XML and CSS

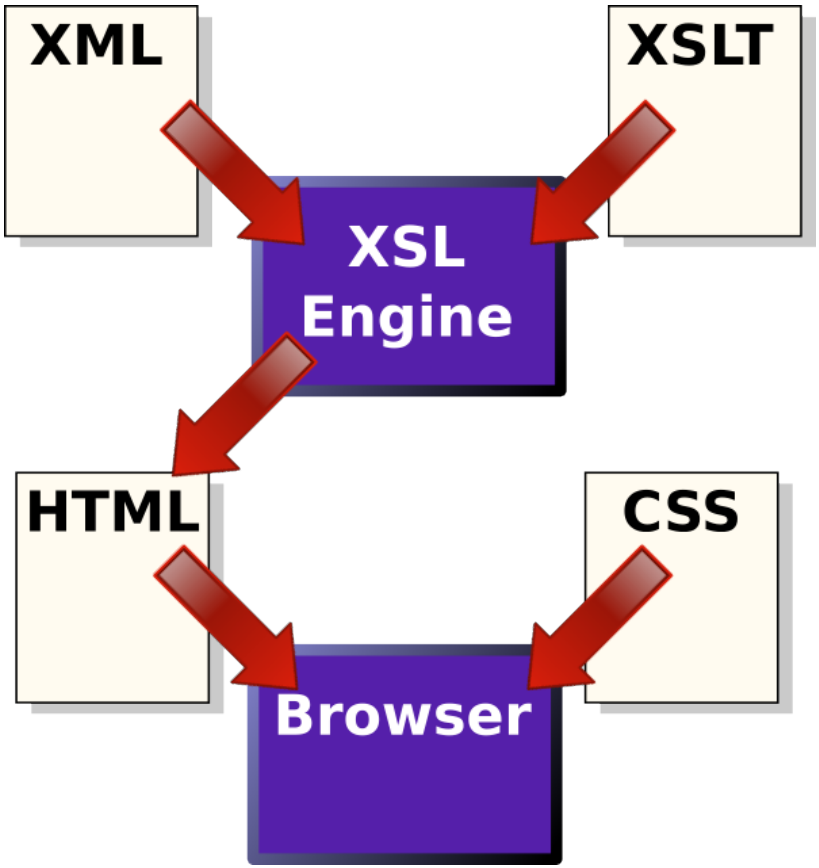
Stewart Arneil & Martin Holmes

University of Victoria Humanities Computing and Media Centre

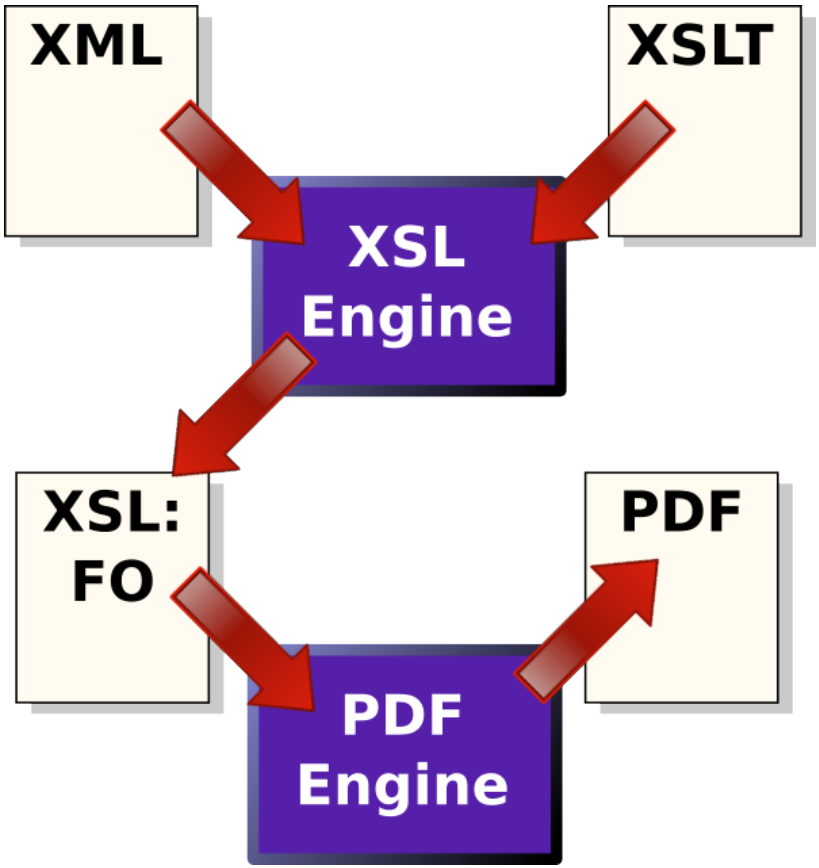
What this workshop will cover

- Why use CSS to present XML?
- How to style XML with CSS
- Caveats: dangers and drawbacks

Conventional approach to presenting XML 1): XSLT → HTML



Conventional approaches to presenting XML: 2) XSLT → FO → PDF



These all involve *changing* the XML into something else

- Core idea: XML is for data storage, not for presentation
- You have to learn other languages (XSLT, XSL:FO, HTML etc.)

