

XML Data Management @ unibz - 2010/2011
Chris Mair

Style for XML documents

XSL

- Extensible Stylesheet Language
- <http://www.w3.org/Style/XSL/>
- Consists of the following three parts:
 - XSLT (transformation)
 - XPath (query a.k.a. location)
 - XSL-FO (formatting)

CSS

- Cascading Style Sheets
- <http://www.w3.org/Style/CSS/>
- Commonly associated with web-pages (HTML or XHTML), but can be applied to any XML document
- CSS level 3 is being pushed in modules
<http://www.w3.org/TR/css3-roadmap/>

So, which one?

- While XSLT and XPath is widely used, XSL-FO never “took off” the way CSS did
- XSL-FO is often used in systems for preparing technical documentation (such as the docbook-based ones), but isn’t hardly used elsewhere
- As content is more and more targeted primarily to the web, CSS is gaining importance
- Some shortcomings CSS had with regard to print layout are being addressed with level 3 of the Paged Media module

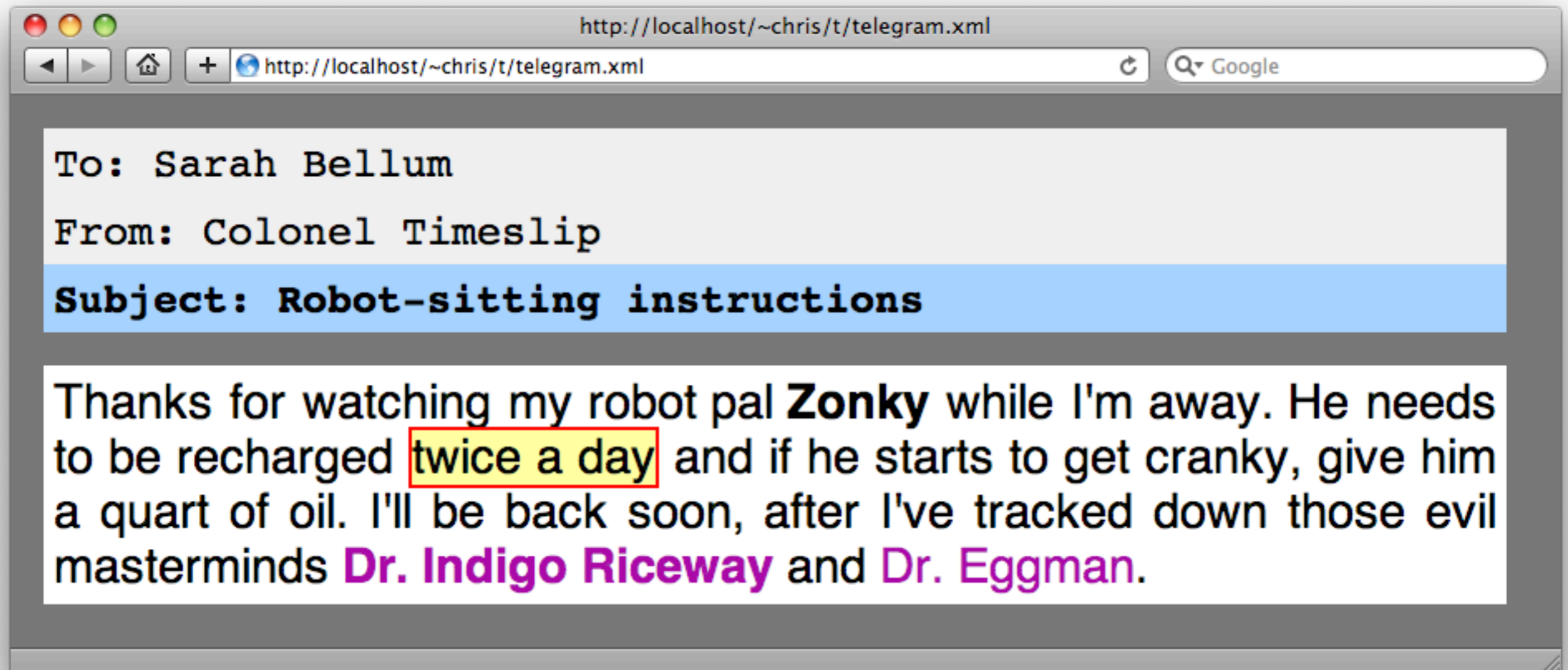
CSS

- Contrary to popular belief, CSS can be applied to any XML document, not just HTML / XHTML
- For the impatient: the dot (.) and hash (#) selectors need to be written using a predicate ([class=...] or [id=...]) when CSS is applied to XML documents

Example: telegram

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="telegram.css"?>
