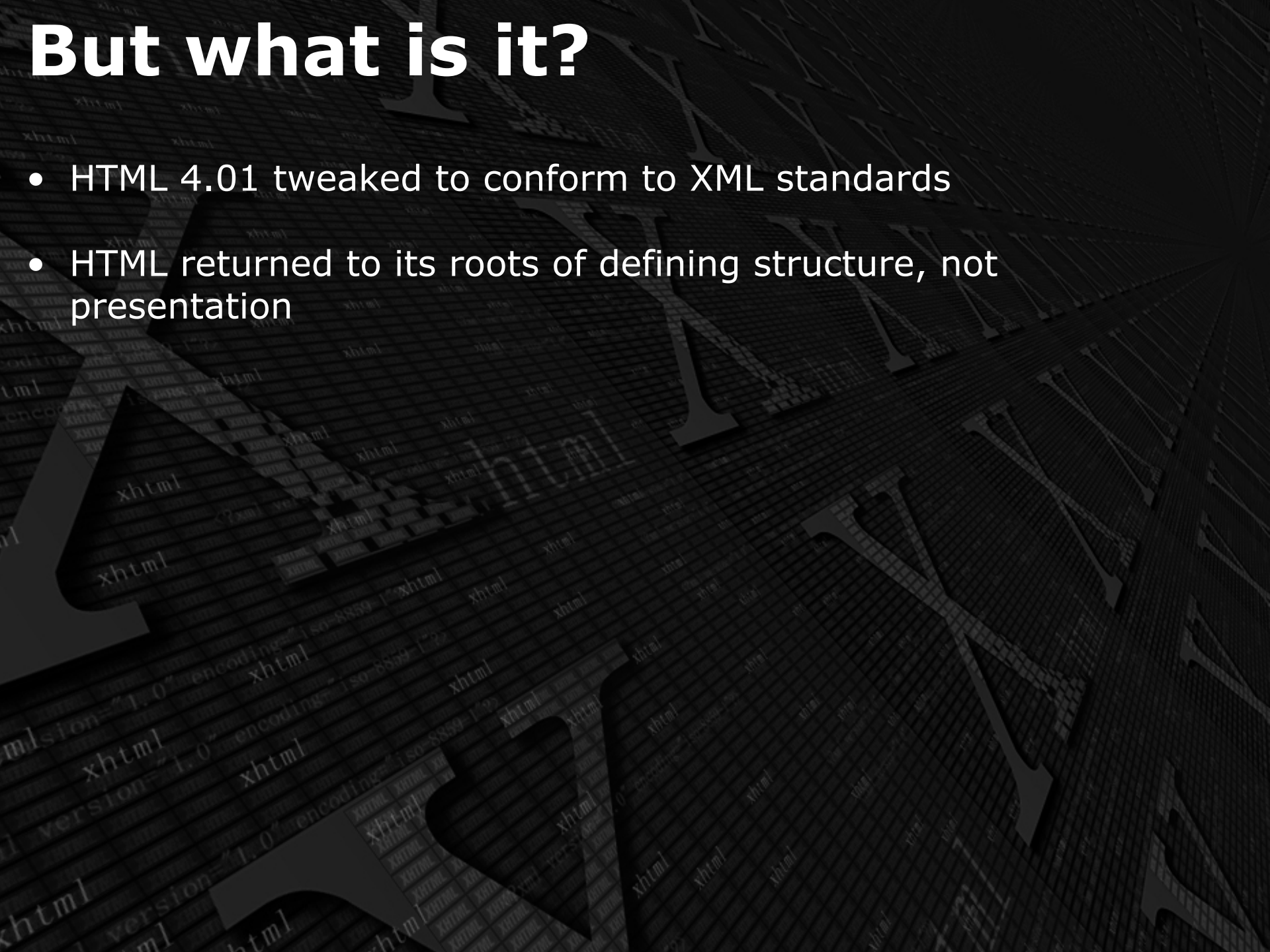

XHTML
XML for Client Side Developers

But what is it?

- HTML 4.01 tweaked to conform to XML standards
- HTML returned to its roots of defining structure, not presentation



Why use it?