And the resulting output is usually styled with CSS anyway

- HTML is styled with CSS
- XSL:FO is described with CSS
- Why not just use pure CSS?

Enough with the acronyms. What are you actually talking about?

- *Cascading Style Sheets*
- *A language for describing what things look like and where they are on the display medium.*

In TEI, there are things like:

-
- <p>
- <name>
- <date>
- <head>
- <list>

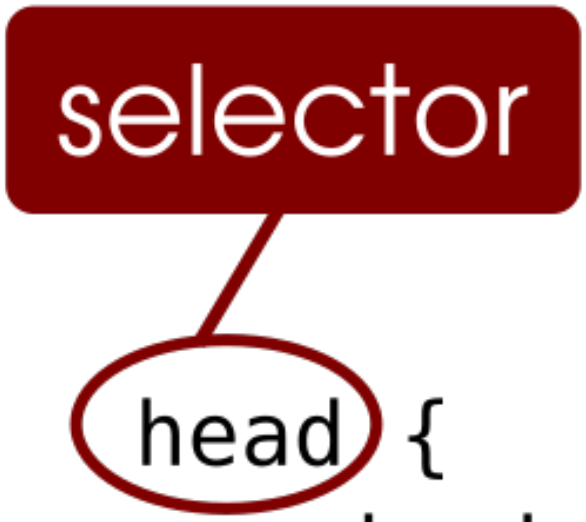
These things form a conceptual document hierarchy...

...as in this Shakespeare sonnet.

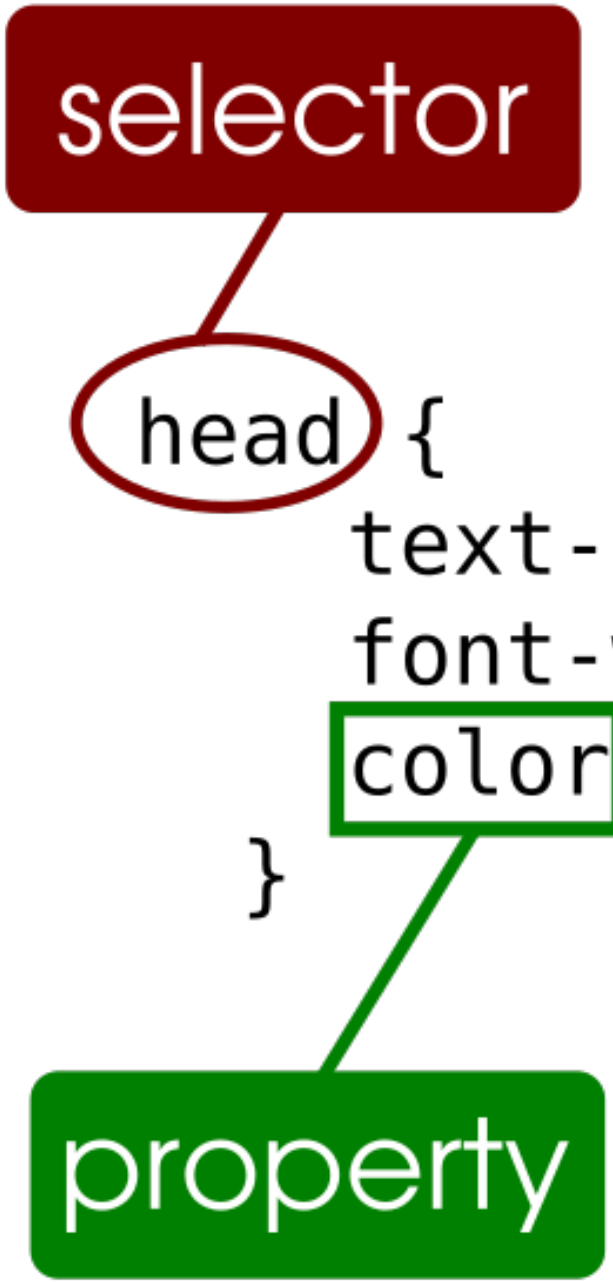
“Separation of concerns”