<telegram pri="important">
  <to>Sarah Bellum</to>
  <from>Colonel Timeslip</from>
  <subject>Robot-sitting instructions</subject>
  <message>Thanks for watching my robot pal
    <name>Zonky</name> while I'm away.
    He needs to be recharged <emphasis>twice a
    day</emphasis> and if he starts to get cranky,
    give him a quart of oil. I'll be back soon,
    after I've tracked down those evil masterminds
    <villain id="vv001">Dr. Indigo Riceway</villain>
    and <villain id="vv002">Dr. Eggman</villain>.
  </message>
</telegram>
```

Telegram with style



The CSS file

```
to, from, subject, message {  
  display: block;  
  width: 420px;  
  padding: 3px; }
```

```
to:before { content: "To: "; }  
from:before { content: "From: "; }  
subject:before { content:  
  "Subject: "; }
```

```
telegram {  
  background: #666666; }
```

```
to, from, subject {  
  font-family: monospace;  
  font-size: 12px;  
  background: #EEEEEE;  
  margin-left: 10px;  
  margin-right: 10px; }
```

```
to {  
  margin-top: 10px; }
```

```
subject {  
  font-weight: bold;  
  background: #99CCFF; }
```

```
message {  
  font-family: sans-serif;  
  font-size: 14px;  
  background: #FFFFFF;  
  margin: 10px;  
  text-align: justify; }
```

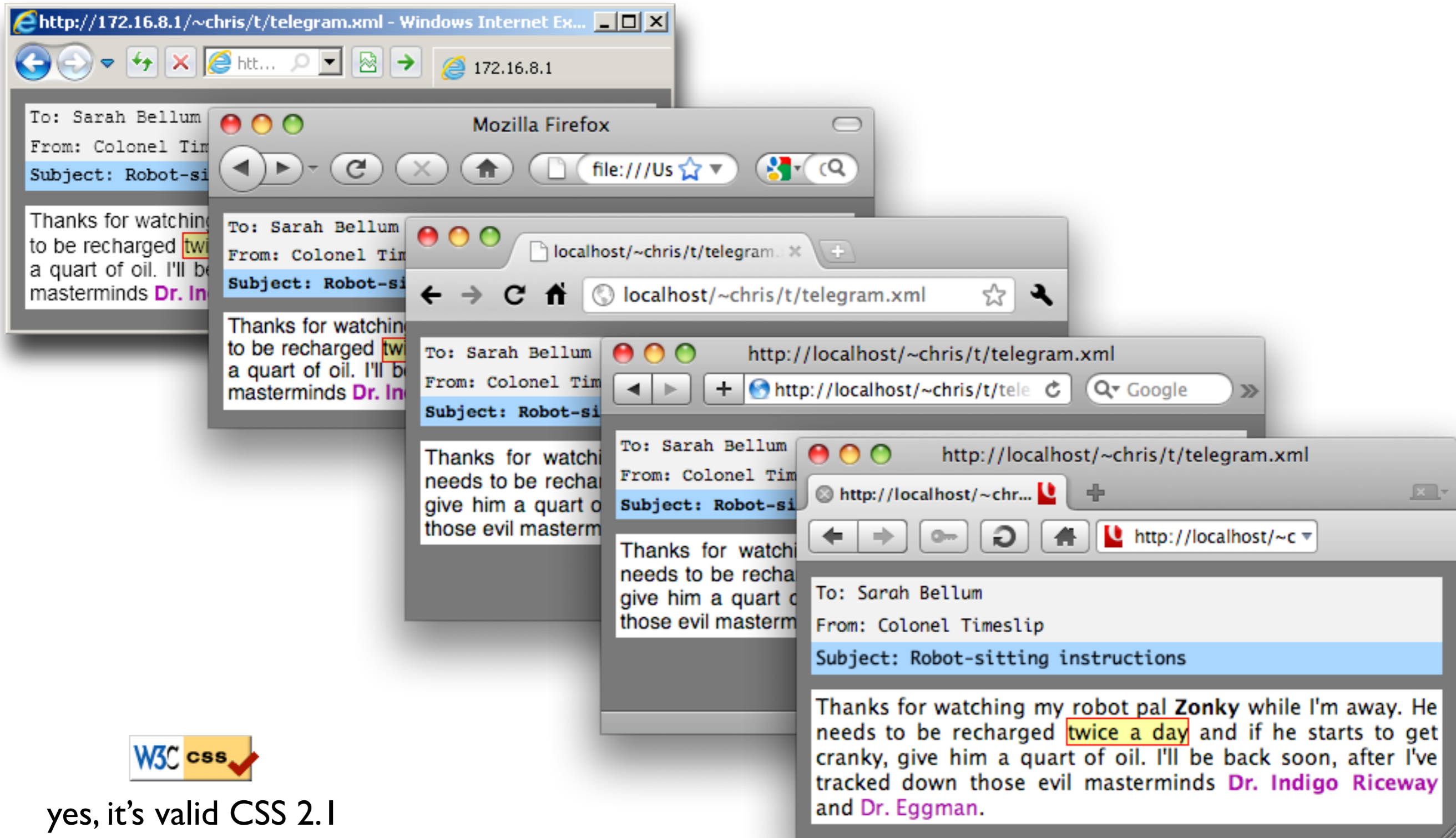
```
name {  
  font-weight: bold; }
```

