Progressive
Enhancement
of Digital Books

let’s take some web content…

c001.html
c002.html
c003.html
mobydick.css
cover.jpg

…put it in a folder…

mobydick

c001.html
c002.html
c003.html
mobydick.css
cover.jpg

This is important, as what makes something a publication is having a boundary*

*more accurately, A *Web Publication* is a *Web Resource* which itself is an aggregated set of interrelated *Web Resources*, and which is intended to be considered as a single *Web Resource*.

add navigation

mobydick

nav.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg

nav.html

```
<nav>
  <ol>
    <li><a href="c001.html">Loomings</a></li>
    <li><a href="c002.html">The Carpet-Bag</a></li>
    <li><a href="c003.html">The Spouter-Inn</a></li>
  </ol>
</nav>
```

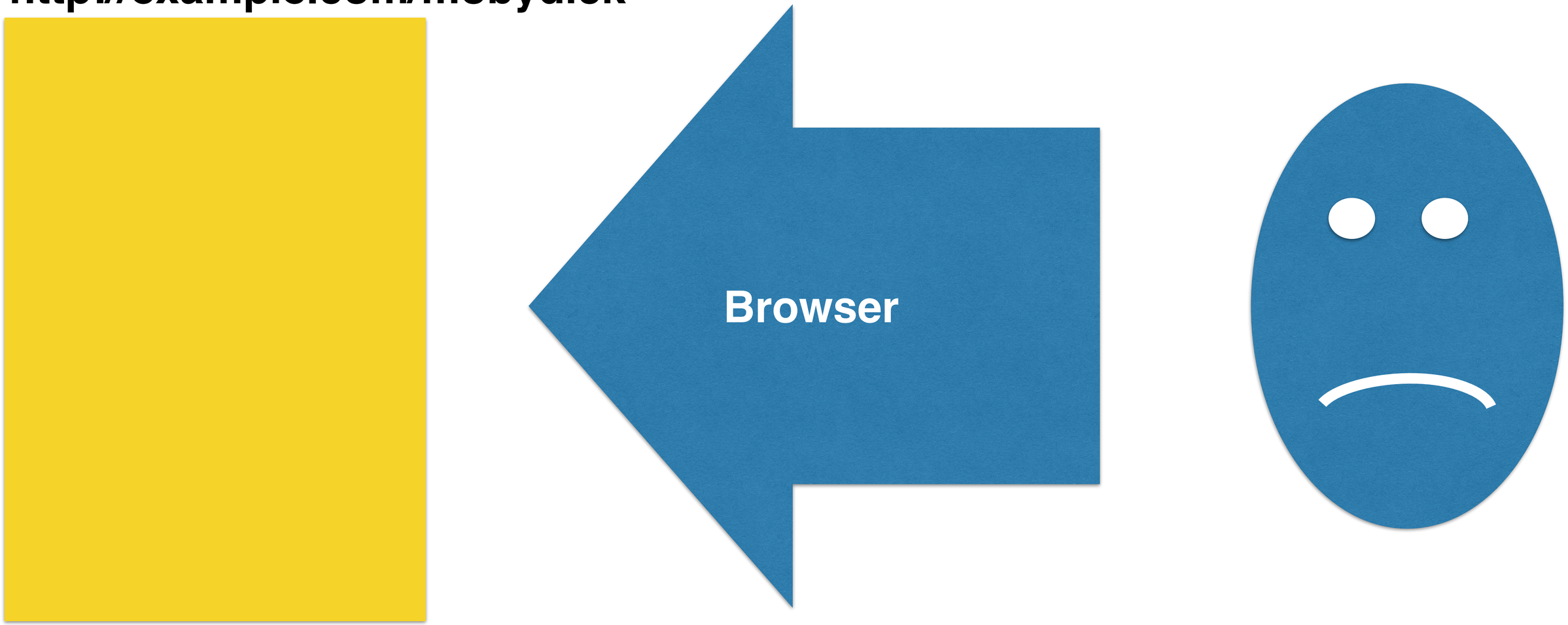
c001.html

```
<link rel="contents" href="index.html">
```