- Consistency
- Efficiency
- Compatibility
- Accessibility

Consistency

- As an XML-based language, XHTML is standardized
- Markup is structural, not presentational
- Pages follow a logical outline
- Results are more predictable across browsers

Efficiency

- Code is free of presentational hacks
- Structure and presentation (and behavior) are no longer integrated
- Pages are easier to markup and update
- One document to serve them all
  - » Web
  - » Print
  - » Palm
  - » Non-visual media
- Consistency in structure speeds troubleshooting

Compatibility

- Backward compatible
- Forward compatible
- Compatible with other XML-based languages, applications and protocols

Accessibility

- One document to serve them all
- Logical structure makes pages easier to follow without visual cues

Learn 'em, Live 'em, Love 'em

1) Use a DOCTYPE

2) Avoid deprecated tags

3) Nest tags correctly

4) Close all tags

5) All tags & attributes in lower case

6) All attributes must have values which are quoted

7) All special characters must be encoded

Tags best left behind

Deprecated Tags

`<font>`	`<xmp>`	`<isindex>`
`<applet>`	`<listing>`	`<strike>`
`<u>`	`<blink>`	`<s>`
`<center>`	`<plaintext>`	`<dir>`
`<basefont>`	`<menu>`	`<nextid>`

Presentational Tags

`<b>`	`<i>`

Browser-specific Extensions

`<embed>`	`<multicol>`	`<layer>`
`<spacer>`	`<marquee>`	`<ilayer>`
`<bgaudio>`		

Dust these off

`<abbr>` for abbreviations:

```
<abbr title="continued">cont.</abbr>
```

`<acronym>` for acronyms:

```
<acronym title="Extensible Hypertext Markup Language">XHTML</acronym>
```

`<dfn>` for introducing new terms or special phrases to users (to be used the first time encountered):

```
I used an <dfn>oscilloscope</dfn> yesterday in the lab. Oscilloscopes are part of a wide range of...
```

Were you raised in a barn?

- All opened tags must be closed:
  ```
  <p>Learning XHTML is easy when you know the rules. It also helps to keep in mind the time you will inevitably save by not having to code several versions.</p>
  ```
- All empty tags must be as well:
  - Image tags:
    ```
    <img src="/foo/bar.jpg" alt="a picture of a bar" />
    ```
  - Line breaks:
    ```
    <br />
    ```
  - Horizontal rules:
    ```
    <hr />
    ```

Proper nesting = good pages

- Tags opened last should be closed first:

```
NO: <strong>This phrase is <em>very
    important.</strong></em>
YES: <strong>This phrase is <em>very
    important</em>.</strong>
```

- Obey nesting rules:
  - `<a>` cannot contain `<a>`
  - `<pre>` cannot contain `<img>`, `<object>`, `<big>`, `<small>`, `<sub>` or `<sup>`
  - `<button>` cannot contain `<input>`, `<select>`, `<textarea>`, `<label>`, `<button>`, `<form>`, `<fieldset>`, `<iframe>`, `<isindex>`
  - `<form>` cannot contain `<form>`
  - `<label>` cannot contain `<label>`

e.e. cummings loves XHTML

```
NO:   <IMG SRC="/foo/BAR.jpg" ALT="A picture" />
NO:   <Img Src="/foo/BAR.jpg" Alt="A picture" />

YES:  <img src="/foo/BAR.jpg" alt="A picture" />
```

Note:
It is okay to have MiXed CaSE or ALL CAPS in the attribute values.

Values exist to be quoted

- All values assigned to tag attributes must be quoted:

```
NO: <img src="/foo/BAR.jpg" alt=description />
YES: <img src="/foo/BAR.jpg" alt="A picture" />
```

- All attributed must have values:

```
NO: <input type="checkbox" checked />
YES: <input type="checkbox" checked="checked" />
```

  - Some affected attributes:

checked	compact	declare
defer	disabled	ismap
multiple	noresize	noshade
nowrap	readonly	selected

They’re special, act like it

- Encode characters using HTML character entity (`<`) or decimal (`&#60;`) references
- Encode characters in both body copy *and* URLs:

```
No:  <a href="default.asp?foo=bob&bar=yes">link</a>

Yes: <a href="default.asp?foo=bob&amp;bar=yes">link</a>
Yes: <a href="default.asp?foo=bob&#38;bar=yes">link</a>
```

3 Flavors and then some

- XHTML 1.0
  - » Transitional
  - » Frameset
  - » Strict
- XHTML 1.1
- XHTML 2.0

XHTML 1.0 Transitional

- Closest match to HTML 4.01
- Easiest way to move into coding XHTML
- Tolerates presentational markup
- Forgives deprecated elements and attributes
- DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" ↵
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML 1.0 Frameset

- Exactly like XHTML 1.0 Transitional
- Replaces <body> with <frameset>
- DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" ↩
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.0 Strict

- A boot-camp for your code
- No presentational markup
- No deprecated elements and attributes
- DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" ↵
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML 1.1

