729G26 Interaction Programming

Lecture 1

LINKÖPINGS UNIVERSITET

Lecture overview

- About the course
- Programming interaction
- HTML
- CSS
- Some tools of the trade

About the course

- Learning objectives
- Course contents
- Motivation behind the course
- Administrative information
- Course structure

LINKÖPINGS UNIVERSITET

Course Learning Objectives

- implement graphical interactivity using a framework
- identify and describe components used in an user interface
- implement a user interface component given a description
- discuss the programming complexity of a set of interaction techniques

Course contents

- interaction programming using a framework (e.g. HTML5+jQuery)
- interaction through text
- use of interactive elements within a framework
- event handling
- handling of input from mouse, keyboard, etc
- vector-based graphics and pixel-based graphics
- animation
- version control

Motivation behind the course

- Introduction to both theoretical knowledge and the practical skills needed to work in areas related to interaction programming.
- Many present-day applications and products are web based, making HTML5 and jQuery a good choice of development framework

Assumed prerequisites

- **Basic programming skills** (abstraction, flow control, data types) and an understanding of **object oriented programming concepts**.
- **Familiarity** with general methods and workflows related to **Interaction Design**

Course literature

- David Sawyer McFarland. 2014. JavaScript & jQuery: The Missing Manual, 3rd Edition (2nd edition is also OK).
- Google is your friend.
- Stack Exchange is your friend.
- Links to various web references can be found on the home page

Course staff

- **Examinator/course leader**

  Jody Foo

- **Teacher's assistants**

  Marcus Liw

- **Course administrator**

  Annelie Almquist

LINKÖPINGS UNIVERSITET

Course structure

- First four weeks (w 40-43), lectures + assignments
- Assignments and project are done in pairs
- Week overview

Week 40: Content and style: HTML+CSS

Week 41: Interaction using JavaScript

Week 42: Interaction using jQuery

Week 43: Using jQuery UI + jQuery plugins

Week 44-45: Project

Assignments

- 4 assignments:

HTML+CSS

JavaScript

jQuery

jQuery UI and jQuery Plugins

- 2-3 supervised computer lab sessions per assignment

- Done in pairs (signup via webreg)

- Grades: G, VG

- Examination: review/presentation sessions

Assignment Review/Presentations

- Assignment examination is done during Assignment Review/ Presentation sessions (one per assignment)
- Submit assignment before review/presentation session
- Deadlines on course web pages

- Sessions are held in the computer lab
- One computer lab per TA group
- Each student pair presents their assignment (web page + code)
- After presentation: review/discussion/questions from other students in the TA group.

Project

- **Specification**: Review/Presentation October 31
- **Implementation**: Review/Presentation November 11 (under consideration to be changed)
- More information about the project week 42

LINKÖPINGS UNIVERSITET

Workload

- Pace of course 2/3 of full time
  → ~27,7h/week
  → ~5,3h/day, 5 days/week
- Scheduled course sessions ~ 8-12h/week

LINKÖPINGS UNIVERSITET

Active knowledge acquisition

- Lectures provide an introduction and a starting point
- Interaction programming for the web is constantly changing
- **Important skills**: keeping up to date with current development, using online documentation/reference material
- Google is your friend!

Programming interaction

LINKÖPINGS UNIVERSITET

Interaction environment

- **application environment**
  - command line interface – CLI
  - full screen application
  - desktop vs web vs mobile
  - games console
- **input/output**
  - touch/mouse/pen/gestures/keyboard/buttons
  - HMD/small screen/large screen/VR/haptics
- **expectations**
  - game
  - move/entertainment
  - enterprise
  - medical

Examples of development frameworks

	Object oriented using MVC	Direct access to canvas	Scripted Components
Java	✔	✔	✘
Objective-C	✔	✔	✘
C# + .net	✔	✔	✘
Flash	✘	✔	✔
HTML5	✘	✔	✔
Axure RP	✘	✔	✔

HTML + jQuery

Development framework for this course

LINKÖPINGS UNIVERSITET

Web page components

Content	HTML
Form	CSS
Interaction	JavaScript

Web page anatomy

Lets explore!

