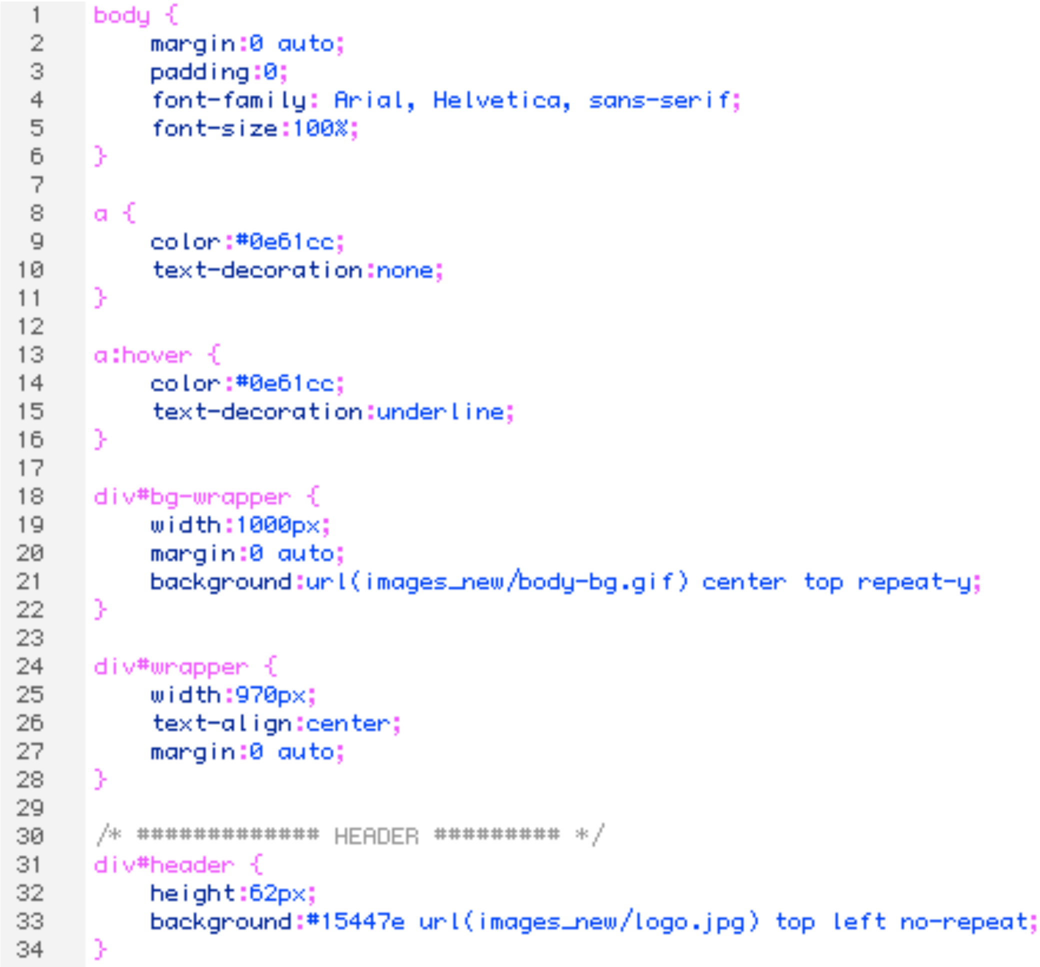
Web Programming

YJ – 2016

CSS

CSS exists to style your HTML - to present your content. The language of Cascading Style Sheets is somewhat different to HTML but it remains simple and straightforward.



```
body {
    margin:0 auto;
    padding:0;
    font-family: Arial, Helvetica, sans-serif;
    font-size:100%;
}

a {
    color:#0e61cc;
    text-decoration:none;
}

a:hover {
    color:#0e61cc;
    text-decoration:underline;
}

div#bg-wrapper {
    width:1000px;
    margin:0 auto;
    background:url(images_new/body-bg.gif) center top repeat-y;
}

div#wrapper {
    width:970px;
    text-align:center;
    margin:0 auto;
}

/* ############# HEADER ######### */
div#header {
    height:62px;
    background:#15447e url(images_new/logo.jpg) top left no-repeat;
}
```

CSS type

There are three ways to apply CSS to HTML: In-line, internal, and external

❖ In-line

```
<p style="color: red">text</p>
```

❖ Internal

```
<html>
<head>
<title>CSS Example</title>
<style>
    p {
        color: red;
    }
    a {
        color: blue;
    }
</style>
```

❖ External

```
<head>
    <title>CSS Example</title>
    <link rel="stylesheet"
href="style.css">
</head>
```

Selectors & Properties & Values

```
body {
    font-size: 14px;
    color: navy;
}
```

General units

- px (such as font-size: 12px) is the unit for pixels.
- em (such as font-size: 2em) is the unit for the calculated size of a font. So "2em", for example, is two times the current font size.
- pt (such as font-size: 12pt) is the unit for points, for measurements typically in printed media.
- % (such as width: 80%) is the unit for… wait for it… percentages.

