- CETElcean either loads a TEI XML document and converts it to Custom Elements or takes a pre-converted document and "upgrades" it.

Motivations for CETELcean

- 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

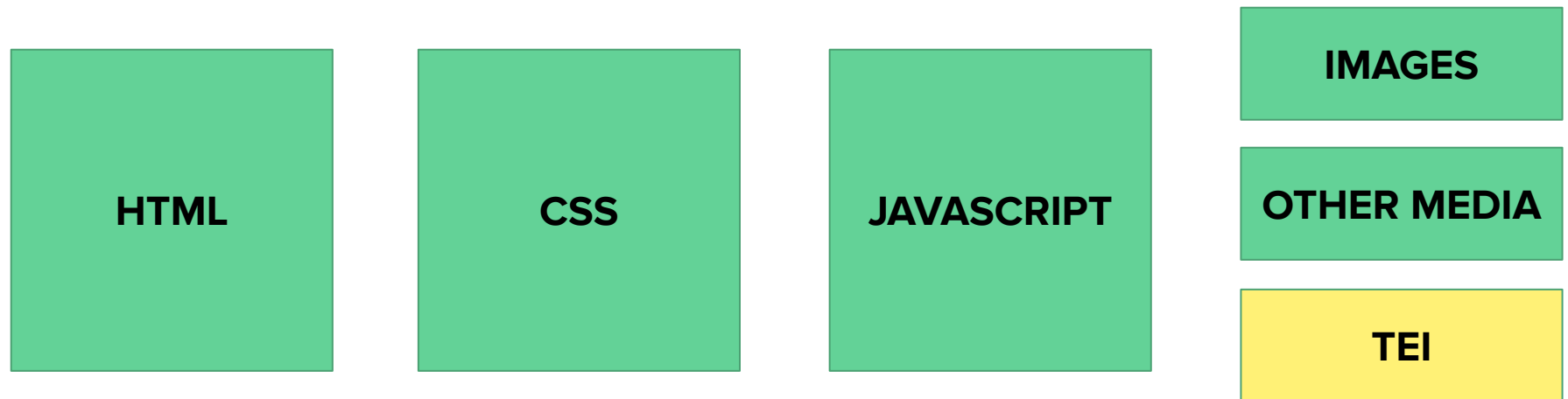
**OTHER
ASSETS**

Creating static sites for TEI documents

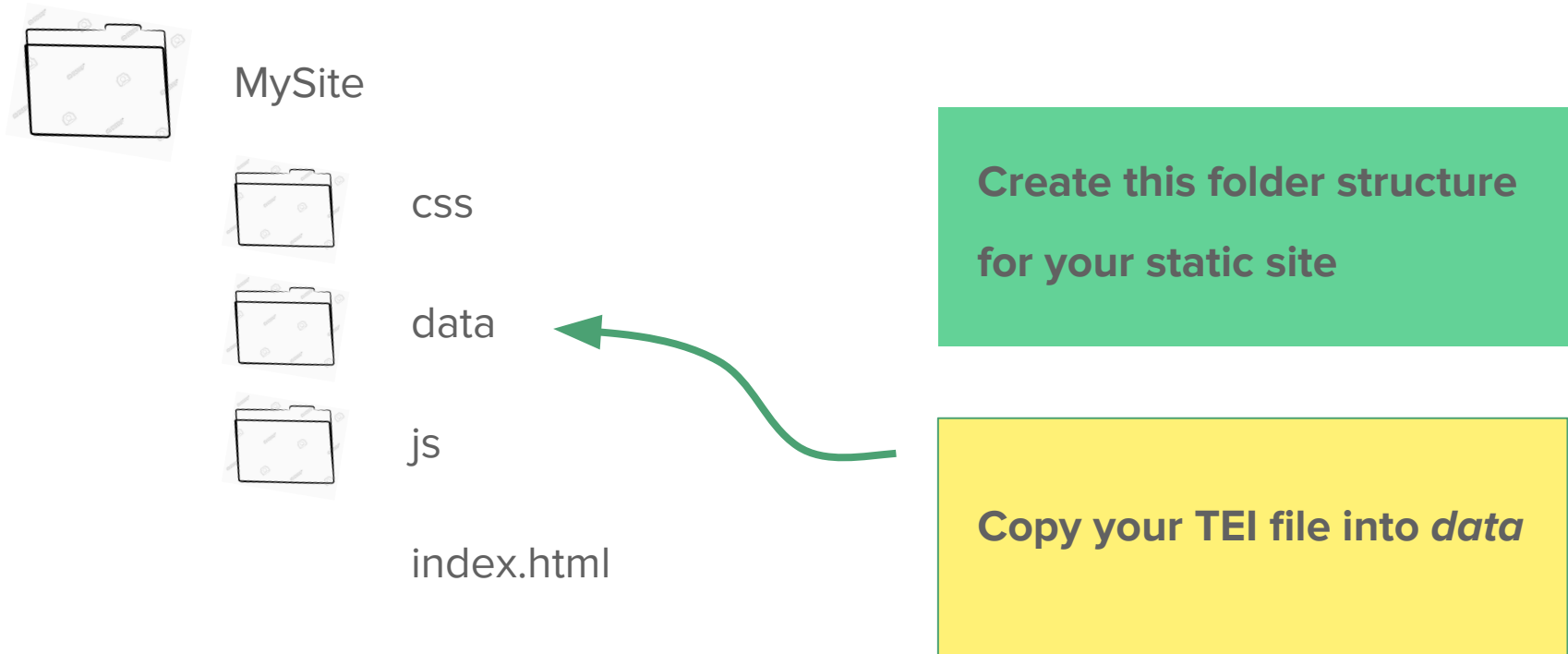
What is a static site?