```
emphasis {  
  background: #FFF999;  
  border: 1px solid #FF0000; }
```

```
villain {  
  color: #990099; }
```

```
villain[id=vv001] {  
  font-weight: bold; }
```

Yeah, but...



yes, it's valid CSS 2.1

The parts explained 1/5

a rule

a selector: here, the named elements are selected

```
to, from, subject, message {
```

```
    display: block;
```

```
    width: 420px;
```

```
    padding: 3px;
```

```
}
```

a declaration: here, the **property** padding is set to the **value** 3px

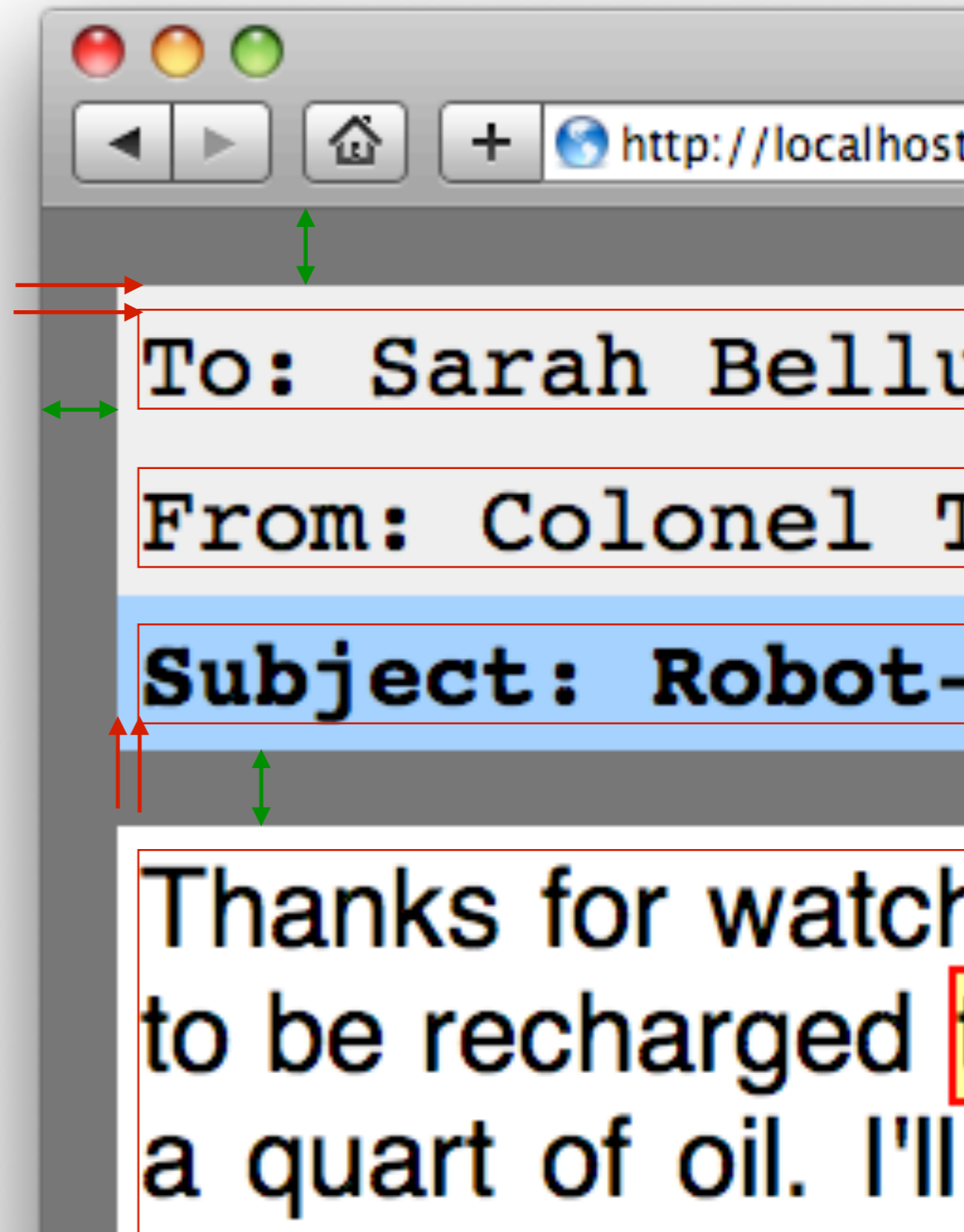
The parts explained 2/5

```
to, from, subject, message {  
  display: block;  
  width: 420px;  
  padding: 3px; }
```

```
to, from, subject {  
  margin-left: 10px;  
  margin-right: 10px; }
```

```
to {  
  margin-top: 10px; }
```

```
message {  
  margin: 10px; }
```



The parts explained 3/5

CSS 2.1 uses an RGB color model

