# CS 104: Web Technology - II Assignment #3

Anup Adhikari
anup.adhikari@gandakiuniversity.edu.np

Gandaki University December 21, 2022

## 1 Introduction to Media Queries

A media query consists of a media type and zero or more expressions that check for the conditions of particular media features.

HTML4 and CSS2 currently support media-dependent style sheets tailored for different media types. For example, a document may use different style sheets for screen and print. In HTML4, this can be written as:

```
<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css">
<link rel="stylesheet" type="text/css" media="print" href="serif.css">
```

Inside a CSS style sheet, one can declare that sections apply to certain media types:

```
@media screen {
  * { font-family: sans-serif }
}
```

The ’print’ and ’screen’ media types are defined in HTML4. The complete list of media types in HTML4 is: ’aural’, ’braille’, ’handheld’, ’print’, ’projection’, ’screen’, ’tty’, ’tv’. CSS2 defines the same list, deprecates ’aural’ and adds ’embossed’ and ’speech’. Also, ’all’ is used to indicate that the style sheet applies to all media types.

Media-specific style sheets are supported by several user agents. The most commonly used feature is to distinguish between ’screen’ and ’print’.

Future versions of HTML may introduce new values and may allow parameterized values. To facilitate the introduction of these extensions, conforming user agents must be able to parse the media attribute value as follows:

- The value is a comma-separated list of entries. For example,

  **Example 1.** `media="screen, 3d-glasses, print and resolution > 90dpi"`

  *is mapped to*

  ```
  "screen"
  "3d-glasses"
  "print and resolution > 90dpi"
  ```

- Each entry is truncated just before the first character that isn’t a US ASCII letter [a-zA-Z] (Unicode decimal 65-90, 97-122), digit [0-9] (Unicode hex 30-39), or hyphen (45). In the example, this gives:

  ```
  "screen"
  "3d-glasses"
  "print"
  ```

Newer versions of HTML reference the Media Queries specification directly and thus updates the rules for HTML.

## 1.1 Syntax

The media query syntax is described in terms of the CSS2 grammar. An @media rule specifies the target media types (separated by commas) of a set of statements (delimited by curly braces). Invalid statements must be ignored per 4.1.7 "Rule sets, declaration blocks, and selectors" and 4.2 "Rules for handling parsing errors." The @media construct allows style sheet rules for various media in the same style sheet:

```
@media print {
    body { font-size: 10pt }
}
@media screen {
body { font-size: 13px }
}
@media screen, print {
body { line-height: 1.2 }
}
```

```
media_query_list
 : S* [media_query [ ',' S* media_query ]* ]?
 ;
media_query
 : [ONLY | NOT]? S* media_type S* [ AND S* expression ]*
 | expression [ AND S* expression ]*
 ;
media_type
 : IDENT
 ;
expression
 : '(' S* media_feature S* [ ':' S* expr ]? ')' S*
 ;
media_feature
 : IDENT
 ;
```

### 1.1.1 Recognized Media Types

```
all - Suitable for all devices.
braille -Intended for braille tactile feedback devices.
embossed - Intended for paged braille printers.
handheld - Intended for handheld devices (typically small screen, limited bandwidth).
print - Intended for paged material and for documents viewed on screen in print preview mode.
    Please consult the section on paged media for information about formatting issues that
    are specific to paged media.
projection - Intended for projected presentations, for example projectors. Please consult the
    section on paged media for information about formatting issues that are specific to paged media.
screen - Intended primarily for color computer screens.
speech - Intended for speech synthesizers. Note: CSS2 had a similar media type called 'aural'
    for this purpose. See the appendix on aural style sheets for details.
tty - Intended for media using a fixed-pitch character grid (such as teletypes, terminals, or
    portable devices with limited display capabilities). Authors should not use pixel units with
    the "tty" media type.
tv - Intended for television-type devices (low resolution, color, limited-scrollability
    screens, sound available).
```

### 1.1.2 Media Features

Syntactically, media features resemble CSS properties: they have names and accept certain values. There are, however, several important differences between properties and media features:

- Properties are used in declarations to give information about how to present a document. Media features are used in expressions to describe requirements of the output device.

- Most media features accept optional 'min-' or 'max-' prefixes to express "greater or equal to" and "smaller or equal to" constraints. This syntax is used to avoid "<" and ">" characters which may conflict with HTML and XML. Those media features that accept prefixes will most often be used with prefixes, but can also be used alone.

