

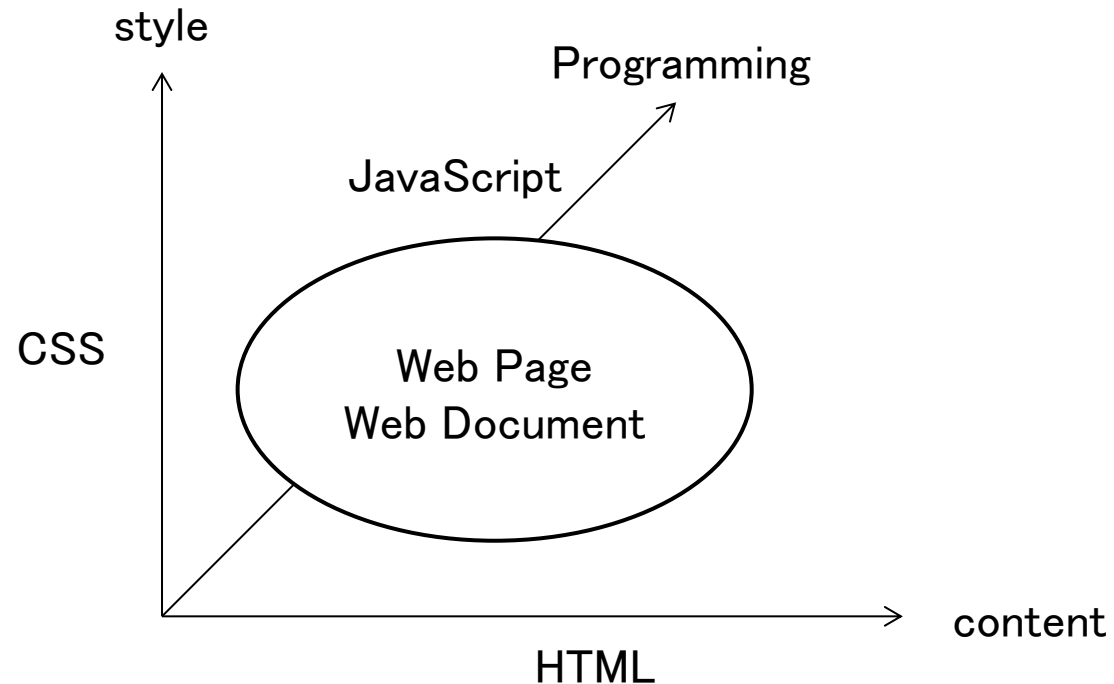
Web Information System Design

No.4 Put Style to Web Documents

Tatsuya Hagino (hagino@sfc.keio.ac.jp)

Web Page Components

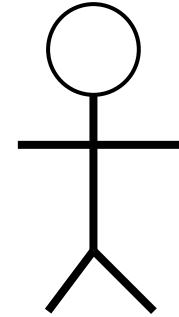
- ▶ Combine orthogonal technologies
 - ▶ content
 - ▶ style
 - ▶ programming