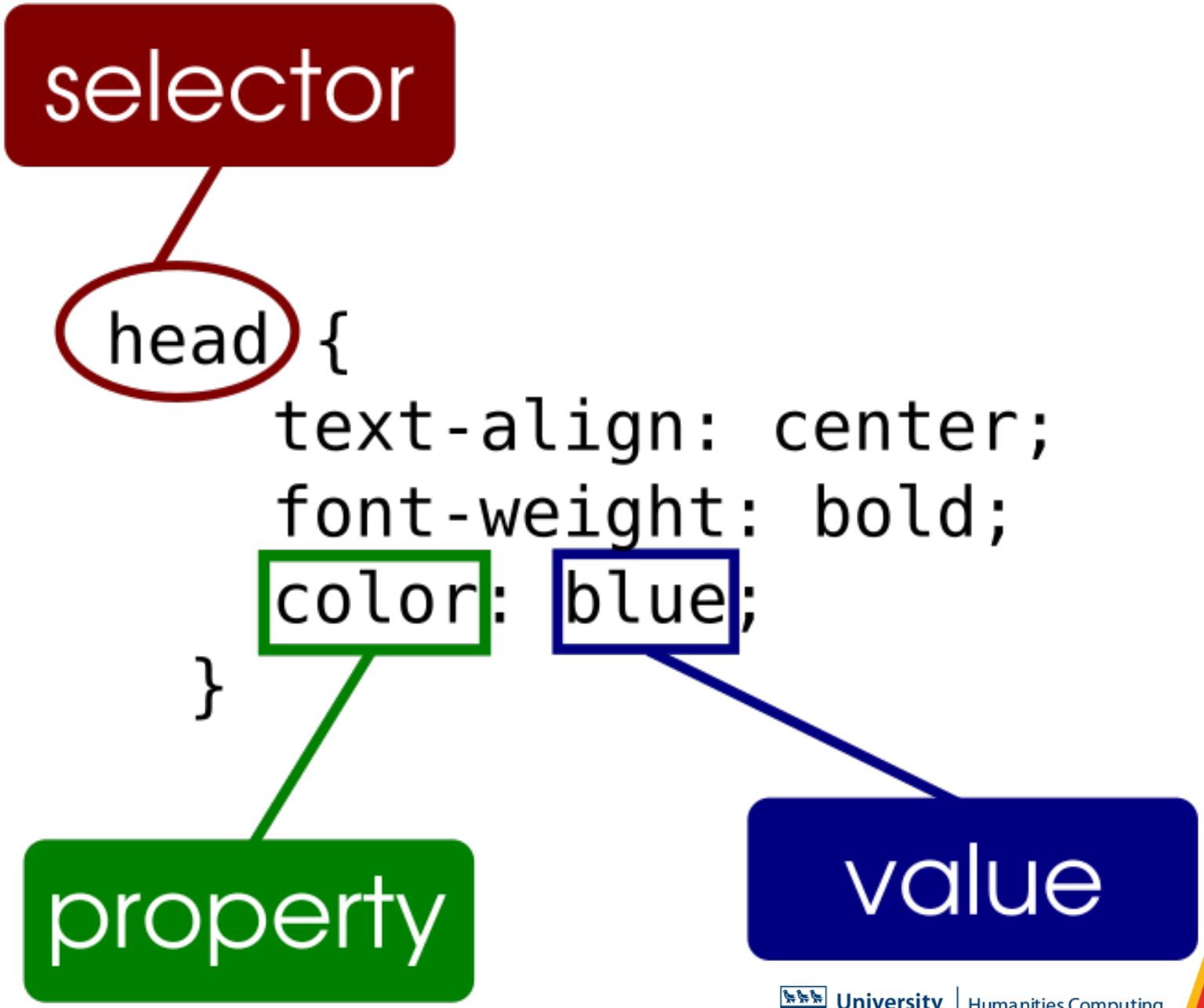
- ...is a key concept in programming:
- Don’t mix *what things are* with *how they appear*. So:
- TEI: *what things are*
- CSS: *how they appear*