A web page delivered to the user exactly as stored

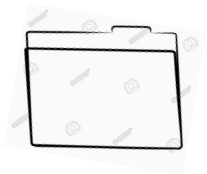
What is it made of?



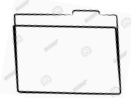
Creating static sites for TEI documents



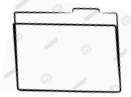
Creating static sites for TEI documents



MySite



css



data



js

index.html

Download CETEIcean from:

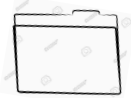
([direct link](#)) or <http://j.mp/CETEI>

Copy CETEI.js into *js*

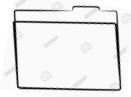
Creating static sites for TEI documents



MySite



css



data



js

index.html

Get CETElcean's example **CSS** file
([direct link](#)) or <https://goo.gl/kwwHPK>

Copy CETElcean.css into css

index.html (also in shared Drive)

```
<!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

```
<!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>
```

Preview with
Ctrl (or ⌘) + Shift + H

Browsers do not allow local JavaScript
to grab files from your computer.
**This only affects testing not
publication**

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>
```

Reference to CSS



Reference to CETEIcean



CETEIcean magic



Loading CETElcean

```
<script>  
  var CETElcean = new CETEI()  
</script>
```

Initialize CETElcean

Loading CETEIcean

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

Tell CETEIcean to load your file

This could be a URL to a TEI file
on the web*

* 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>A
```

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


Loading CETEIcean

```
<script>  
  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.

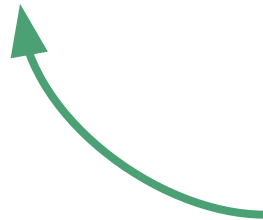
Adding some style

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:

```
<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

Styling with CSS

Writing new styles

- Basic CSS syntax

selector

div

declaration

{ color: red; text-align: center }

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

selector

tei-div

declaration

{ color: red; text-align: center }

Styling with CSS

Let's add a new style declaration for `tei-head` in `style.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: "]"
```

```
}
```

Get this presentation:
<http://j.mp/EasyTEI>

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.

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

 raffazizzi ▾

Repositories

 New

Find a repository...

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


or **HTTPS** **SSH** `git@github.com:raffazizzi/myTEIsite.git` 

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

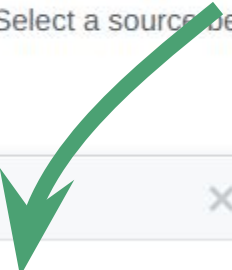
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾ **Save**

Select source

master branch
Use the `master` branch for GitHub Pages.

... using the `master` branch. [Learn more.](#)



Publishing your site with GitHub Pages

- Wait a little bit
- Then visit your website!

<https://USERNAME.github.io/PROJECT>



Behaviors

CETElcean behaviors

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

CETElcean behaviors

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

JS object, registered via `CETEl.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

```
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
`<persName ref="https://viaf.org/viaf/46897170/">Rainer Rilke</persName>`

```
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.

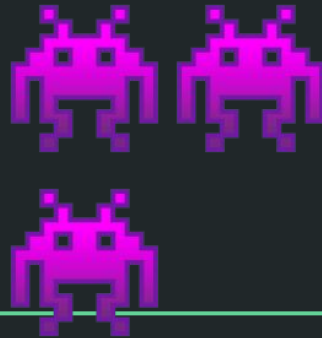
CETELcean behaviors

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

<https://github.com/TEIC/CETELcean/wiki/Anatomy-of-a-behaviors-object>

Get this presentation:
<http://j.mp/EasyTEI>

Play Time!



Work in groups or individually

Prompts:

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

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

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

Give add some color and a pointer cursor

```
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.