Content and Presentation

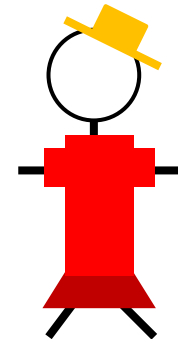
▶ Content

- ▶ Information
- ▶ Data
- ▶ Main part of each Web page



▶ Presentation

- ▶ Style
- ▶ Decoration
- ▶ Design
- ▶ How to show the content



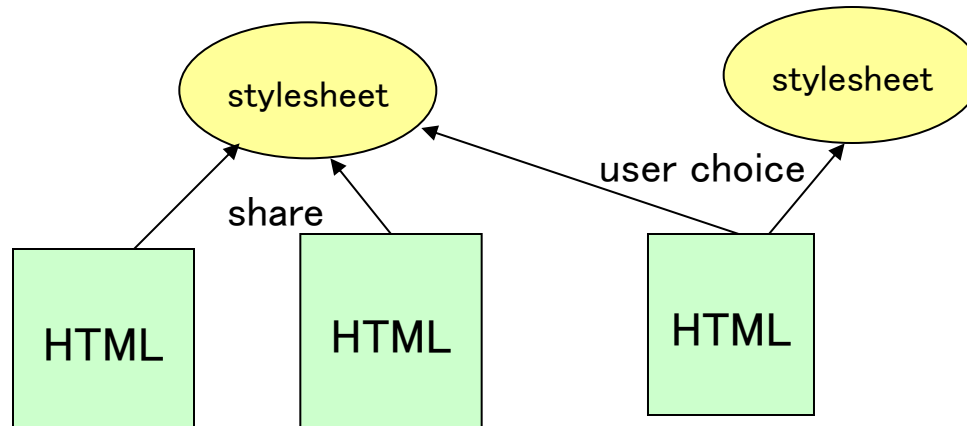
Separation of content and presentation

CSS Zen Garden

<http://www.csszengarden.com/>

▶ Merit of separation

- ▶ The role of HTML is clear.
- ▶ Possible to change style without changing content.
- ▶ Share the same style for multiple documents.
- ▶ Keep the uniformity of the site.
- ▶ User can choose or change the style.
 - ▶ For better accessibility



What is CSS?

- ▶ Give presentation to structured documents
- ▶ Mainly used for HTML documents, but may be used for other XML applications
- ▶ Separation of content and presentation
 - ▶ Easy to write content
 - ▶ Easy to manage the site
 - ▶ Improve accessibility
- ▶ 3 different level of CSS
 - ▶ CSS1 (Cascading Style Sheets, level 1) <http://www.w3.org/TR/REC-CSS1-961217.html>
 - ▶ CSS2 (Cascading Style Sheets, level 2) <http://www.w3.org/TR/REC-CSS2>
 - ▶ CSS3 (Cascading Style Sheets, level 3)

How to specify CSS in HTML

▶ Specify in head section of HTML

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bach's home page</title>
    <style type="text/css">
      h1 { color: blue }
    </style>
  </head>
  <body>
    <h1>Bach's home page</h1>
    <p>Johann Sebastian Bach
      was a prolific composer.</p>
  </body>
</html>
```

▶ Use style attributes

```
<h1 style="color: blue">Bach's
home page</h1>
<p>Johann Sebastian Bach
  was a prolific composer.</p>
```

▶ Link to CSS file

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bach's home page</title>
    <link rel="stylesheet" href="bach.css"
      type="text/css" />
  </head>
  <body>
    <h1>Bach's home page</h1>
    <p>Johann Sebastian Bach
      was a prolific composer.</p>
  </body>
</html>
```

stylesheet

```
h1 {
  color: blue;
  text-align: center;
}
```

How to write CSS?

- ▶ Use selector to specify elements

- ▶ Element name, ID, pattern

- ▶ List attribute and value pair

- ▶ Declarative

```
body {  
    font-family: "Gill Sans", sans-serif;  
    font-size: 12pt;  
    margin: 3em;  
}
```

- ▶ Inheritance and cascading

- ▶ Children inherit parent attribute values
- ▶ Can cascade multiple stylesheets

Declarative vs Procedural

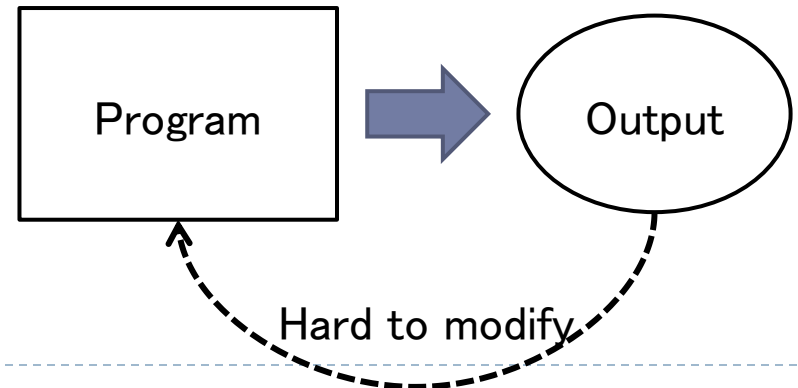
Declarative vs Procedural

▶ Declarative

- ▶ Write description
- ▶ List rules
- ▶ Easy to modify
- ▶ Need to list all the cases

