CS/COE 1520

www.cs.pitt.edu/~nlf4/cs1520/

Responsive Web Design
A tale of two

In this mini-series I will be looking at various important elements, as well as the environment.

On this page we’re going to start by covering the environment. If you’re totally new to mobile, you might want to read this first.

The problem of...
Viewing a webpage on a smartphone
Viewports

Visual viewport

Layout viewport
The idea behind *responsive design*

- "If you put water into a cup, it becomes the cup. You put water into a bottle and it becomes the bottle. You put it in a teapot, it becomes the teapot."
  
  - Bruce Lee
What happens when the user zooms in on their phone?
  ○ Need to display same portion of the page using more pixels
    ■ Should this scale up the size of the layout viewport?

Pixel density of displays has begun to increase dramatically
  ○ How can we render the same page on both standard and HiDPI displays?

In both cases, we'll consider an abstract "pixel" size when drawing the layout viewport, and map that to hardware pixels in the visual viewport
  ○ Layout viewport size is measured in "CSS pixels"
Mobile browsers attempt to show the entire layout viewport in the browser window.

- The first tiny Wikipedia page a few slides back

How do we size the layout viewport appropriately?

- We want to ensure that our webpage isn't rendered at the default layout viewport size and then "zoomed out" to fit.
Meta viewport tags

- HTML `<meta>` tags are used to specify metadata that cannot be encoded in other tags.
- With the development of their Retina displays, Apple started using the `<meta name="viewport" ...>` tag to instruct the browser on sizing the layout viewport to properly display webpages formatted for mobile.
- E.g.:

  `<meta name="viewport" content="width=device-width, initial-scale=1">`
Great! But how to we build one page for all?

● CSS media queries
  ○ Allow the developer to tailor the site to presentation on a variety of output media without changing the content
  ○ Relevant for our case:
    ■ max-width: 600px
    ■ min-width: 500px
    ■ orientation: landscape
      ● orientation: portrait
  ○ Can be included in <link> tags to stylesheets, @import statements, or directly in css via @media tags
MDN's "pseudo-BNF" for media queries

- media_query_list: <media_query> [, <media_query> ]*
  - media_query: [[only | not]? <media_type> [ and <expression> ]]*
    | <expression> [ and <expression> ]*
  - expression: ( <media_feature> [: <value>]? )
  - media_type: all | aural | braille | handheld | print |
    | projection | screen | tty | tv | embossed | speech
  - media_feature: width | min-width | max-width
    | height | min-height | max-height
    | aspect-ratio | min-aspect-ratio | max-aspect-ratio
    | color | min-color | max-color
    | color-index | min-color-index | max-color-index
    | monochrome | min-monochrome | max-monochrome
    | resolution | min-resolution | max-resolution
    | scan | grid
Aside: BNF, or Backus–Naur form

- A way to describe a grammar
- Symbols are enclosed in `< >`
- Symbols are defined using `::=`
- Options for defining a symbol are enumerated with `|`
- E.g.:
  - `<integer> ::= <digit> | <digit> <integer>`
    `<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"`
  - `<loop-statement> ::= <while-loop> | <for-loop>`
    `<while-loop> ::= "while (" <condition> ")" <statement>"
US Postal address BNF Example

<postal-address> ::= <name-part> <street-address> <zip-part>

<name-part> ::= <personal-part> <last-name> <opt-suffix-part> <EOL>
    |  <personal-part> <name-part>

<personal-part> ::= <initial> "."  |  <first-name>

<street-address> ::= <house-num> <street-name> <opt-apt-num> <EOL>

<zip-part> ::= <town-name> "," <state-code> <ZIP-code> <EOL>

<opt-suffix-part> ::= "Sr." | "Jr." | <roman-numeral> | ""

<opt-apt-num> ::= <apt-num> | ""
BNF in BNF

```plaintext
<syntax> ::= <rule> | <rule> <syntax>
<rule> ::= <opt-whitespace> "<" <rule-name> ">" <opt-whitespace> ":=" <opt-whitespace> <expression> <line-end>
<opt-whitespace> ::= " " <opt-whitespace> | ""
<expression> ::= <list> | <list> <opt-whitespace> "|" <opt-whitespace> <expression>
<line-end> ::= <opt-whitespace> <EOL> | <line-end> <line-end>
<list> ::= <term> | <term> <opt-whitespace> <list>
<term> ::= <literal> | "<" <rule-name> "">
<literal> ::= '"' <text1> '"' | "'" <text2> "'
<text1> ::= '"' <text1> '"' | "'" <text2> "'
<text2> ::= '"' | <character1> <text1>
<character1> ::= '"' | <character2> <text2>
<character> ::= <letter> | <digit> | <symbol>
<letter> ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
<symbol> ::= "|" | """ | "!" | ";" | ";" | "(" | ")" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";" | ";"
<character1> ::= <character> | """
<character2> ::= <character> | """
<rule-name> ::= <letter> | <rule-name> <rule-char>
<rule-char> ::= <letter> | <digit> | "-"
```

Common BNF extensions

- Optional items are enclosed in [ ]
- Items repeated 0 or more time are suffixed with *
- Items repeated 1 or more time are suffixed with +
Back to responsive design

- A couple of guidelines:
  - Use relative sizes
    - E.g., define the width of divs as a percentage of the page instead of a fixed pixel size
  - Start with the smallest needed size and define "breakpoints" as necessary