- More restrictive
- Eliminates all deprecated elements and attributes
- `lang` attribute becomes `xml:lang`
- `name` attribute removed from `<a>` and `<map>`
- DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" ↵
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

XHTML 2.0

- Still in working draft stage at W3C
- In current state…
  - »
 becomes <line>
  - » <img> becomes <object>
  - » <a> becomes <hlink>
- Likely to change before release

MIME Types

- According to the spec, all XHTML pages should be delivered as `application/xhtml+xml` and not `text/html`
- `text/html` is discouraged for XHTML 1.0 and invalid for XHTML 1.1
- Only Mozilla-based browsers understand `application/xhtml+xml`
- Use "content negotiation" to serve pages via the browser-appropriate MIME type if you want

Creating valid XHTML Pt. 1

- XML declaration

- Basic document requirements:
  1) Always include a DOCTYPE statement
  2) Establish your namespace (xmlns) and language (lang)
  3) Define a character set

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" ↵
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title> ... Page Title ... </title>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
  <link rel="stylesheet" type="text/css" media="all" href="/css/main.css" />
  <script language="JavaScript" type="text/javascript" ↵
src="/scripts/main.js"></script>
</head>
<body>


  ... Page content …


</body>
</html>
```

Creating valid XHTML Pt. 2

- Page divisions ( )
  - » Use semantic id names
  - » Avoid using tables for anything apart from tabular data
- Document structure – why use bloated markup when semantic will do?
  - » When marking up a document, think structure not presentation. CSS can be used to make it look how you want.
  - » Use <h1>-<h6> for headlines
  - » Use unordered lists for navigational link lists

```
<body>

<div id="header">
  ... your header here ...
</div>

<div id="pageBody">
  <ul id="navigation">
    <li><a href="/">link 1</a></li>
    <li><a href="/">link 2</a></li>
    <li><a href="/">link 3</a></li>
  </ul>

  <div id="content">
    <h1> ... Page title ... </h1>
    <h2> ... Subhead ... </h2>

    <p> ... paragraph text ... </p>
    <ul>
      <li>list item one</li>
      <li>list item <a href="#">two</a></li>
    </ul>

    <h3> ... Section Head ... </h3>
    <p> ... paragraph text ... </p>
  </div>
</div>