LINKÖPINGS UNIVERSITET

HTML + CSS

Today: Quick overview. More details next lecture on Wednesday

LINKÖPINGS UNIVERSITET

HTML5

http://docs.webplatform.org/wiki/html

http://www.w3schools.com/html/default.asp

LINKÖPINGS UNIVERSITET

HTML Structure

- HTML document consists of elements
- Elements can be nested - hierarchy
- Document Object Model: DOM - tree structure

LINKÖPINGS UNIVERSITET

HTML Code

LINKÖPINGS UNIVERSITET

HTML5

```html
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Document 1</title>
    </head>
    <body>
        <p>The first and only paragraph.</p>
    </body>
</html>
```

HTML5

```html
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Document 1</title>
    </head>
    <body>
        <p>The first and only paragraph.</p>
    </body>
</html>
```

- **Elements**
  - start + end
  - attributes + values
  - content
- Elements can be nested, but may not overlap.

Tag syntax

- Docyment type

```
<!doctype html>
```

- Non-closing tag

```
<tag>
```

- Start and end tag

```
<tag>content</tag>
```

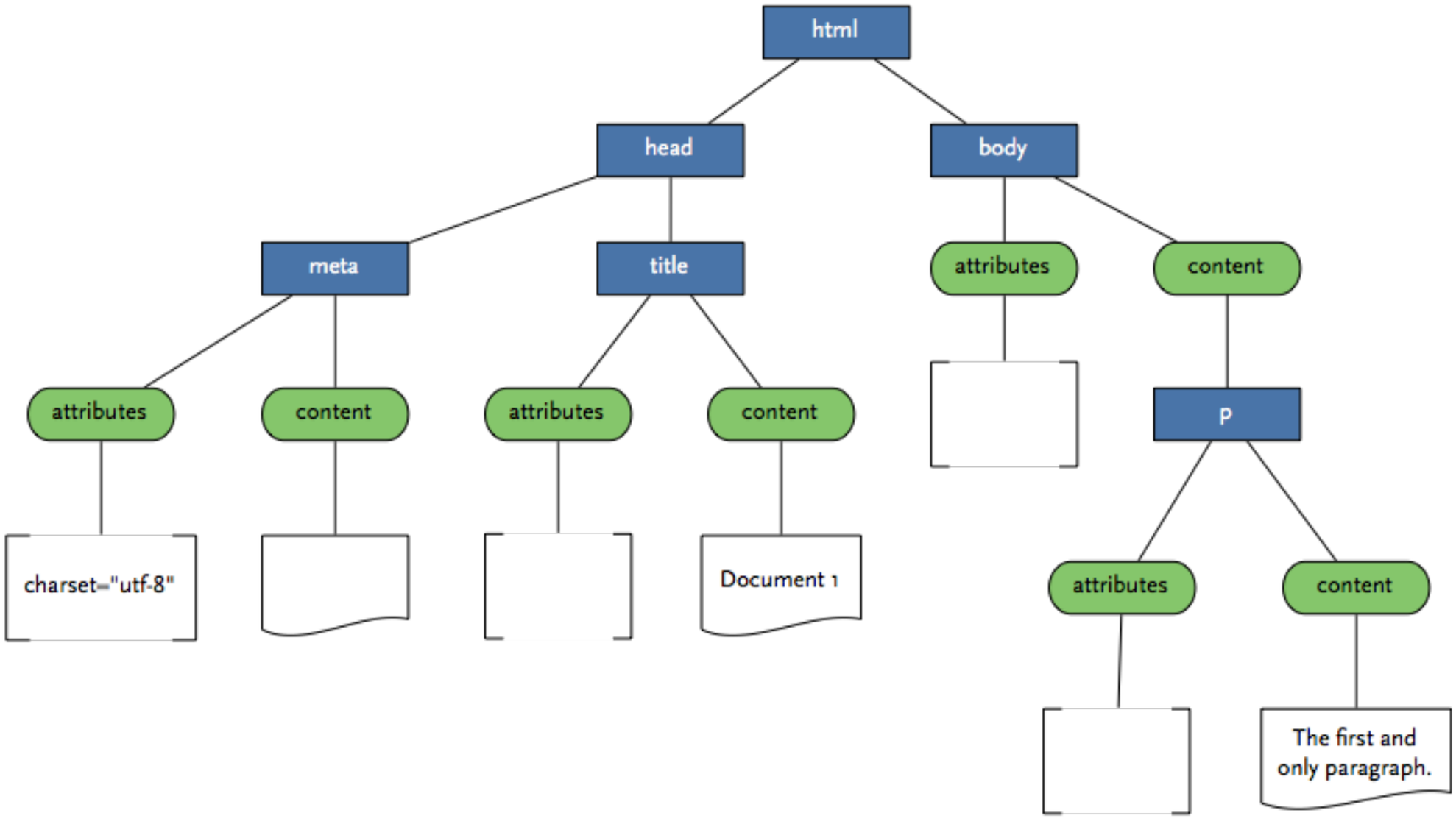
- Non-closing tag with an attribute+value

```
<tag attribute="value">
```