▶ Procedural

- ▶ Write procedure
- ▶ Write programs
- ▶ Hard to modify
- ▶ A small set of primitives



Selector

- ▶ Select elements to which the style is applied
- ▶ Pattern match

Pattern	Meaning
*	Matches any element.
E	Matches any E element (i.e., an element of type E).
E F	Matches any F element that is a descendant of an E element.
E > F	Matches any F element that is a child of an element E.
E:first-child	Matches element E when E is the first child of its parent.
E:link E:visited	Matches element E if E is the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited).
E:active E:hover E:focus	Matches E during certain user actions.

Selector (cont.)

Pattern	Meaning
E:lang(c)	Matches element of type E if it is in (human) language c (the document language specifies how language is determined).
E + F	Matches any F element immediately preceded by an element E.
E[foo]	Matches any E element with the "foo" attribute set (whatever the value).
E[foo="warning"]	Matches any E element whose "foo" attribute value is exactly equal to "warning".
E[foo~="warning"]	Matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning".
E[lang = "en"]	Matches any E element whose "lang" attribute has a hyphen-separated list of values beginning (from the left) with "en".
E.warning	<i>HTML only.</i> The same as E[class~="warning"].
E#myid	Matches any E element ID equal to "myid".



Example of Selector

▶ Grouping

- ▶ `h1, h2, h3 { font-family: sans-serif }`

▶ Descendants

- ▶ `h1 { color: red }`
- ▶ `em { color: red }`
- ▶ `h1 em { color: blue }`

▶ Children

- ▶ `body > p { line-height: 1.3 }`
- ▶ `div ol > li p`

▶ Neighbor

- ▶ `math + p { text-indent: 0 }`
- ▶ `h1 + h2 { margin-top: -5mm }`

▶ Attributes

- ▶ `h1[title] { color: blue; }`
- ▶ `span[class=example] { color: blue; }`
- ▶ `a[rel~="copyright"]`
- ▶ `*[lang="en"] { color : red }`

▶ Class attributes

- ▶ `*.pastoral { color: green }`
- ▶ `.pastoral { color: green }`
- ▶ `*[class~="pastoral"] { color: green }`
- ▶ `j1.pastoral { color: green }`
- ▶ `p.pastoral.marine { color: green }`

▶ ID

- ▶ `h1#chapter1 { text-align: center }`

Pseudo Elements and Pseudo Classes

▶ Pseudo classes

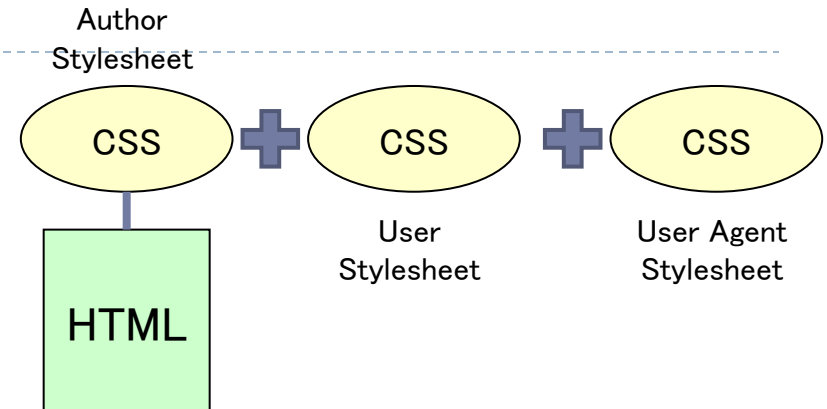
- ▶ `div > p:first-child { text-indent: 0 }`
- ▶ `a:link { color: red } /* unvisited links */`
- ▶ `a:visited { color: blue } /* visited links */`
- ▶ `a:hover { color: yellow } /* user hovers */`
- ▶ `a:active { color: lime } /* active links */`
- ▶ `:lang(fr) > q { quotes: '«' '»' }`

