CSS techniques

Web development I: Front-end engineering

CSS reset

Goal: Set a starting point that is as neutral as possible.

Method: Create style rules that overrides all user agent styles.

```
/* http://meyerweb.com/eric/tools/css/reset/
    v2.0 | 20110126 License: none (public domain)*/
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center, dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
```

Image replacement techniques

```css
h1#logo {

  background: url(f.png) no-repeat;

  overflow: hidden;

  text-indent: -9999px;

}
```

What users see:

What's actually happening:

text-indent: 100%;
overflow: hidden;
white-space: no wrap;

Edges of h1 visiible
elemnt box (outlines
added for clarity)

The h1 text content
is pushed outside the
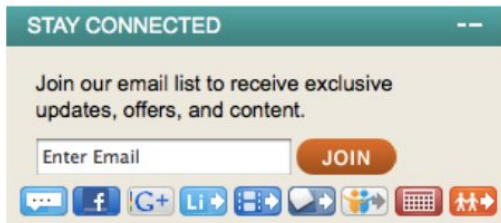visible element box
and can not be seen.

CSS sprites
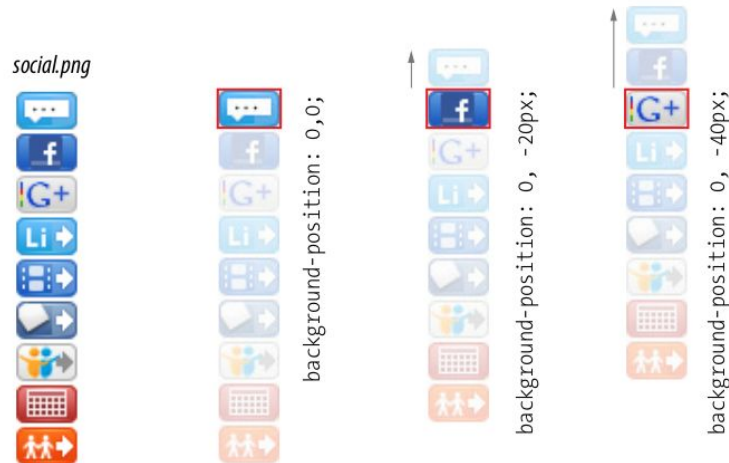
```
<ul>

  <li><a class="hide tw"
href="…">Twitter</a></li>

  <li><a class="hide fb"
href="…">Facebook</a></li>

  …

</ul>
```


The icons in this panel are contained on one spite (*social.png*) and positioned in each link using `background-position`.

social.png

background-position: 0,0;

background-position: 0, -20px;

background-position: 0, -40px;

CSS variables and functions

```css
:root {
  --main-color: red;
}


p {
  color: var(--main-color);
}
```

```css
div {
  width: calc(10px + 100px);
}
```

Preprocessors (Sass, SCSS, Less)

```scss
$my-red: #900;

a {

  border-color: $my-red;

}
```

```scss
@mixin special {

    color: #fff;

    background-color: #befc6d;

    border: 1px dotted #59950c;

}

a.nav {

    @include special;

}
```

Methodologies

OOCSS   Object-Oriented CSS

SMACSS   Scalable and Modular Architecture for CSS

SUIT CSS   Style tools for UI components

BEM   Blocks, Elements and Modifiers
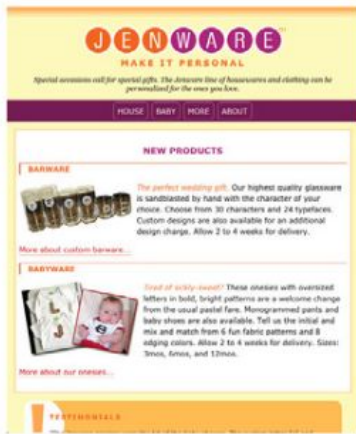
Responsive Web Design

Goal: adapt a page's layout based on screen size.

Method: Fluid/Hybrid layout + Flexible images + CSS media queries.

320px (iphone)   768px (iPad portrait)   1024px (iPad landscape)   > 1024px (Chrome/desktop)

Responsive Web Design

Setting the viewport is key:

```html
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Making images flexible:

```css
img { max-width: 100%; }
```

Media types
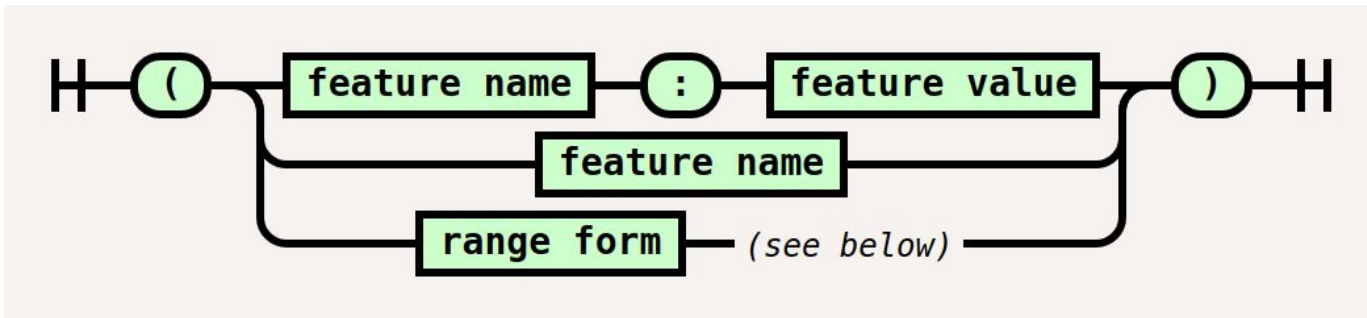
all: Matches all devices.

print: Matches printers.

screen: Matches non-printers.

Examples:

