Cascading Style Sheets
CSS

Part #1

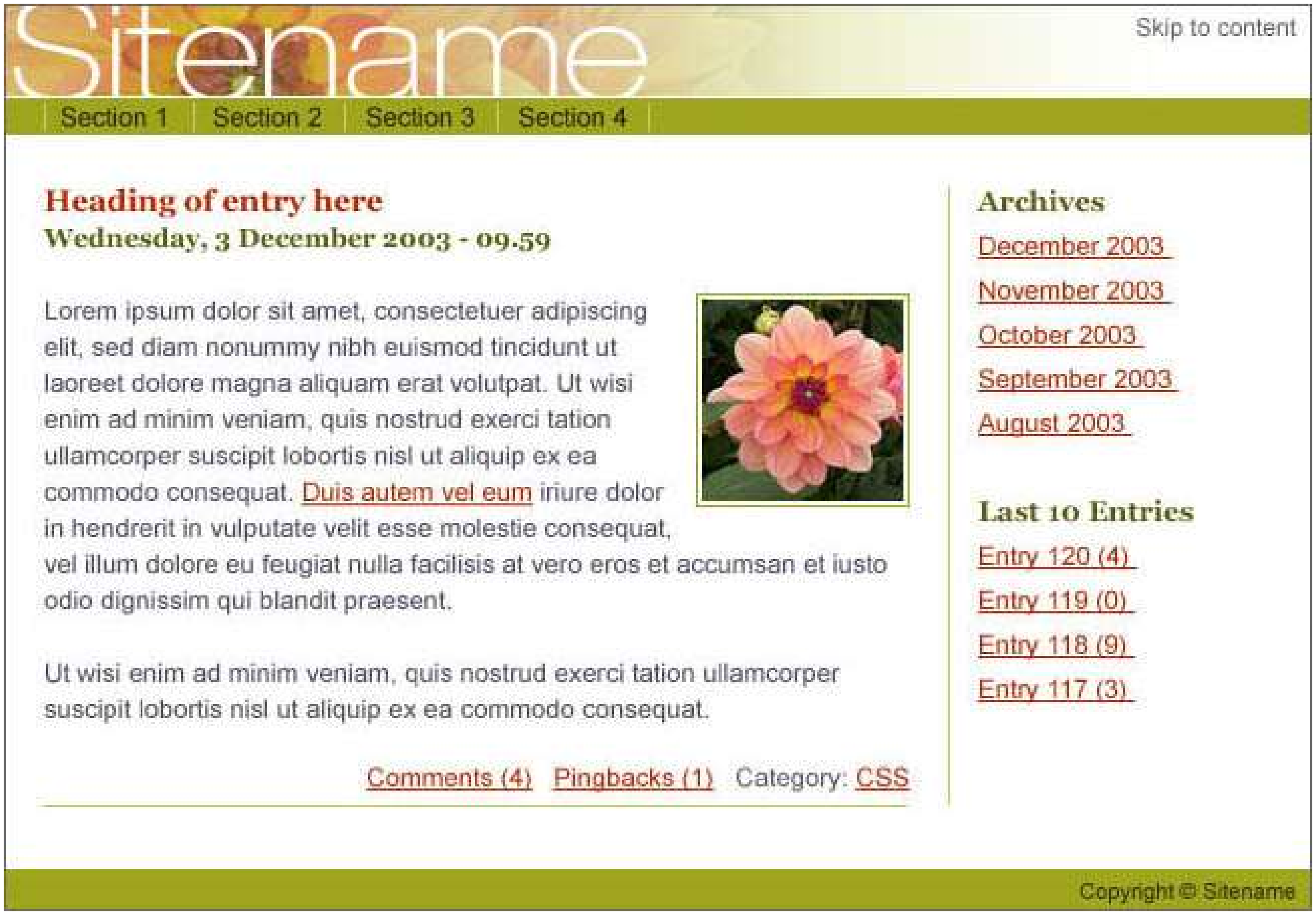
Topics

- History
- Basic Syntax
- Selectors
- Fonts
- Lists
- Colors
- Alignments
- Background images
- Borders

What is Cascading Style Sheets?