Let's tell the content where to find the navigation

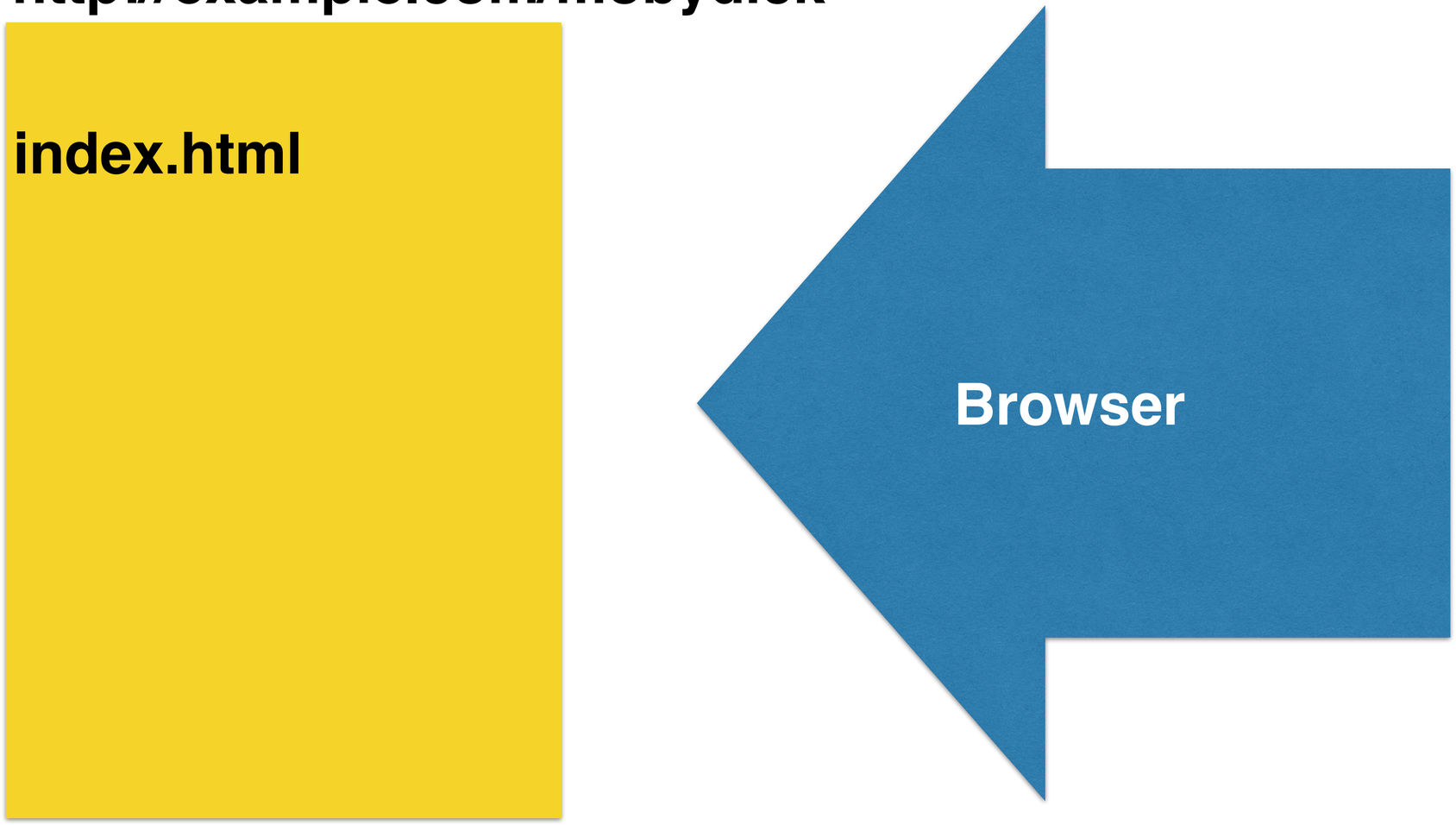
try to view in ordinary browser

http://example.com/mobydick



if nav is called index.html…

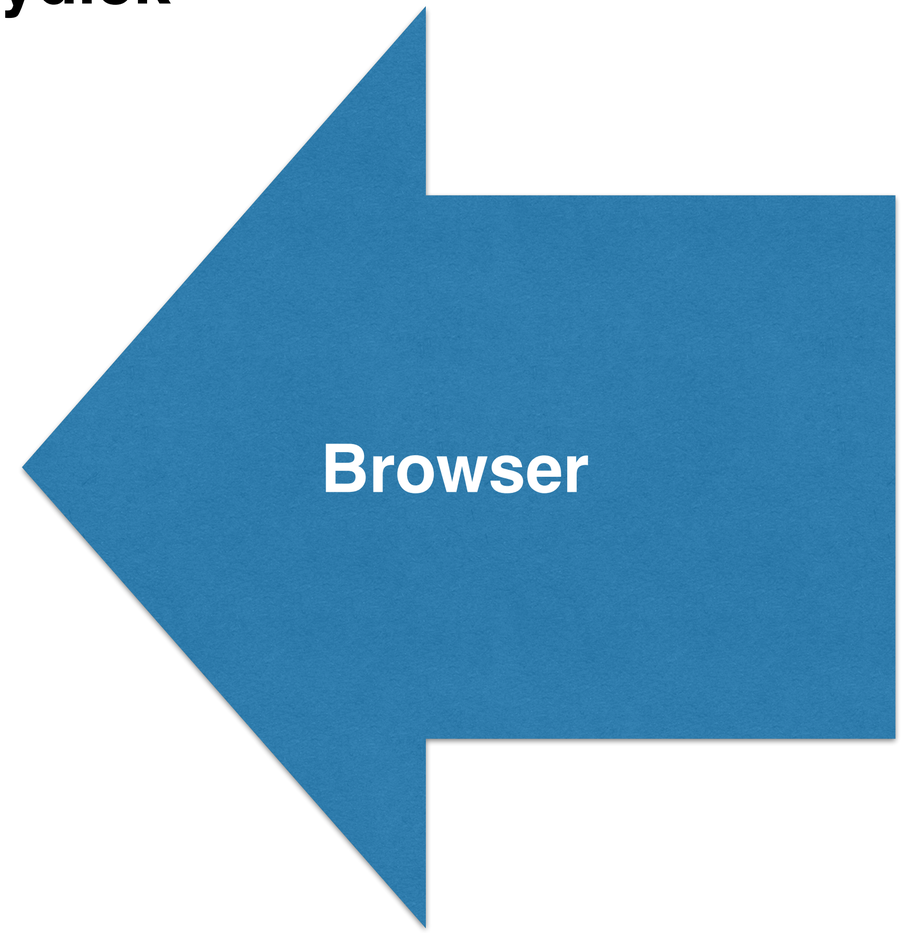
http://example.com/mobydick

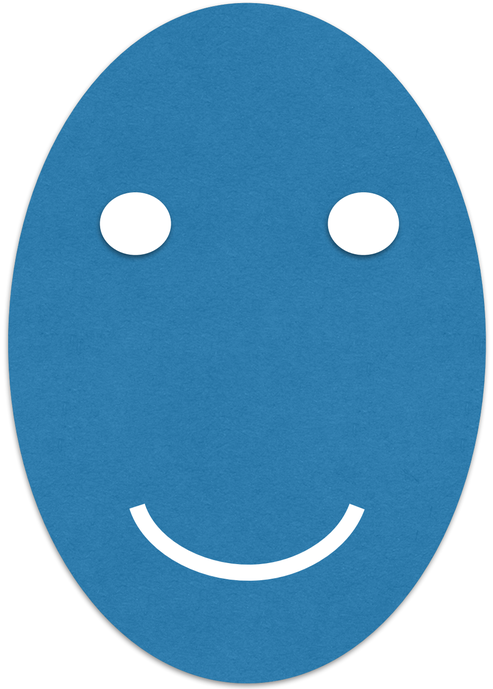


…you can reach all the content!

http://example.com/mobydick

index.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg





but in what order?

http://example.com/mobydick

index.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg

```
<nav>
  <ol>
    <li><a href="c001.html">Loomings</a></li>
    <li><a href="c002.html">The Carpet-Bag</a></li>
    <li><a href="c003.html">The Spouter-Inn</a></li>
  </ol>
</nav>
```

```
<link rel="prev" href="c001.html">
<link rel="next" href="c003.html">
```

Why don't all browsers support these link relationships?

Let’s do some progressive enhancement.

add manifest for offline reading

http://example.com/mobydick

index.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg
pub-manifest.json

```
{
  "title": "Moby-Dick",
  "edition": 1,
  "assets": [
    "",
    "index.html",
    "mobydick.css",
    "cover.jpg",
    "c001.html",
    "c002.html",
    "c003.html"
  ]
}
```

Service Worker

Let's tell the content where to find the manifest

```
<link rel="manifest"
href="pub-manifest.json">
```

Cache

index.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg

add publication metadata

http://example.com/mobydick

index.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg
pub-manifest.json

```
<script type="application/ld+json">

{
  "@context": "http://schema.org",
  "@type": "Book",
  "accessibilityAPI": "ARIA",
  "accessibilityControl": [
    "fullKeyboardControl",
    "fullMouseControl"
  ],
  "bookFormat": "EBook/e0",
  "copyrightHolder": {
    "@type": "Organization",
    "name": "Harper &amp; Row"
  },
  "author": "Herman Melville",
  "datePublished": "1851-10-19",
  "image": "moby-dick-book-
```

We can also link to external metadata

```
<link rel="meta"
href="metadata.json">
```

What if I want to use an EPUB reading system?

just add lots of XML…and ZIP

MobyDick.epub

index.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg
META-INF/container.xml
package.opf
mimetype

```
<?xml version="1.0" encoding="UTF-8" ?>
<container version="1.0" xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
   <rootfiles>
      <rootfile full-path="package.opf" media-type="application/oebps-package+xml"/>
   </rootfiles>
</container>
```