- Start and end tag with attribute+value

```
<tag attribute="value">content</tag>
```

DOM tree - family relationships



HTML defines structure

- semantic use

for search engines

- organizational use

prepare for css styles

prepare for interaction

Basic HTML

https://trinket.io/html/14f93a2e6e

LINKÖPINGS UNIVERSITET

Web pages at IDA

How to publish pages via your student account at IDA

LINKÖPINGS UNIVERSITET

Short guide to publishing web pages at IDA

- The directory `~/www-pub` is made available on the web via IDA's web server
- Place files in this directory with read access for everybody for them to be available at `https://www-und.ida.liu.se/~yourliuid`
- Subdirectories must be made readable and executable for files inside to be accessible
- You can upload files to your account via SFTP. Connect to `remote-und.ida.liu.se` and log in with your LiU-ID.

LINKÖPINGS UNIVERSITET

HTML + CSS

- Text files
- CSS can be written in the same file, but we will use an external file.
  - Link HTML and CSS using the `link` tag in the header

HTML5

```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <link rel="stylesheet" type="text/css" href="style.css">
        <title>Document 1</title>
    </head>
    <body>
        <p>The first and only paragraph.</p>
    </body>
</html>
```

- **Elements**
  - start + end
  - attributes + values
  - content
- Elements can be nested, but may not overlap.

Filepaths in HTML and CSS

- Unless otherwise specified, all filepaths are relative to the **current file**.
- `/` means the root of the web site, i.e. for `https://www-und.ida.liu.se/~yourliuid/index.html` the root is `https://www-und.ida.liu.se/`
- `../` means go up one directory

<!-- This is a comment -->

LINKÖPINGS UNIVERSITET

Headings

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Paragrphs

```
<p>paragraph 1</p>

<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.</p>

<p>paragraph 3</p>

<p>paragraph 4</p>
```

Links

```
<a href="url">Link text</a>
```

Images

```
<img href="url">
```

Structural elements

LINKÖPINGS UNIVERSITET

Structural elements

- A block of something `<div> </div>`
- An inline element `<span> </span>`

LINKÖPINGS UNIVERSITET

Semantic elements in HTML5

<header>

<nav>

<section>

<article>

<aside>

<figcaption>

<figure>

<footer>

http://www.w3.org/TR/html5/

CSS

http://docs.webplatform.org/wiki/css

http://www.w3schools.com/css/default.asp

LINKÖPINGS UNIVERSITET

Cascading Style Sheets

- Specifying **visual style** and **layout** for an HTML document
- HTML elements inherit CSS properties from their ancestors

  e.g. the value of the font-family property for the body element is inherited by the p element.

page.html & style.css

```html
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>A document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>The Heading</h1>
    <p>The paragraph</p>
<body>
</html>
```

```css
body {
    font-family: Times, Serif;
    font-size: 16px;
    padding: 0px 0px 0px 0px;
}

h1 {
    font-family: Helvetica, Sans-Serif;
    font-size: 32px;
}
```

Aspects of style

- colors and borders, e.g.

background color/image

borders around elements

- typography, e.g.