- Properties always require a value to form a declaration. Media features, on the other hand, can also be used without a value. For a media feature feature, (feature) will evaluate to true if (feature:x) will evaluate to true for a value x other than zero or zero followed by a unit identifier (i.e., other than 0, 0px, 0em, etc.). Media features that are prefixed by min/max cannot be used without a value. When a media feature prefixed with min/max is used without a value it makes the media query malformed.

- Properties may accept more complex values, e.g., calculations that involve several other values. Media features only accept single values: one keyword, one number, or a number with a unit identifier. (The only exceptions are the 'aspect-ratio' and 'device-aspect-ratio' media features.)

| **Media features** | **Values** | **Applies To** | **Accepts min/max prefixes** |
|---|---|---|---|
| **width** | length | Visual Media | Yes |
| **height** | length | Visual Media | Yes |
| **device-width** | length | Visual Media | Yes |
| **device-height** | length | Visual Media | Yes |
| **orientation** | portrait\|landscape | Visual Media | No |
| **aspect-ratio** | ratio | Visual Media | Yes |
| **device-aspect-ratio** | ratio | Visual Media | Yes |
| **color** | integer | Visual Media | Yes |
| **color-index** | integer | Visual Media | Yes |
| **monochrome** | integer | Visual Media | Yes |
| **resolution** | resolution | BITMAP media types | Yes |
| **scan** | progressive \| interlace | TV media | No |
| **grid** | integer | Visual & Tactile Media | No |

- The ratio value is positive (not zero or negative) followed by optional whitespace, followed by a solidus("/"), followed by optional whitespace, followed by a positive integer.

```
ratio: 4/5
```

- Resolution units are "dpi" and 'dpcm', ie. the density of device pixels.

- The 'grid' media feature is used to query whether the output device is grid or bitmap. If the output device is grid-based (e.g., a "tty" terminal, or a phone display with only one fixed font), the value will be 1. Otherwise, the value will be 0.

- The 'monochrome' media feature describes the number of bits per pixel in a monochrome frame buffer. If the device is not a monochrome device, the output device value will be 0.

- The 'color-index' media feature describes the number of entries in the color lookup table of the output device. If the device does not use a color lookup table, the value is zero.

## 1.2 Multiple Column Layouts

To flow content into several columns, two new CSS properties are proposed., columns and column-rule.

columns: [length] > number | percentage. column-rule: [width][height] > number | percentage

```
<STYLE>
  DIV  {
    columns:  20em  3em;
    width:  auto;
  }
  H1 { columns: 1 }
</STYLE>

<BODY>
  <H1>THIS IS THE HEADLINE WHICH IS QUITE LONG</H1>
  <DIV>
  <P>This is the first paragraph. The first paragraph comes
      first, before the second.
  <P>After the first paragraph comes the second paragraph
      which you are reading now.
  <P>The third paragraph is the last paragraph. Not much
      more to say about that.
  </DIV>
</BODY>
```

The output will look something like this

```
THIS IS THE HEADLINE WHICH GOES ON TOP

This is the first     graph which you
paragraph. The        are reading now.
first paragraph
comes first, be-      The third para-
fore the second.      graph is the last
                      paragraph. Not
After the first       much more to say
paragraph comes       about that.
the second para-
```

## 1.3 Table Layout

Many types of information (ex: weather readings collected over the past year) are best visually represented in a two-axis grid where rows represent one item of the list (ex: a date, and the various weather properties measured during that day), and where columns represent the successive values of an item's property (ex: the temperatures measured over the past year).

### 1.3.1 Content Model

The CSS table model is based on the HTML4 table model, in which the structure of a table closely parallels the visual layout of the table. In this model, a table consists of an optional caption and any number of rows of cells.

In addition, adjacent rows and columns may be grouped structurally and this grouping can be reflected in presentation (e.g., a border may be drawn around a group of rows).

The instance of table model consists of:

- table-root
  - zero of more table rows, optionally in row groups
    - * Each of them containing one or more table cells.
  - optional: one or more table-columns, optionally in column groups.
  - optional: one or more table caption