CSS 3 adds a HSL (hue saturation lightness) model

```
background: #666666;  
background: #EEEEEE;  
background: #99CCFF;  
background: #FFFFFF;  
background: #FFFF99;  
border: 1px solid #FF0000;  
color: #990099;
```

#000000



#990099



#99ccff



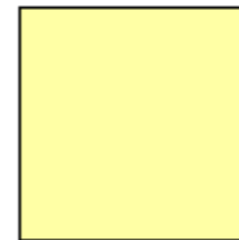
#eeeeee



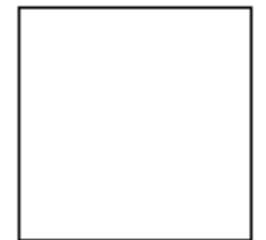
#ff0000



#ffff99



#ffffff



The parts explained 4/5

select *all* villain elements

```
villain {  
    color: #990099; }  
}
```

select only those villain elements that
have an attribute **id** with value **vv001**
(attribute selector)

```
villain[id=vv001] {  
    font-weight: bold; }  
}
```

after I've tracked down those evil masterminds
<villain id="vv001">Dr. Indigo Riceway</villain>
and <villain id="vv002">Dr. Eggman</villain>.



Dr. Indigo Riceway and Dr. Eggman.

The parts explained 5/5