font, font size

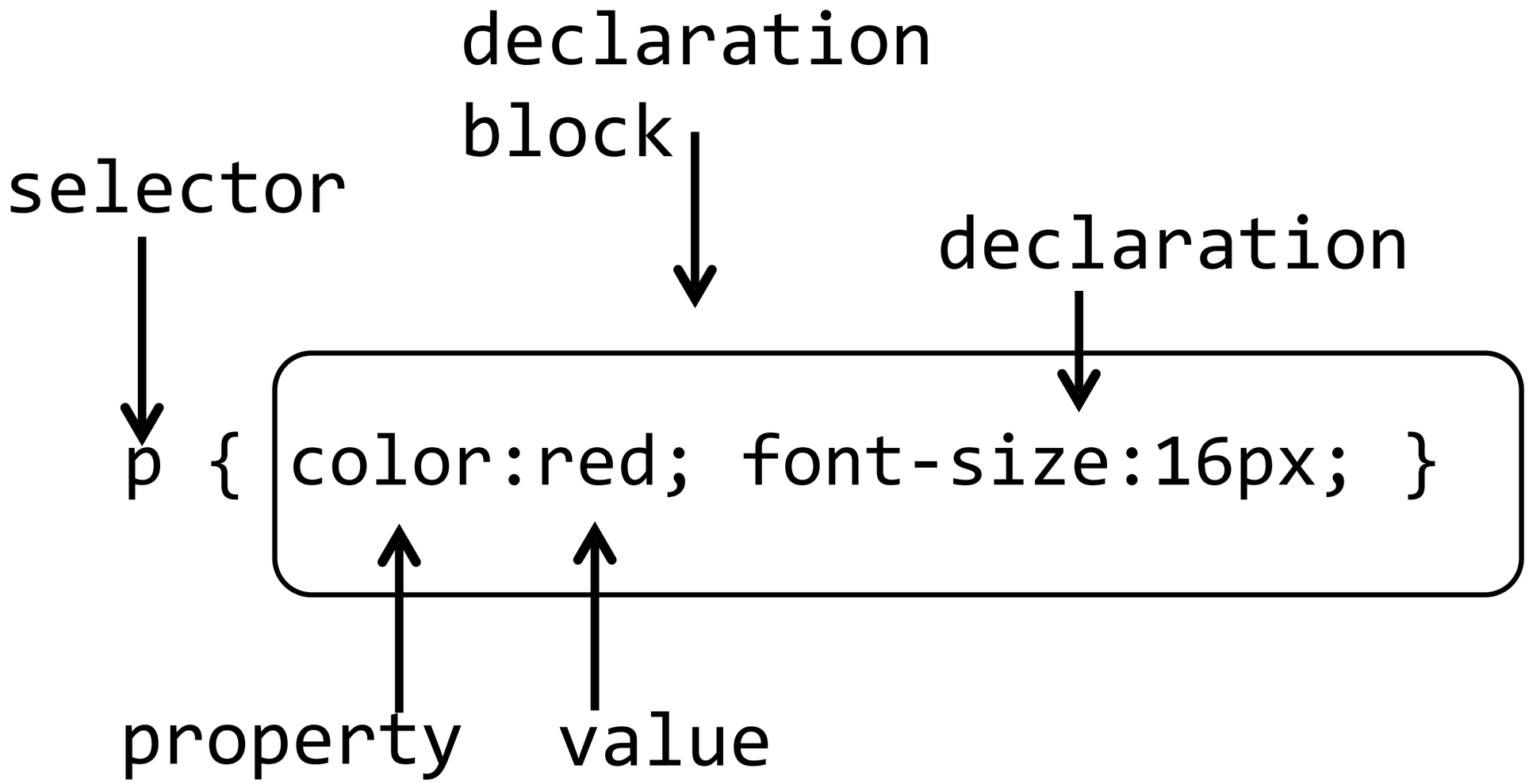
line height

- layout: size and positioning

The style sheet

- A collection of style specifications
- A HTML document can link to **one or more stylesheets**
- Style specifications are evaluated in serial from the top.
- Later specifications can overrule previous specifications.

Syntax


declaration
block
selector
declaration
p { color:red; font-size:16px; }
property
value

Syntax

```
/* This is a comment */
body {
    font-family: Times, Serif;
    font-size: 16px;
    padding: 0px 0px 0px 0px;
}

h1 {
    font-family: Helvetica, Sans-Serif;
    font-size: 32px;
}
```

Basic CSS

https://trinket.io/html/e8d82af055

LINKÖPINGS UNIVERSITET

Typography

LINKÖPINGS UNIVERSITET

font-family and font-size

```css
/* Use a CSS selector to delimit the scope of the declarations */

p {
    font-family: "Comic Sans";
    font-size: 48pt
}
```

font-weight and font-style

- Use `font-weight` to set how thick a font's lines should be, e.g.

```
font-weight: bold;
font-weight: 900;
```

- Use `font-style` to set a font's style to normal, italic or oblique

italic is a cursive style of a font

oblique is a slanting style of a font

The font stack

```
/* Use a font stack to specify font family preference in descending
order. */

p {
    font-family: Helvetica, Arial, Sans-Serif;
    font-size: 42pt;
}
```

Some further reading

- CSS Font stacks
  http://coding.smashingmagazine.com/2009/09/22/complete-guide-to-css-font-stacks/
- @font-face
  http://coding.smashingmagazine.com/2013/02/14/setting-weights-and-styles-at-font-face-declaration/