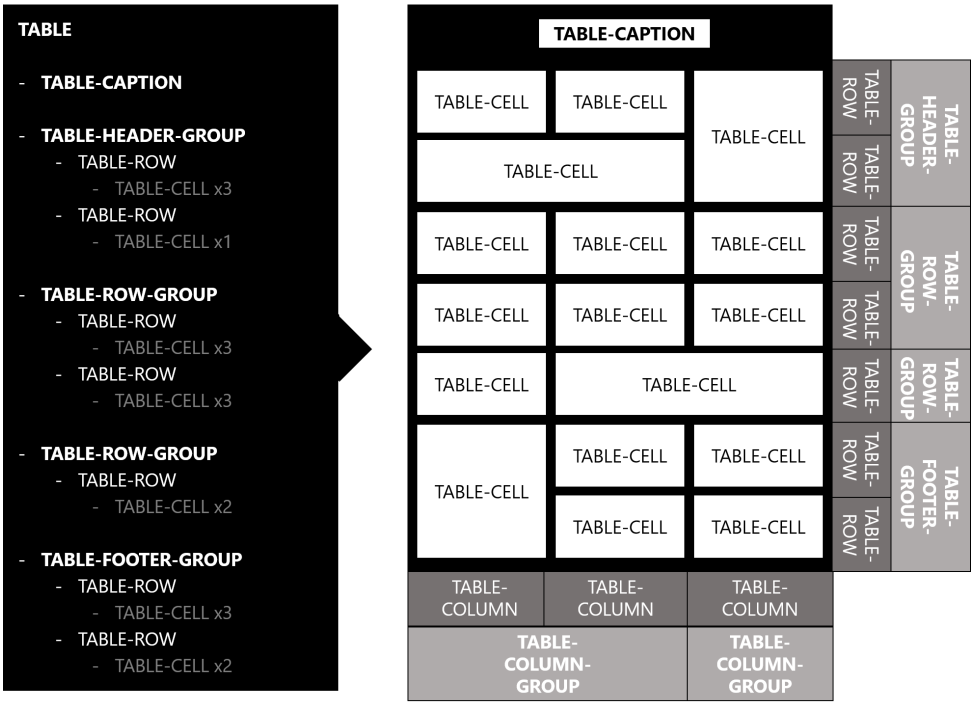


Figure 1: The instance of table Model

The following 'display' values assign table formatting rules to an arbitrary element.

table (equivalent to HTML: <table>)
Specifies that an element defines a table that is block-level when placed in flow layout.

inline-table (equivalent to HTML: <table>)
Specifies that an element defines a table that is inline-level when placed in flow layout.

table-row (equivalent to HTML: <tr>)

Specifies that an element is a row of cells. table-row-group (equivalent to HTML: <tbody>) Specifies that an element groups some amount of rows.

table-header-group (equivalent to HTML: <thead>)

Like table-row-group but, for layout purposes, the first such row group is always displayed before all other rows and row groups.

table-footer-group (equivalent to HTML: <tfoot>)

Like table-row-group but, for layout purposes, the fist such row group is always displayed after all other rows and row groups.

table-column (equivalent to HTML: <col>)

Specifies that an element describes a column of cells.

table-column-group (equivalent to HTML: <colgroup>)

Specifies that an element groups one or more columns.

table-cell (equivalent to HTML: <td> or <th>)

Specifies that an element represents a table cell.

table-caption (equivalent to HTML: <caption>)

Specifies a caption for the table. Table captions are positioned between the table margins and its borders.

### 1.3.2 Implementation

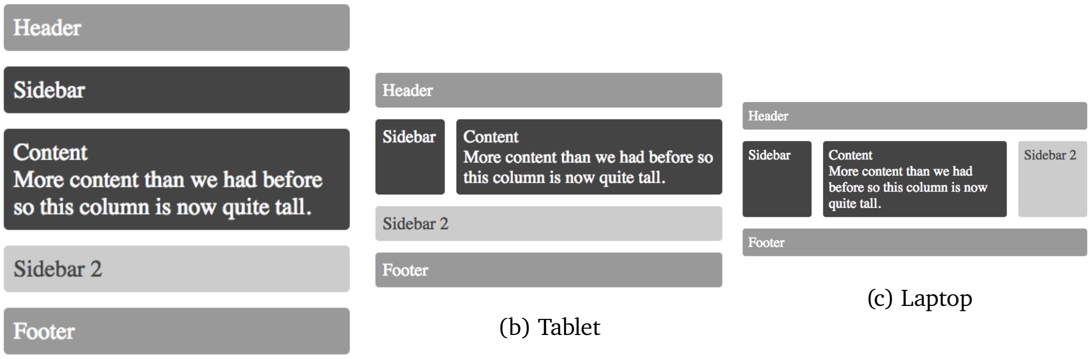
```
<div class="row">
  <div class="cell">George</div>
  <div class="cell">4287</div>
  <div class="cell">1998</div>
</div>
```

The equivalent styles is:

```
    .row { display: table-row }
    .cell { display: table-cell }
```

Question 1

Use media queries to make your layout similar to the images shown in Fig 2:



More content than we had before so this column is now quite tall.

(a) Phone

More content than we had before so this column is now quite tall.

(b) Tablet

More content than we had before so this column is now quite tall.

(c) Laptop

Figure 2: Media Rules Q1