```
to:before      { content: "To: ";      }  
from:before   { content: "From: ";     }  
subject:before { content: "Subject: "; }
```

- **:before** is a pseudo element. It generates content not originally present in the document tree.
- In the telegram case it would have been a better idea to use some templating mechanism or XSLT to add the words. In CSS **:before** is seldomly used, but other pseudo elements (and pseudo classes) are quite common.

Important selectors 1/2

| Selector | Description | CSS |
|------------------|---|----------------|
| * | any element | 2.1 or earlier |
| E | E element | 2.1 or earlier |
| E F | F element descendant of an E element | 2.1 or earlier |
| E > F | F element child of an E element | 2.1 or earlier |
| E + F | F element, immediately preceded by E sibling | 2.1 or earlier |
| E:first-child | E element, first child of its parent (pseudo class) | 2.1 or earlier |
| E:last-child | E element, last child of its parent (pseudo class) | 3 |
| E:nth-child(n) | E element, n-th child of its parent (pseudo class) | 3 |
| E:nth-of-type(n) | E element, n-th sibling of its type (pseudo class) | 3 |
| E:first-letter | first letter of an E element (pseudo element) | 2.1 or earlier |
| E:before | pseudo element before E element | 2.1 or earlier |
| E:after | pseudo element after E element | 2.1 or earlier |

see full list at <http://www.w3.org/TR/css3-selectors/#selectors>

Important selectors 2/2

| Selector | Description | CSS |
|-----------------------------|---|----------------|
| <code>E[foo]</code> | E element with foo attribute set (regardless the value) | 2.1 or earlier |
| <code>E[foo="bar"]</code> | E element with foo attribute set to value bar | 2.1 or earlier |
| <code>E[foo~="bar"]</code> | E element with foo attribute set to a list of space-separated values, one of which is bar | 2.1 or earlier |
| <code>E[foo^="bar"]</code> | E elem. with foo attribute set to value starting with bar | 3 |
| <code>E[foo\$="bar"]</code> | E elem. with foo attribute set to value ending in bar | 3 |
| <code>E[foo*="bar"]</code> | E elem. with foo attribute containing bar | 3 |
| <code>E:not(s)</code> | E element that does not match simple selector s | 3 |
| <code>E.baz</code> | HTML class selector → <code>E[class~="baz"]</code> | 2.1 or earlier |
| <code>E#baz</code> | HTML id selector → <code>E[id="baz"]</code> , id is unique | 2.1 or earlier |
| <code>E:hover</code> | HTML "mouse over" pseudo class | 2.1 or earlier |

see full list at <http://www.w3.org/TR/css3-selectors/#selectors>

Example: tables

```
<table id="t1"><tr><th colspan="2">table t1</th></tr>
  <tr><td>1</td><td>apple</td></tr>
  <tr><td>2</td><td>peach</td></tr>
  <tr><td>3</td><td>apricot</td></tr>
  <tr><td>4</td><td>cherry</td></tr></table>
```

```
<table id="t2"><tr><th colspan="2">table t2</th></tr>
  <tr><td>1</td><td>apple</td></tr>
  <tr><td>2</td><td>peach</td></tr>
  <tr><td>3</td><td>apricot</td></tr>
  <tr><td>4</td><td>cherry</td></tr></table>
```

```
<table id="x"> <tr><th colspan="2">table x</th></tr>
  <tr><td>1</td><td>apple</td></tr>
  <tr><td>2</td><td>peach</td></tr>
  <tr><td>3</td><td>apricot</td></tr>
  <tr><td>4</td><td>cherry</td></tr></table>
```

```
<table>
  <tr><th colspan="2">table y</th></tr>
  <tr><td>1</td><td>apple</td></tr>
  <tr><td>2</td><td>peach</td></tr>
  <tr><td>3</td><td>apricot</td></tr>
  <tr><td>4</td><td>cherry</td></tr></table>
```

Tables with style