<div id="footer">
  ... your footer here ...
</div>

</body>
```

Side-by-side comparison

Common HTML

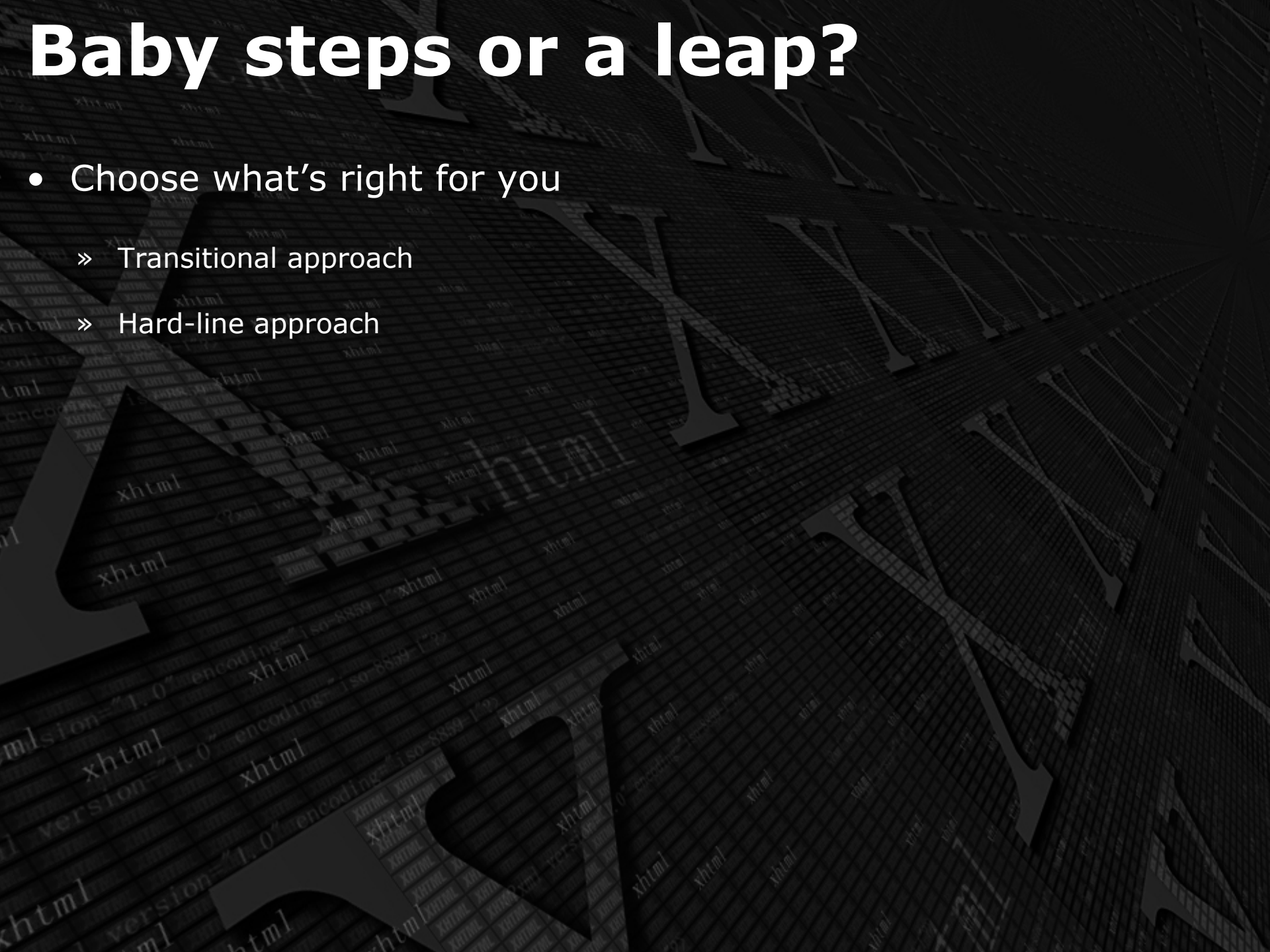
```
<body>
  <table cellpadding="0" cellspacing="0"
border="0" bgcolor="#ffffff">
    <tr>
      <td colspan="2" bgcolor="#000000">
        ... your header here ...
      </td>
    </tr>
    <tr>
      <td bgcolor="#fff000">
        <a href="/"><font face="verdana, arial,
helvetica, sans-serif" size="3"><b>Link
1</b></font></a><br>
        <a href="/"><font face="verdana, arial,
helvetica, sans-serif" size="3"><b>Link
2</b></font></a><br>
        <a href="/"><font face="verdana, arial,
helvetica, sans-serif" size="3"><b>Link
3</b></font></a><br>
        <a href="/"><font face="verdana, arial,
helvetica, sans-serif" size="3"><b>Link
4</b></font></a>
      </td>
      <td>
        <font face="verdana, arial, helvetica,
sans-serif" size="5"><b> ... page title ...
</b></font><br><br>
        <font face="verdana, arial, helvetica,
sans-serif" size="4"><b> ... Subhead ...
</b></font><br><br>
        <p><font face="verdana, arial, helvetica,
sans-serif" size="3"> ... paragraph text ...
</font></p>
                        ...
```

XHTML

```
<body>

<div id="header">
  ... your header here ...
</div>

<div id="pageBody">
  <ul id="navigation">
    <li><a href="/">link 1</a></li>
    <li><a href="/">link 2</a></li>
    <li><a href="/">link 3</a></li>
  </ul>

  <div id="content">
    <h1> ... Page title ... </h1>
    <h2> ... Subhead ... </h2>

    <p> ... paragraph text ... </p>
    <ul>
      <li>list item one</li>
      <li>list item <a href="#">two</a></li>
    </ul>

    <h3> ... Section Head ... </h3>
    <p> ... paragraph text ... </p>
  </div>
</div>

<div id="footer">
  ... your footer here ...
</div>

</body>
```

Baby steps or a leap?

- Choose what's right for you
  - » Transitional approach
  - » Hard-line approach



Taking it slow

- Migrating into XHTML Transitional
- Minimizing `<table>` usage
- Using CSS for presentation

Pros

- Somewhat forward-compatible
- Gets you used to XML-based markup
- Fewer maintenance issues
- Restores some structure to the document

Cons

- Presentation still handled by markup (somewhat)
- Harder to migrate into the site of the future

Diving right in

- XHTML Transitional or Strict
- No `<table>` layouts
- CSS for presentation
- Structure, structure, structure

Pros

- Forward-compatible
- XML-based markup
- Faster & easier maintenance & production
- Document structure intact
- Serve more with less

Cons

- Sites can look plain in older browsers
- Browser support for CSS imperfect


XHTML
XML for Client Side Developers