Colors and borders

LINKÖPINGS UNIVERSITET

Specifying color: RGB

- Red, Green, Blue - Additive color model
- Values from `0-255` (decimal) or `0-F` or `00-FF` (hex)
- Black

  `rgb(0, 0, 0)` or `#000` or `#000000`

- White

  `rgb(255, 255, 255)` or `#FFF` or `#FFFFFF`

- Purple

  `rgb(128, 0, 255)` or `#8000FF`

Color Inspiration

- Adobe Kuler
  https://kuler.adobe.com/explore/
- COLOURlovers
  http://www.colourlovers.com/

background-color, color

```
h1 {
  color: #fff;
  background-color: #075488;
}
```

Use a CSS selector to delimit the scope of the declarations

CSS Borders

- Certain elements can have a border
- A border has the following properties:

  style

  width

  color

CSS Borders

- Shorthand declaration

  border: <width> <style> <color>

- Specific properties

  border-style, border-width, border-color

  border-top, border-right, border-bottom, border-left

  border-top-style, border-top-color … etc

More CSS

https://trinket.io/html/e3ef9c4225

LINKÖPINGS UNIVERSITET

Selectors

LINKÖPINGS UNIVERSITET

Selecting your selector

- **Targeting a group of elements**

  "select all paragraphs and list items"

- **Targeting adjacent siblings**

  "select all paragraphs that directly follow a heading"

- **Targeting descendants**

  "select any image that is inside a <article>"

- **Targeting children**

  "select all first level list items in unordered lists with the class 'toc'"

http://www.w3.org/TR/selectors/

Select a group of elements

```css
/* Target all h1, h2 and h3 element */

h1, h2, h3 {
    border: 2px solid #000;
}
```

Descendant combinator

```css
/* Select all li element that are nested within a nav element. */

nav li {
    color: #F00;
}
```

Child combinator

```css
/* Target all p elements that are children of a div */

div > p {
    border: 2px solid #000;
}
```

Adjacent sibling combinator

```css
/* Target all p elements that are on the same level as a h1 and follow
a h1 */

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

Pseudo selectors

- :hover

  e.g. a:hover

- :visited

  e.g. a:visited

Selector demonstration

LINKÖPINGS UNIVERSITET

Classes and

id:s

LINKÖPINGS UNIVERSITET

What is a class? When should I use it?

- Elements can be assigned one or more classes
- More than one element can be assigned the same class.
- Use classes for recurring components of your web page

What is an id? When should I use it?

- Elements can be assigned an id
- An id can only be assigned to a single element in a HTML document.
- Use ids for unique elements on your page that you want to target for a specific style.

Selectors using classes and ids

```
.infobox {
    font-family: Helvetica, Arial, Sans-Serif;
    font-size: 0.9em;
    background-color: #999;
    color: #000;
    border: 2px solid black;
}