▶ Pseudo elements

- ▶ `p:first-line { text-transform: uppercase }`
- ▶ `p:first-letter { font-size: 200%; font-style: italic; font-weight: bold; float: left }`
- ▶ `h1:before { content: counter(chapno, upper-roman) ". " }`
- ▶ `body:end { content: "The End" }`

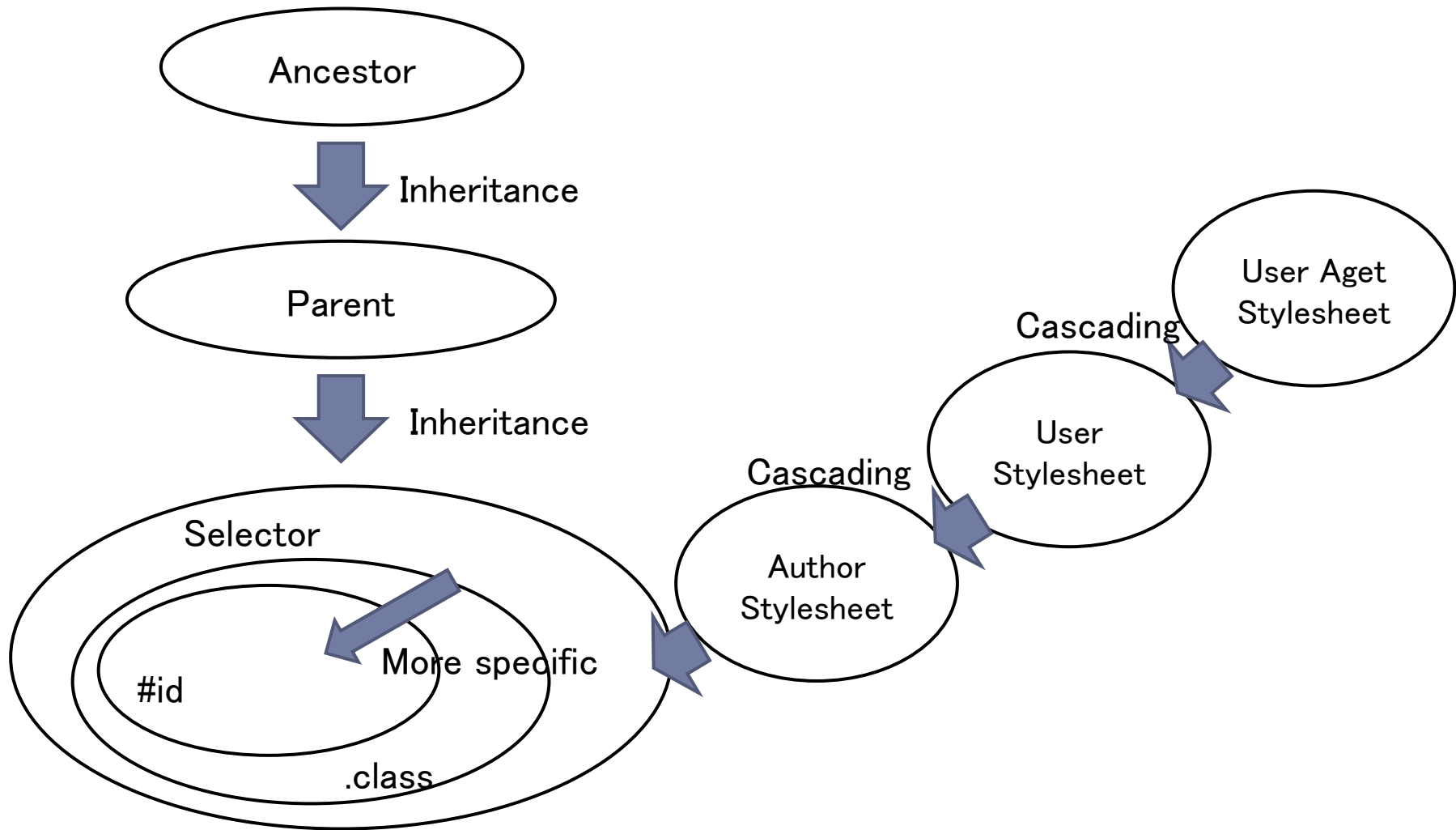
Cascading

- ▶ **Multiple stylesheets**
 - ▶ Author stylesheet
 - ▶ User stylesheet
 - ▶ User Agent (browser) stylesheet
- ▶ **Priority**
 1. User !important value
 2. Author !important value
 3. Author value
 4. User value
 5. Default value
- ▶ **Selector order**
 1. ID specified one (including inline)
 2. Attribute specified one
 3. Element specified one
- ▶ **In case of same order**