- CSS is a language that describes the style of an HTML document
- CSS describes how HTML elements should be displayed
- Like HTML, CSS is NOT a programming language

Use CSS to define the page layout



Sitename

Skip to content

Section 1 Section 2 Section 3 Section 4

Heading of entry here

Wednesday, 3 December 2003 - 09.59

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Comments (4) Pingbacks (1) Category: CSS

Archives

December 2003

November 2003

October 2003

September 2003

August 2003

Last 10 Entries

Entry 120 (4)

Entry 119 (0)

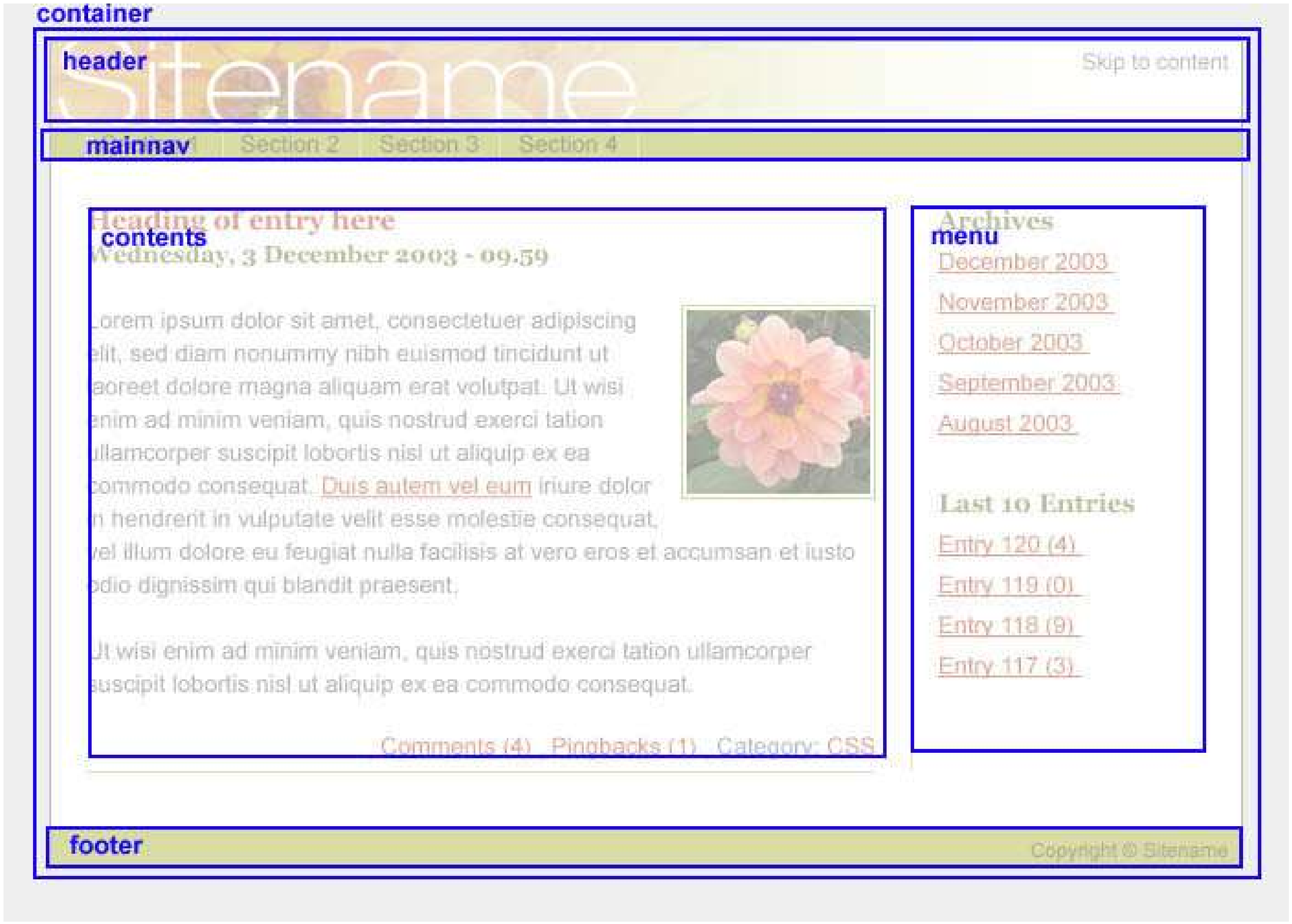
Entry 118 (9)