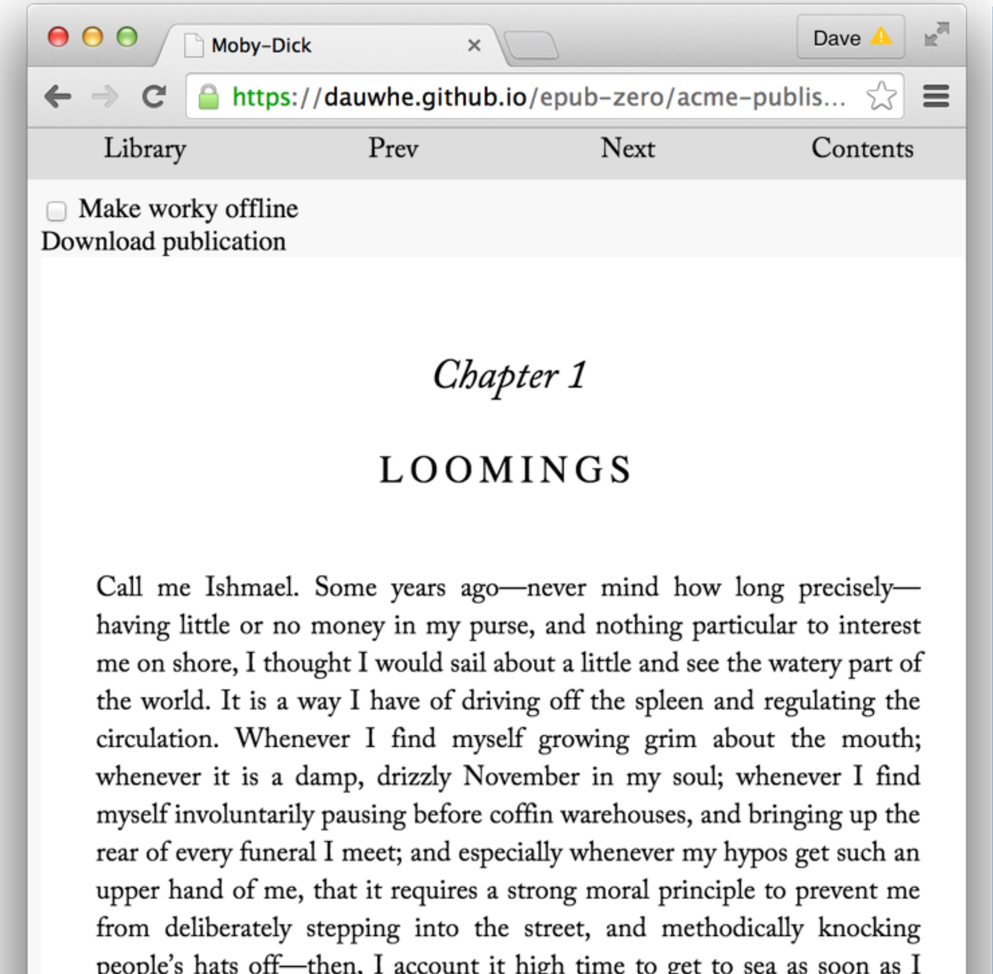
EPUB Reading System

What if I want Books in Browsers?

homebrew user interface

https://example.com/mobydick

index.html
c001.html
c002.html
c003.html
mobydick.css
cover.jpg
pub-manifest.json
acme.js




```
// use arrow keys to navigate from chapter to cha

   document.onkeyup = function(e) { // key presse
    if(document.activeElement.nodeName === "INPUT
    || document.activeElement.nodeName === "TEXTA
        return; // abort if focusing input box
    }

    var elems = document.getElementsByTagName("l
        links = {};

    for(var i = 0; i < elems.length; i++) { // f
        var elem = elems[i];
        if(elem.rel === "prev") { // add prev to
            links.prev = elem;
        } else if(elem.rel === "next") { // ad ne
            links.next = elem;
        }
    }

    if(e.keyCode === 37) { // left key
        location.href = links.prev.href;
    } else if(e.keyCode === 39) { // right key
        location.href = links.next.href;
    }
};
```

PWP vs EPUB vs Web

Layer	PWP	EPUB	Web
user interaction	javascript		
fragment identifiers	unknown	epub CFI	#id
archive	unknown	container specification	no standard
renditions / collections	unknown	container.xml + new package(s)	n/a
user interface	browser + JS?	reading system	browser + JS?
offline	manifest + SW		manifest + SW
design	CSS		
additional a11y semantics	aria roles		
content sequence	link relations	package > spine	link relations
navigation	nav	nav	nav (optional)
content	HTML + SVG + MathML + images		