 - ▶ Use one which is specified last
 - ▶ @import is treated as inserted first



```
/* From the user's style sheet */  
p { text-indent: 1em ! important }  
p { font-style: italic ! important }  
p { font-size: 18pt }  
  
/* From the author's style sheet */  
p { text-indent: 1.5em !important }  
p { font: 12pt sans-serif !important }  
p { font-size: 24pt }
```

Selector, Inheritance and Cascading



Calculation of Value

1. Specified value

- ▶ Use cascade value
- ▶ Use inherit value from parent element
- ▶ Use default value

2. Calculated value

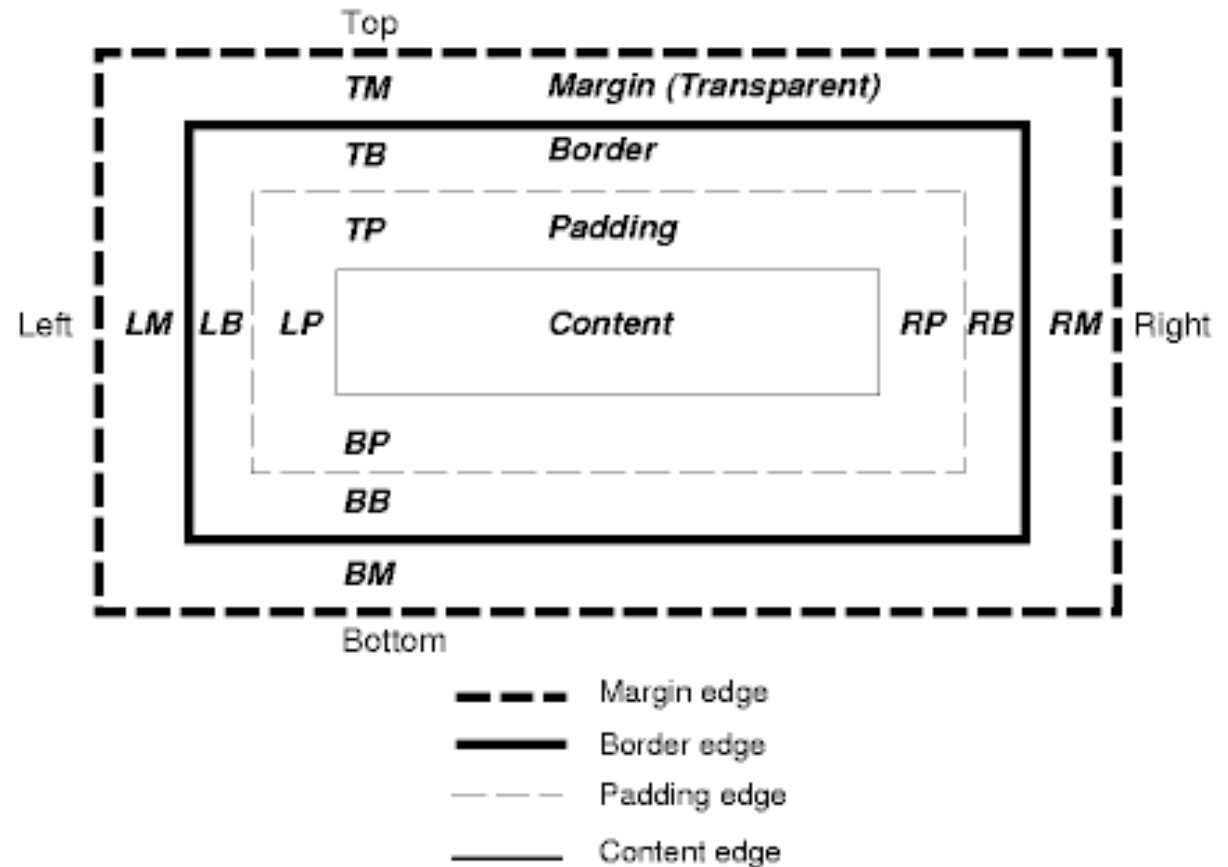
- ▶ For absolute value, use as it is
- ▶ For relative value, calculate absolute value