#menu {
    background-color: #000;
    color: #FFF;
}
```

LINKÖPINGS UNIVERSITET

The CSS Box & Layout Model

http://reference.sitepoint.com/css/csslayout

http://docs.webplatform.org/wiki/css/tutorials#CSS_layout

http://learnlayout.com/toc.html

LINKÖPINGS UNIVERSITET

Formatting contexts

- Elements can be set to be displayed either in the block context, or the inline context.
- The inline context is “in line with text” with the generic inline element being `<span>`
- The block context is “outside text” with the generic block element being `<div>`. A block element occupies an area from left to right.

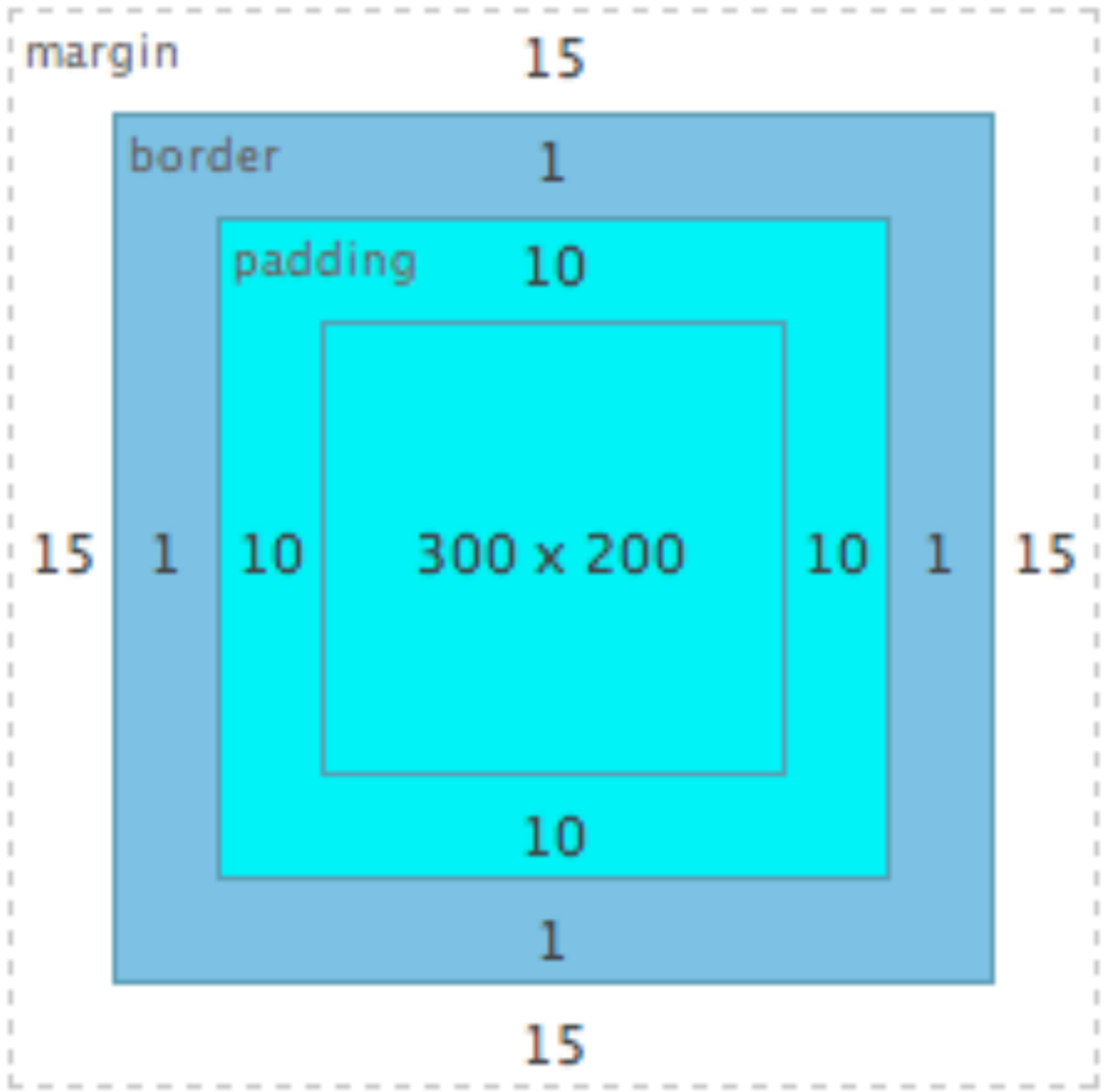
The display property

```css
/* The formatting context is set using the display property */

.infobox {
    display: block;
}

.question {
    display: inline;
}
```

The CSS box model (block context)



Specifying an elements padding

- `padding: <north>, <east>, <south>, <west>`
- `padding-top: <value>;`
- `padding-right: <value>;`
- `padding-bottom: <value>;`
- `padding-left: <value>;`

Specifying an elements margin

- `margin: <north>, <east>, <south>, <west>`
- `margin-top: <value>;`
- `margin-right: <value>;`
- `margin-bottom: <value>;`
- `margin-left: <value>;`

Positioning

http://learnlayout.com/position.html

LINKÖPINGS UNIVERSITET

Layout using positioning

- Blocks are statically positioned by default

```
position: static
```

- Relative positioning adjusts the static position relatively

```
position: relative;
top: -20px;
left: 20px;
```

Layout using positioning

- A block can be fixed to a position relative to the viewport

```
position: fixed;
bottom: 0px;
right: 0px;
```

- Elements positions using “absolute” are positioned relative to the nearest positioned ancestor.

```
position: relative;
top: -20px;
left: 20px;
```

Floating stuff

LINKÖPINGS UNIVERSITET

Floating

- Floating removes an element from the document flow - think floating image in e.g. Microsoft Word.
- An element can e.g. be floated left or right.
- Floating is relative to the elements containing block.

Responsive design

- Respond to the display used to render a HTML document

  high resolution desktop

  tablet

  smartphone

  etc

Responsive design

- A design can respond by e.g.

  altering sizes of images

  adjusting column widths

  adjusting number of columns used

  placement of navigation

  etc