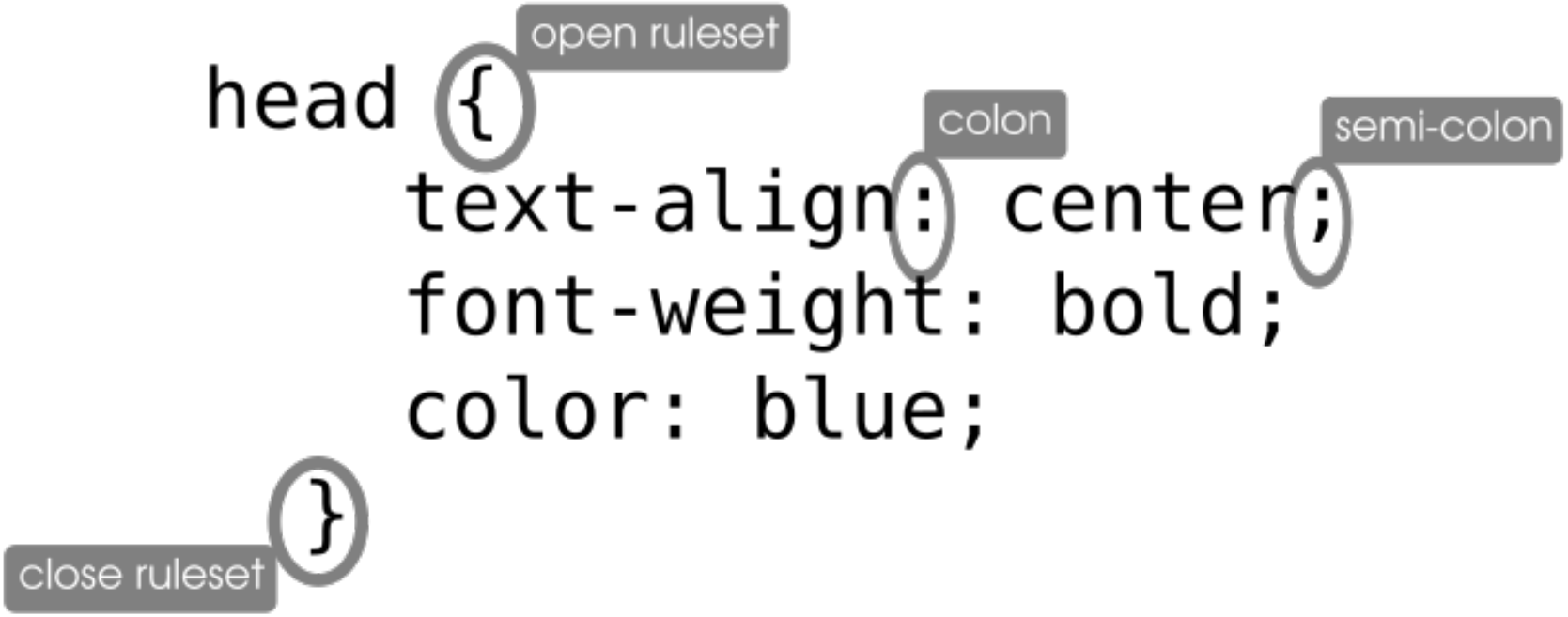
```
head {
    text-align: center;
    font-weight: bold;
    color: blue;
  }
```