Colors

```
h1 {
     color: yellow;
     background-color: blue;
}
```

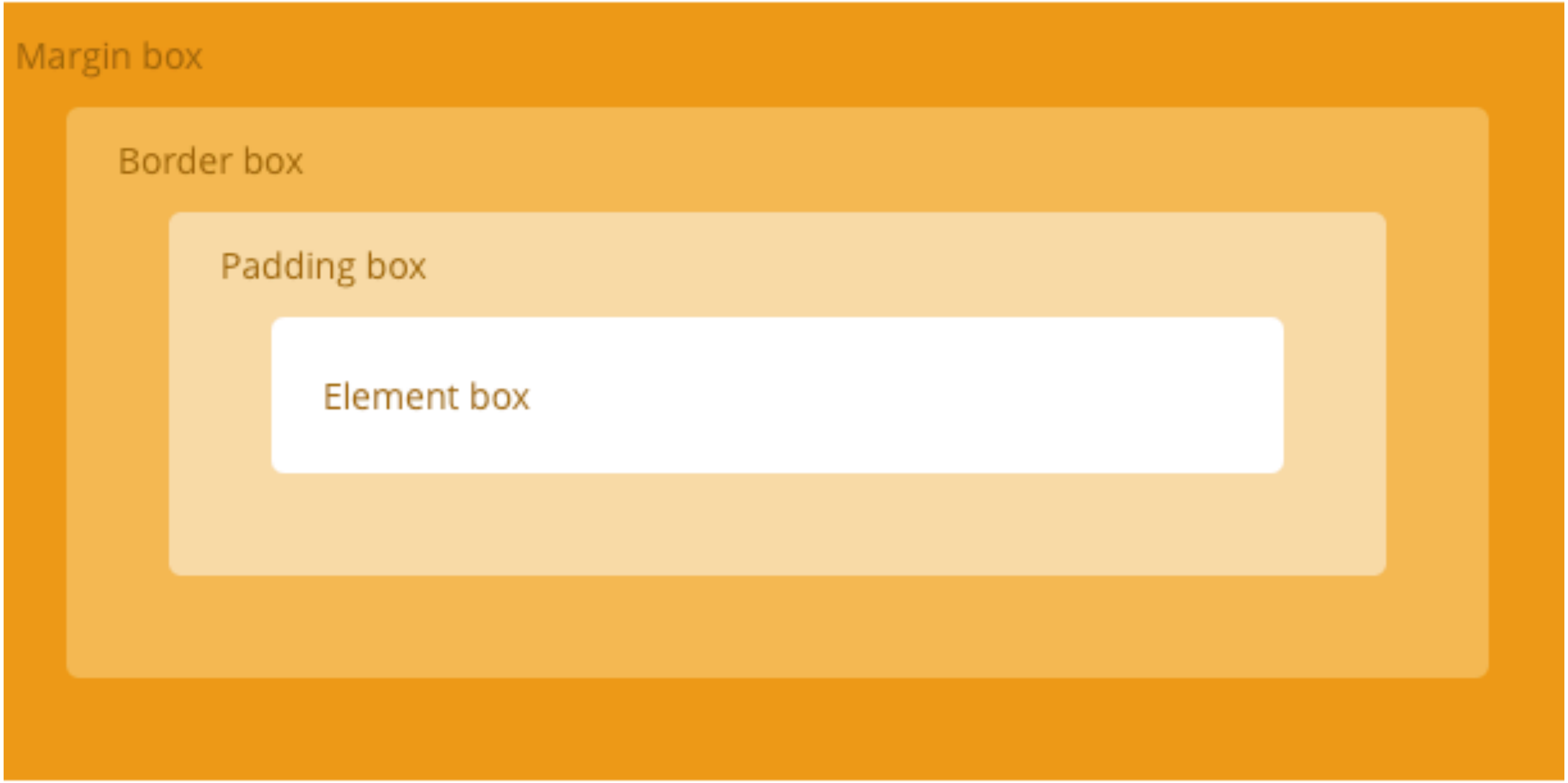
- red
- rgb(255,0,0)
- rgb(100%,0%,0%)
- #ff0000
- #f00