```
<link rel="stylesheet" type="text/css" media="print" href="print.css" />
@import "print-styles.css" print;
```

Media queries


```
<link rel="stylesheet" type="text/css" media="screen and (color)"
href="file.css" />

@import "file.css" screen and (color);

@media screen and (color) { … }
```

Media features

`width` and `height`   Device width and height

`aspect-ratio`   Aspect ratio: width / height

`orientation`   Device orientation: portrait, landscape

`display-mode`   Only in manifest of PWAs

`resolution`   Pixel density of the device

`grid`   Is it a grid (`1`) or bitmap (`0`) screen?

`color`   Number of bits per color component (RGB)

`monochrome`   Monochrome display

Media features (level 4)

update   Frequency of rendered content

pointer   Match *primary* pointing device (e.g. mouse)

hover   *Primary* input device can hover

any-pointer   Match any pointing device

any-hover   Match any hovering capabilities

Media features (level 5)

`prefers-color-scheme`   User wants a light or dark color scheme

`prefers-reduced-motion`   User wants less animations

`prefers-contrast`   User wants higher (or lower) contrast

`scripting`   Whether JavaScript is enabled

Media features: logical operators

`and`   Chain features

`not`   Negate features

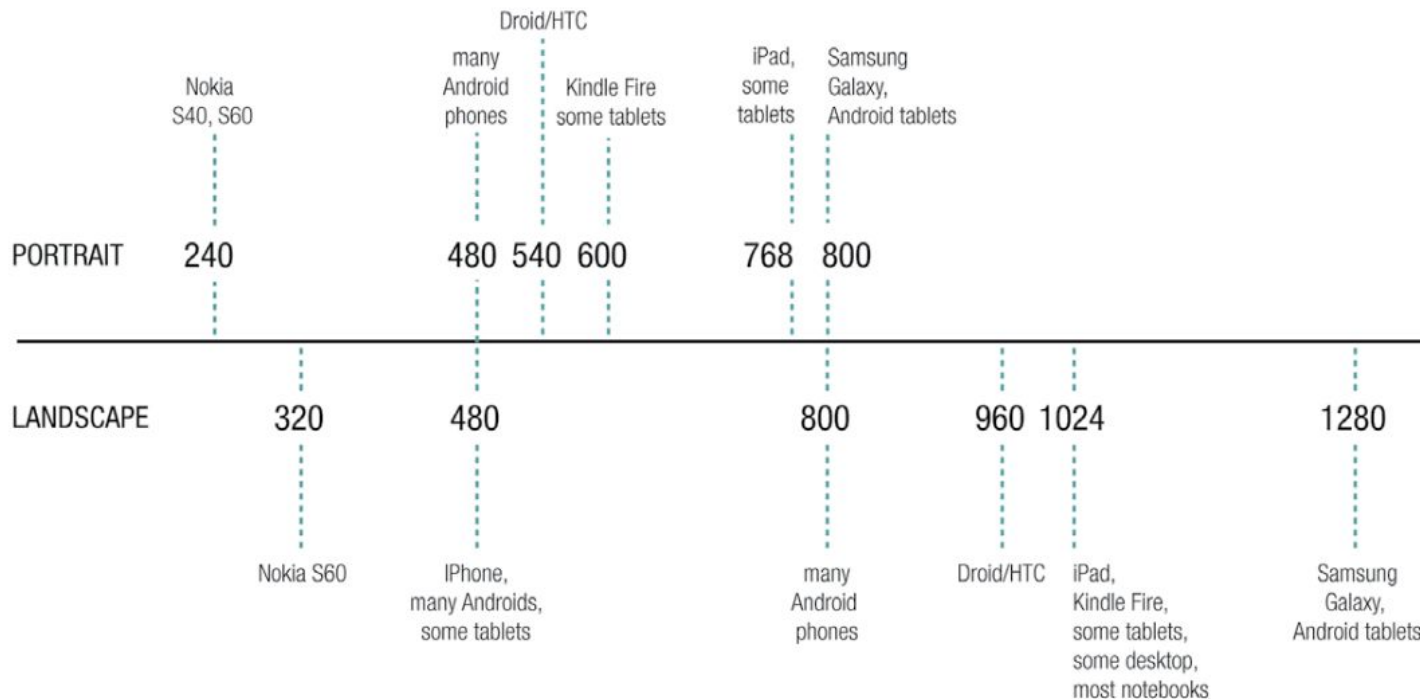`only`   Apply a style only if an entire query
                                        matches

Targeting devices

```css
@media screen and (max-width: 480px) {

    /* Styles for mobile browsers go here. */

}



@media screen and (min-resolution: 300dpi) {

    /* Styles referencing high-resolution images. */

}
```

Choosing breakpoints

Feature queries

```css
@supports (display: grid) {
  div {
    display: grid;
  }
}
```

```css
@supports not (display: grid) {
  div {
    float: right;
  }
}
```