Entry 117 (3)

Copyright © Sitename

from maxdesign.com.au

Layout



Heading of entry here

Wednesday, 3 December 2003 - 09.59

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Best Practices

- Responsiveness
  - Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.
- Mobile-first
  - First design the style for mobile devices
  - Do you really need to develop a mobile app?
- Browser compatibility
  - Chrome, Firefox, IE, Edge, Safari, Opera
- Animation with CSS is preferred over animation with JavaScript

History

- CSS1 specification was developed in 1996
- CSS2 was released in 1998
- CSS3 was released in 1999
- Usage: consistent presentation style
  - CSSs provide the means to control and change presentation of HTML documents
  - CSS is not technically HTML, but can be embedded in HTML documents
  - Style sheets allow you to impose a standard style on:
    - a special element, or
    - a whole document, or
    - a whole collection of documents

More resources

- Check the w3school tutorial on responsive CSS at https://www.w3schools.com/css/css_rwd_intro.asp
- Check the bootstrap 5 tutorial at w3school https://www.w3schools.com/bootstrap5/

CSS References

- CSS reference and tutorials at w3school
  - https://www.w3schools.com/css/default.asp
- CSS layouts http://www.maxdesign.com.au/articles/css-layouts/

(Almost) all CSS references

- https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

Extreme CSS

- CSS zen garden: http://www.csszengarden.com/

General Syntax

```
Selector {
        property1: value1;
        property2: value2;
        …
        property3: value3;
}
```

Note: this is for non-inline CSS

Four types of selectors

- HTML elements
- Classes (most used)
- IDs
- * (universal)

Levels of Style Sheets

- Inline
  - specified for a specific occurrence of a tag and apply only to that tag
  - defeats the purpose of style sheets - uniform style
  - W3C deprecated inline style in XHTML1.1 (2001)
- Document-level style sheets
  - defined in the head section
  - apply to the whole document in which they appear
- External style sheets
  - Defined in a separate file
  - can be applied to any number of documents
  - file included in the head section
- How browsers use CSS?
  - Cascading order
  - Implementation

Inline

- Avoid this method!
- Within an HTML element, define properties using the style attribute:

```
<p style="color:red; margin-left:20px">This is a paragraph.</p>
```

Needs copy/paste to use for another paragraphs

```
<p style="color:red; margin-left:20px">This is another paragraph.</p>
```

Document Level

- Define the style in the head element of the HTML document using the <style> tag

Selector

```
<head>
  <style>
    h1 { color: red; }
    p  { margin-left: 20px;}
    body { background-color: grey; }
  </style>
</head>
```

Needs copy/paste to use in other HTML documents

External CSS

- Define styles in a separate document and link it in the head element

```
<head>
  <link rel="stylesheet" href="mystyle.css" />
</head>
```

Relationship

- Inside mystyle.css

```
h1 { color: red;}
p { margin-left: 20px;}
body { background-color: grey;}
```