Text

- Font-family
  - E.g. “Arial”
- Font-size
- Font-weight
  - Normal, bold, bolder, lighter, 100, 200, 300, 400 (same as normal), 500, 600, 700 (same as bold), 800 or 900
- Font-style
  - Italic
- Text-decoration
  - Underline, overline, line-through
- Text-transform
  - capitalize, uppercase, lowercase, none
- Text spacing

```
body {
    font-family: arial, helvetica, sans-serif;
    font-size: 14px;
}

h1 {
    font-size: 2em;
}

h2 {
    font-size: 1.5em;
}

a {
    text-decoration: none;
}

strong {
    font-style: italic;
    text-transform: uppercase;
}
p {
    letter-spacing: 0.5em;
    word-spacing: 2em;
    line-height: 1.5;
    text-align: center;
}
```

Margin & Padding



Margin & Padding

- margin-top, margin-right, margin-bottom, margin-left, padding-top, padding-right, padding-bottom and padding-left

```
h2 {
    font-size: 1.5em;
    background-color: #ccc;
    margin: 20px;
    padding: 40px auto;
}
```

Borders

```
h2 {
    border-style: dashed;
    border-width: 3px;
    border-left-width: 10px;
    border-right-width: 10px;
    border-color: red;
}
```

Class & ID

- In the CSS, a class selector is a name preceded by a full stop (“.”) and an ID selector is a name preceded by a hash character (“#”).

```
#top {
    background-color: #ccc;
    padding: 20px
}

.intro {
    color: red;
    font-weight: bold;
}
////////////////////////////////////////
<div id="top">

<h1>Chocolate curry</h1>

<p class="intro">This is my recipe for making curry purely with chocolate</p>

<p class="intro">Mmm mm mmmmm</p>

</div>
```

Grouping

```
h2 {
    color: red;
}

.thisOtherClass {
    color: red;
}

.yetAnotherClass {
    color: red;
}
```

Equals

```
h2, .thisOtherClass,
.yetAnotherClass {
    color: red;
}
```

Nesting

```
#top {
    background-color: #ccc;
    padding: 1em
}

#top h1 {
    color: #ff0;
}

#top p {
    color: red;
    font-weight: bold;
}

/////////////////////////////
<div id="top">
    <h1>Chocolate curry</h1>
    <p>This is my recipe for making curry
purely with chocolate</p>
    <p>Mmm mm mmmmm</p>
</div>
```

Pseudo Classes

- Pseudo classes are bolted on to selectors to specify a state or relation to the selector. They take the form of selector:pseudo_class { property: value; }, simply with a colon in between the selector and the pseudo class.

```css
a:link {
    color: blue;
}
a:visited {
    color: purple;
}
a:active {
    color: red;
}
a:hover {
    text-decoration: none;
    color: blue;
    background-color: yellow;
}

input:focus, textarea:focus {
    background: #eee;
}
p:first-child {
    font-weight: bold;
    font-size: 40px;
}
```

Shorthand Properties

```css
p {
    font: 14px/1.5 "Times New Roman", times, serif;
    padding: 30px 10px;
    border: 1px black solid;
    border-width: 1px 5px 5px 1px;
    border-color: red green blue yellow;
    margin: 10px 50px;
}
```

Background

- background-color, which we have come across before.
- background-image, which is the location of the image itself.
- background-repeat, which is how the image repeats itself. Its value can be:
    - repeat, the equivalent of a “tile” effect across the whole background,
    - repeat-y, repeating on the y-axis, above and below,
    - repeat-x (repeating on the x-axis, side-by-side), or
    - no-repeat (which shows just one instance of the image).
- background-position, which can be top, center, bottom, left, right, a length, or a percentage, or any sensible combination, such as top right.

```
body {
  background:
  url(sweettexture.jpg)   /* image */
  top center / 200px 200px /* position / size */
  no-repeat               /* repeat */
  fixed                  /* attachment */
  padding-box             /* origin */
  content-box             /* clip */
  red;                   /* color */
}
```