```
/* all th elements */  
th { background-color: #DDDDDD; }
```

#dddddd



```
/* all table elements having an id starting with t */  
table[id^="t"] { background: #FFFEEDD; }
```

#fffeedd



```
/* all tr that are even children of table with id = t1 */  
table#t1 tr:nth-child(2n) { background-color: #FFFFAA; }
```

#ffffaa



```
/* all tables having or lacking attribute id */  
table[id] { border: 4px solid #AAFFAA; }  
table:not([id]) { border: 4px solid #FFAAAA; }
```

#aaffaa



#ffaana



| table t1 | |
|----------|---------|
| 1 | apple |
| 2 | peach |
| 3 | apricot |
| 4 | cherry |

| table t2 | |
|----------|---------|
| 1 | apple |
| 2 | peach |
| 3 | apricot |
| 4 | cherry |

| table x | |
|---------|---------|
| 1 | apple |
| 2 | peach |
| 3 | apricot |
| 4 | cherry |

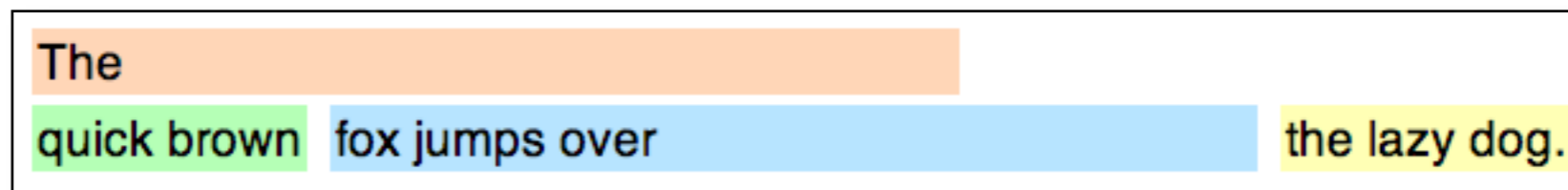
| table y | |
|---------|---------|
| 1 | apple |
| 2 | peach |
| 3 | apricot |
| 4 | cherry |

CSS display property

In HTML `` has default `display: inline` and `<div>` has default `display: block`

