729G26 Interaction Programming

Lecture 2

LINKÖPINGS UNIVERSITET

Lecture overview

- Interplay between HTML and CSS
- JavaScript
- Development tools

LINKÖPINGS UNIVERSITET

HTML/CSS Interplay

CSS is about colors, shapes, typography and layout, but without content to apply these attributes to, CSS is nothing.

Aspects of CSS styling

- **Appearance**:
  - Color
  - Size
  - Shape
  - Typography
  - ...
- **Layout**
  - Position on the webpage
  - Ordering with respect to what is on top and what is below
  - Behavior when the window resizes
  - ...

Structural aspects

- **The DOM**

  Which nodes are **contained** by which elements

  What is the **element type** of each node in the DOM

  Which **classes**, if any, have been assigned to each node in the DOM

  What **ID** if any, has been assigned to each node in the DOM

What can JavaScript do?

In the world of HTML and CSS, JavaScript can do **anything**.

JavaScript can be used to

- **Manipulate the DOM, i.e.**
  - Change DOM structure
  - Change the ID of DOM nodes
  - Change the classes of DOM nodes
  - Change contents of DOM nodes
- **Manipulate the CSS**
  - Add inline CSS to the DOM
  - Change the contents of the stylesheet
- **Capture web browser events**
  - changes to the web browser window (size, active/not active, URL etc)
  - changes originating from input devices: mouse movement, keyboard key presses, single and multi-touch events

LINKÖPINGS UNIVERSITET

CSS Selectors

specify which nodes in the DOM
should be affected by the declarations

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/

LINKÖPINGS UNIVERSITET

Select a group of elements

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

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

Descendant combinator

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

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

LINKÖPINGS UNIVERSITET

Child combinator

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

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

Adjacent sibling combinator

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

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

LINKÖPINGS UNIVERSITET

Pseudo classes

- Pseudo classes added to selectors to target specific states elements. Two examples of pseudo classes are `:hover` and `:visited`.
- `:hover`

  Used to style the state of an element when the mouse is hovering over it e.g. `a:hover` to style how a link looks like when the mouse cursor is over it.
- `:visited`

  E.g. used to style a link that has been visited. `a:visited`

LINKÖPINGS UNIVERSITET

Classes and id:s

DOM node independent structure

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

LINKÖPINGS UNIVERSITET

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

CSS layout

margin and padding

display

position

float

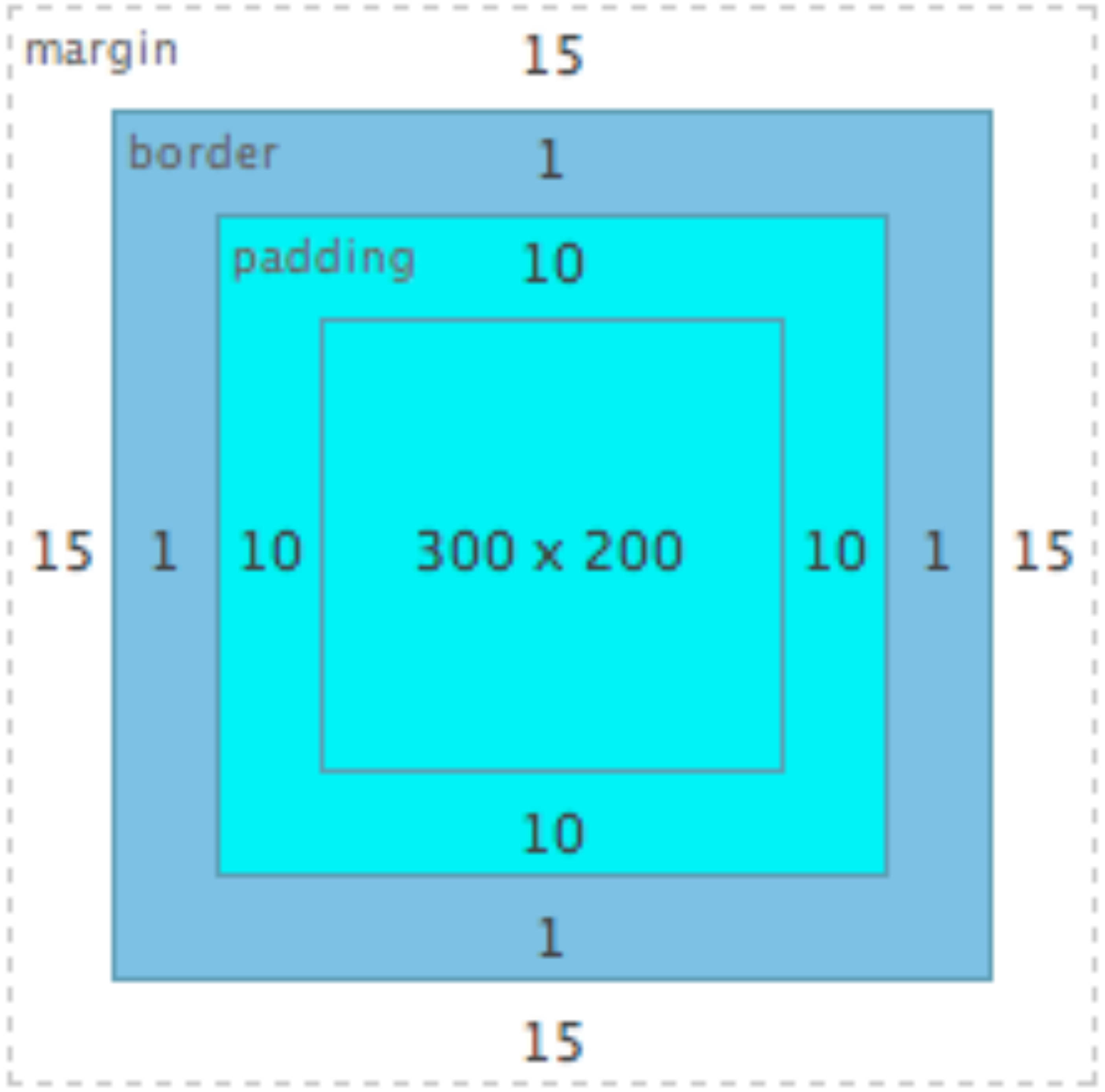
The display property

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

.infobox {
    display: block;
}

.question {
    display: inline;
}
```

The CSS box model (block context)



LINKÖPINGS UNIVERSITET

Specifying an elements padding

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

LINKÖPINGS UNIVERSITET

Specifying an elements margin

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

LINKÖPINGS UNIVERSITET

position

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;
```

float

LINKÖPINGS UNIVERSITET

Float

- Float 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.

```
float: left
float: right
```

- Float is relative to the elements containing block.

LINKÖPINGS UNIVERSITET

Responsive design

- Respond to the display used to render a HTML document
  - high resolution desktop
  - tablet
  - smartphone
  - ...
- Examples of responsive adaptations are:
  - changing the layout of the page
  - changing the sizes of elements on the page