Background

Multiple backgrounds

```
background: url(bg.png), url(bullet.png) 0 50% no-repeat,
url(arrow.png) right no-repeat;
```

Gradients

```
background: linear-gradient(orange, red);

background: linear-gradient(to bottom right, orange, red);

background: linear-gradient(20deg, orange, red);

background: linear-
gradient(hsl(0,100%,50%),hsl(60,100%,50%),hsl(120,100%,50%
),hsl(180,100%,50%),hsl(240,100%,50%),hsl(300,100%,50%));
```

Background

Radial gradients

```
background: radial-gradient(yellow, green);
background: radial-gradient(circle closest-side, yellow, green);
background: radial-gradient(at top left, yellow, green);
```

Repeating gradients

```
background: repeating-linear-gradient(white, black 10px, white 20px);
```

Pseudo Elements

```
p {
    font-size: 12px;
}

p:first-letter {
    font-size: 24px;
    float: left;
}

p:first-line {
    font-weight: bold;
}

li:before {
    content: "POW! ";
}

P:after {
    content: url(images/jam.jpg);
}
```

Positioning

- static is the default value and renders a box in the normal order of things, as they appear in the HTML.
- relative is much like static but the box can be offset from its original position with the properties top, right, bottom and left.
- absolute pulls a box out of the normal flow of the HTML and delivers it to a world all of its own. In this crazy little world, the absolute box can be placed anywhere on the page using top, right, bottom and left.
- fixed behaves like absolute, but it will absolutely position a box in reference to the browser window as opposed to the web page, so fixed boxes should stay exactly where they are on the screen even when the page is scrolled.

Floating

- Floating a box will shift it to the right or left of a line, with surrounding content flowing around it.
- Float:Left
- Float:right
- Float:center

- clear: left will clear left floated boxes
- clear: right will clear right floated boxes
- clear: both will clear both left and right floated boxes.

Border Radius

- border-radius: 10px;

- Multiple Value
- border-radius: 3px 6px 9px 12px;

- Ellispses
- border-radius: 50px/100px;
- border-bottom-left-radius: 50px;
- border-bottom-right-radius: 50px;

Shadows

- `box-shadow: 5px 5px 3px 1px #999`
- `box-shadow: inset 0 0 7px 5px #ddd;`
- `text-shadow: -2px 2px 2px #999;`
  - The first value is the horizontal offset
  - The second value is the vertical offset
  - The third value is the blur radius (optional)
  - The fourth value is the color (optional, although omitting this will make the shadow the same color as the text itself)

Selectors

- Universal selectors

```
* {
    margin: 0;
    padding: 0;
}

#contact * {
    display: block;
}
```

- Child selectors

```
#genus_examples > li { border: 1px solid red }
```

- Adjacent selectors

```
h1 + p { font-weight: bold }
```

Attribute Selectors

- `input[type=text] { width: 200px; }`
- `input[type=text][disabled] { border: 1px solid #ccc }`

- `[attribute^=something]` will match a the value of an attribute that begins with something.
- `[attribute$=something]` will match a the value of an attribute that ends with something.
- `[attribute*=something]` will match a the value of an attribute that contains something, be it in the beginning, middle, or end.

Transition

```
a:link {
    transition: all .5s linear 0;
    color: #ff0000;
}
a:hover {
    color: #ffff00;
}

transition: .5s;
transition-property: color, font-size;

transition: color .5s, font-size 2s;
```

Rules

Importing

```
@import url(morestyles.css);
```

Media

```
@media print {
    body {
        font-size: 10pt;
        font-family: times, serif;
    }
}
@media screen, projection {
    /* ... */
}
```

Rules

Screen size

```
@media screen and (max-width: 1000px) {
    #content { width: 100% }
}

@media screen and (max-width: 800px) {
    #nav { float: none }
}

@media screen and (max-width: 600px) {
    #content aside {
        float: none;
        display: block;
    }
}
```

Rules

Orientation

```
@media screen and (orientation: landscape) {
    #nav { float: left }
}

@media screen and (orientation: portrait) {
    #nav { float: none }
}
```

Device

```
@media screen and (min-device-height: 768px) and (max-device-width: 1024px) {
    /* You can apply numerous conditions separated by "and" */
}
@media screen and (resolution: 326dpi) { /* */ }

@media screen and (min-resolution: 96dpi) { /* */ }

@media screen and (device-aspect-ratio: 16/9) { /* */ }
```