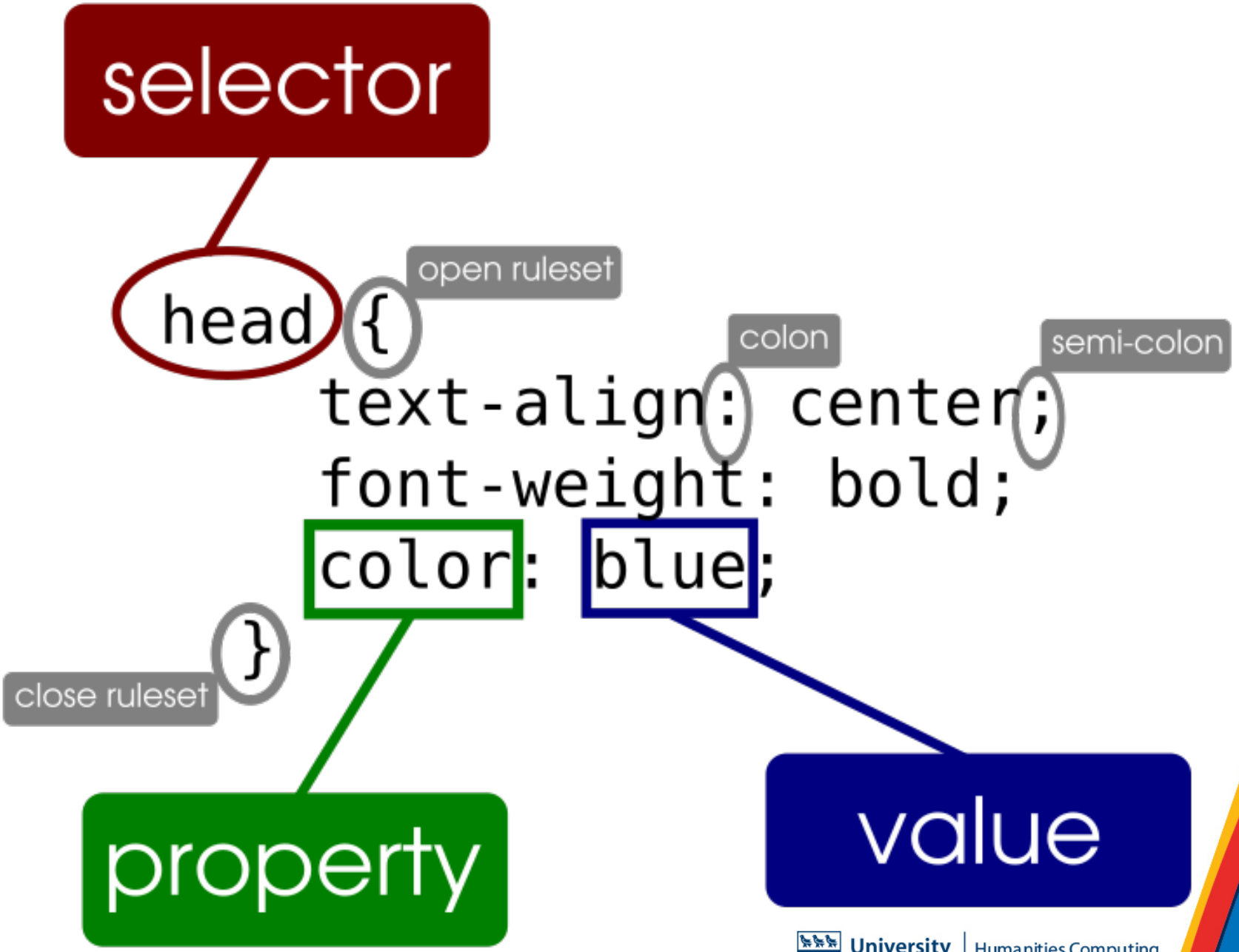



```
head {
    text-align: center;
    font-weight: bold;
    color: blue;
}
```