3. Real value

- ▶ Round to real (possible) value

Box Model

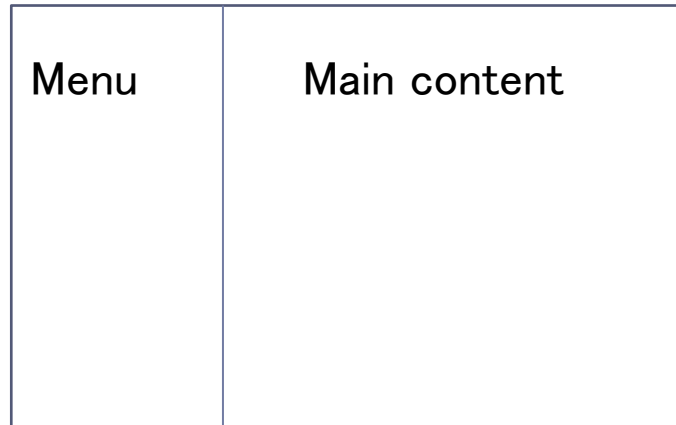
- ▶ block box
 - ▶ for paragraph
 - ▶ vertical concatenation
- ▶ inline box
 - ▶ create lines
 - ▶ horizontal concatenation



Position Algorithm

- ▶ **Containing block**
 - ▶ Rectangle region for rendering child elements
 - ▶ Child blocks are placed in the containing block
 - ▶ Allow to overflow
- ▶ **Initial containing block**
 - ▶ Containing block of the root element
 - ▶ width and height attributes
 - ▶ If width is auto, the width of viewport is used.
 - ▶ If height is auto, the height is automatically extended.
- ▶ **Float boxes**
 - ▶ float:left and float:right moves elements to left or right.
 - ▶ After float boxes are placed, block boxes are placed ignoring float ones.
 - ▶ Inline boxes are placed not overlapping with float boxes.
 - ▶ clear property prevents overlapping block boxes and float boxes
- ▶ **Specify position**
 - ▶ position: static
 - ▶ placed in the normal flow
 - ▶ position: relative
 - ▶ placed in the normal flow and moved relatively
 - ▶ position: absolute
 - ▶ placed relative to the containing box
 - ▶ position: fixed
 - ▶ placed relative to the viewport

Example of Layout



```
<body>
  <section>
    Menu
  </section>
  <article>
    Main content
  </article>
</body>
```

```
section {
  float: left;
  width: 200px;
}

article {
  left-margin: 210px;
}
```

Voice Stylesheet

- ▶ volume
- ▶ speak
- ▶ pause-before, pause-after
- ▶ cue-before, cue-after
- ▶ play-during
- ▶ azimuth, elevation (3 dimensional sound)
- ▶ speech-rate, voice-family, pitch, pitch-range, stress, richness
- ▶ speak-punctuation, speak-numeral

Summary

- ▶ Principle

- ▶ Declarative vs Procedural

- ▶ Stylesheet

- ▶ Separation of content and presentation

- ▶ CSS

- ▶ Selector
- ▶ Cascading
- ▶ Inheritance