Simple Selector

- tag names {property_1: value_1; property_2:value_2; …}

```
h1 {color: white;}
```

h1	selector
`{color:white;}`	declaration
color	property
white	value

- Selectors can be grouped as in

```
h1, h2, h3 {color: green;}
```

- Declarations can be grouped as in

```
h1 {color: white; background: black;}
```

Contextual Selectors

- Applied to child within parent
- Applied to em inside h1

```
h1 em { color:blue }
```

- NOT this

```
h1, em { color:blue }
```

Class selectors

```
p.normal {property-value list}
p.warning {property-value list}
```

```
<p class = "normal">
 A paragraph in 'normal' presentation style
</p>
```

```
<p class = "warning">
 A paragraph in 'warning' presentation style
</p>
```

- http://www.w3schools.com/Css/tryit.asp?filename=trycss_syntax_element_class

Generic selectors

```
.sale {property-value list}
```

```
<h1 class = "sale"> Weekend Sale </h1>
<p class = "sale"> … </p>
```

- http://www.w3schools.com/Css/tryit.asp?filename=trycss_syntax_class

Universal selectors

```
* {property-value list}
```

`id` selectors

```
#section14 {font-size: 20}          ✓
h2#section14 {font-size: 20}        ✗
p#section14 {font-size: 20}
```

```
<h2 id = "section14"> Header 2 text</h2>
```

- http://www.w3schools.com/Css/tryit.asp?filename=trycss_syntax_id

Pseudo Classes

- A pseudo-class is used to define a special state of an element.
- For example, it can be used to:
  - Style an element when a user mouses over it
  - Style visited and unvisited links differently
  - Style an element when it gets focus

Pseudo Classes (2)

```
<!DOCTYPE html>
<html>
  <head> <title> Checkboxes </title>
    <style>
      input:hover {color: red; background: pink;}
      input:focus {color: green; background: blue;}
    </style>
  </head>
  <body>
    <form action = "">
      <p>
        Your name:
        <input type = "text" />
      </p>
    </form>
  </body>
 </html>
```

Pseudo class is useful for animation

- Previous example: with text input
- Another example: with hyperlink
  - `:link` a hyperlink that has not been visited
  - `:visited` a hyperlink that has been visited
  - `:active` a hyperlink that is being clicked on
  - `:hover` a hyperlink over which the mouse is

```
a:link     {color:#FF0000;}      /* unvisited link */
a:visited {color:#00FF00;}      /* visited link */
a:hover   {color:#FF00FF;}      /* mouse over link */
a:active  {color:#0000FF;}      /* link being clicked */
```

http://www.w3schools.com/css/tryit.asp?filename=trycss_link

The Cascade

- Most properties are inherited. For example, in

```
<!DOCTYPE html>
<head>
    <style>
        h1 {color: gray;}
    </style>
</head>
<body>
      <h1>It's a wonderful day for <em>science!</em>
</h1>
</body>
```

- Some properties, such as border, do not inherit.
- More *specific* selectors dominate.
  - *id* dominates *class* which dominates all others

Order

- id > class/pseudo class > simple/contextual selector> * > default style

```
<!DOCTYPE html>
<head>
     <style>
          .blue {color:blue;}
           h1 {color:green;}
           #highlight {color: red;}
     </style>
</head>
<body>
  <h1 class="blue">This is blue</h1>
  <h1 id = "highlight"> This is red </h1>
</body>
```

How about this one?

```
<!DOCTYPE html>
<head>
     <style>
          .blue {color: blue;}
           h1 {color: green;}
           #highlight {color: red;}
           * {color: yellow; font-size: 5mm}
     </style>
</head>
<body>
      <h1> this is ________</h1>
      <h2> this is ________</h2>
</body>
```

Same selector type: the last one defined later wins

```
<!DOCTYPE html>
<head>
     <style>
        .green {color: green}
        a {color: yellow;}
        h1 {color: red;}
        h1 {color: blue;}
     </style></head>
<body>
<h1>This is blue </h1>
<h1 class="green">This is green</h1>
</body>
```