selector
head {
text-align: center;
font-weight: bold;
color: blue;
}
property


selector
head {
text-align: center;
font-weight: bold;
color: blue;
}
property
value


open ruleset
head {
colon
semi-colon
text-align: center;
font-weight: bold;
color: blue;
}
close ruleset



A worked example

- Download the package from:

http://hcmc.uvic.ca/workshops/xml_css/xml_css.zip

- Unzip it to produce a folder.
- Start Oxygen.
- Click on Project / Open Project
- Find “xml_css.xpr” and choose it.

In Oxygen, open two files:

- css_intro.xml
- css_intro.css

- Now also open css_intro.xml in your browser. What do you see?

We need it to look like this:

Cascading Style Sheets

A very brief introduction

Cascading Style Sheets (**CSS**) provide a method for styling HTML or XML documents. According to the World Wide Web Consortium (**W3C**),

> Style sheets describe how documents are presented on screens, in print, or perhaps how they are pronounced. W3C has actively promoted the use of style sheets on the Web since the Consortium was founded in 1994. The Style Activity has produced several W3C Recommendations (CSS1, CSS2, XPath, XSLT). CSS especially is widely implemented in browsers.
>
> W3C Web Style Sheets Home Page (http://www.w3.org/Style/)

A CSS style sheet consists of a set of *rules*. Each rule has a *selector* and a set of *declarations*. Each declaration has a *property* and a *value*. For example, here is a simple rule:

```
p{
font-size: 12pt;
text-align: justify;
}
```

Here the *selector* is "p". Two *declarations* appear between the parentheses; the first sets the font size to 12 points, and the second sets the text alignment to justified. This means "whenever you find a p tag in the document, render the text inside it at 12 points, and justify it".

Steps in building a stylesheet

Steps in building a stylesheet

1. Specify which elements to *hide*.

Steps in building a stylesheet

1. Specify which elements to *hide*.
2. Specify which elements are *blocks*.

Steps in building a stylesheet

1. Specify which elements to *hide*.
2. Specify which elements are *blocks*.
3. Set *margins* on block elements.

Steps in building a stylesheet

1. Specify which elements to *hide*.
2. Specify which elements are *blocks*.
3. Set *margins* on block elements.
4. Set *text alignment* on block elements.

Steps in building a stylesheet

1. Specify which elements to *hide*.
2. Specify which elements are *blocks*.
3. Set *margins* on block elements.
4. Set *text alignment* on block elements.
5. Set *font size* on block elements.

Steps in building a stylesheet

1. Specify which elements to *hide*.
2. Specify which elements are *blocks*.
3. Set *margins* on block elements.
4. Set *text alignment* on block elements.
5. Set *font size* on block elements.
6. Style *inline* elements.

Caveats, dangers and limitations

Caveats, dangers and limitations

- No interactivity (without adding scripts)

Caveats, dangers and limitations

- No interactivity (without adding scripts)
- Print support in browsers is rather flaky

Caveats, dangers and limitations

- No interactivity (without adding scripts)
- Print support in browsers is rather flaky
- Images can’t easily be displayed (without using the XHTML namespace or scripting hacks)

Caveats, dangers and limitations

- No interactivity (without adding scripts)
- Print support in browsers is rather flaky
- Images can’t easily be displayed (without using the XHTML namespace or scripting hacks)
- CSS selectors are rather limited (compared, say, with XPath)

Useful for:

- Quick and easy static display
- Proof-of-concept for rendering design
- Proofing your encoding
- Oxygen author mode

CSS is:

- ...used in a wide variety of digital publication contexts and tools
- ...increasingly powerful and user-friendly
- ...a bridge between digital documents and our long MS & print tradition (font, em, italic, bold, margin, block,)