```
<span id="s1">The</span>  
<span id="s2">quick brown</span>  
<span id="s3">fox jumps over</span>  
<span id="s4">the lazy dog.</span>
```

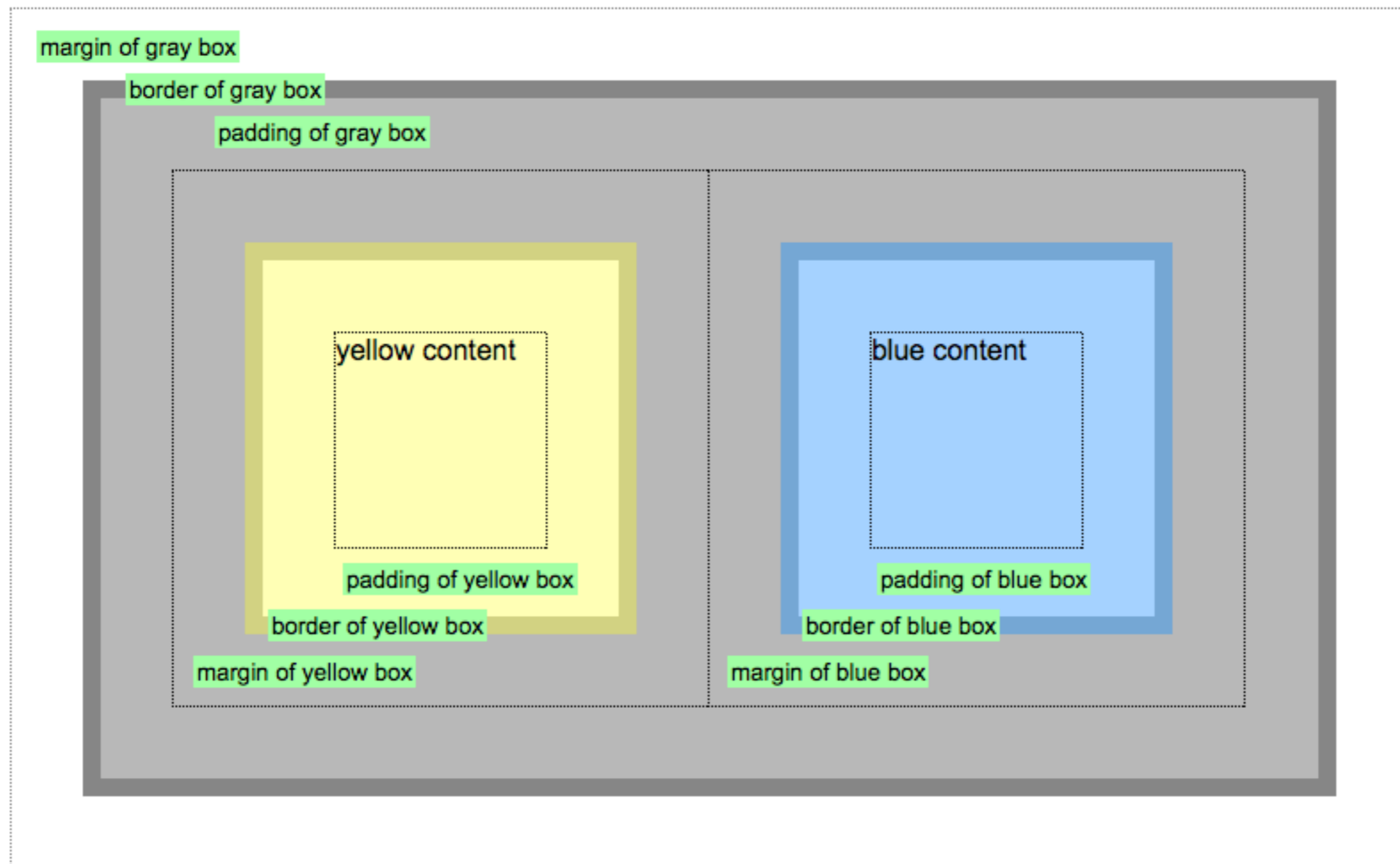
```
/* block => break before and  
after s1, width honoured */  
span#s1 { display: block;  
background: #FFCCAA;  
width: 300px; }  
/* inline => no breaks, width  
ignored for s2 */  
span#s2 { display: inline;  
background: #AAFFAA;  
width: 300px; }  
/* inline-block => no breaks for  
s3, width still honoured */  
span#s3 { display: inline-block;  
background: #AADDFD;  
width: 300px; }  
/* inline => like s2 */  
span#s4 { display: inline;  
background: #FFFFAA;  
width: 300px; }
```



The
quick brown fox jumps over the lazy dog.

CSS box model

the properties *margin*, *border* and *padding*



CSS text flow

```
div#img_01 {
  float: right;
  margin: 0px 0px 10px 20px;
  padding: 5px;
  border: 1px solid black;
  /* CSS 3 */
  box-shadow: #AAAAAA 3px 3px 5px;
}

div.clearer {
  clear: both;
}

p {
  text-align: justify;
}

p:first-letter {
  font-size: 20px;
}

p#par_02 {
  column-count: 2; /* CSS 3 */
}
```



CSS paged media

```
@page { margin: 1cm; }
```

```
@page:first { margin-top: 7cm; }
```

```
@media print {  
  h1 { font-size: 48pt; }  
  p { font-size: 12pt; }  
  div#img_01 {  
    float: right;  
    margin: 0cm 0cm 0.25cm 0.25cm;  
    padding: 0.125cm;  
    border: 1pt solid black;  
  }  
  h2.chapter {  
    page-break-before: always;  
  }  
  /* ... */  
}
```

CSS 2.1:

- @page rules
- media selectors (print, screen)

CSS 3:

- margin boxes
- many other things
- not yet implemented :(

