Generated and inserted content

MGR. DANIELA PONCE, PHD.

(2019)

Lecture content

- Advanced CSS selectors
- Content generated by CSS rule
- Animation with @keyframes rules
- Canvas, svg graphics elements

Advanced selectors

- Basic selectors:
  - Element (h1)
  - Class (.mojeTrida)
  - Individual (#mujElement)
- Pseudoelements:
  - ::first-letter, ::first-line
- Pseudoclasses:
  - :link, :visited, :focus, :hover, :active
  - :disabled, :checked

- Complete list of pseudoselectors: https://www.w3schools.com/css/css_pseudo_classes.asp

Usage examples

```
p::first-letter {font-size: 200%; color: green;}
```

My name is Donald.

I live in Duckburg.

My best friend is Mickey.

Media query is a CSS technique introduced in CSS3.

It uses the @media rule to include a block of CSS properties only if a certain condition is true.

Note: For this selector to work in IE 5.5-8, you must specify the old, single-colon CSS2 syntax (:first-line instead of ::first-line).

```
p::first-line {background-color: yellow;}
```

```
input:focus {
background-color: yellow;
}
```

First name:

Jan

Last name:

Mal

Submit

Example: link colors depending at user action

- a:link {color: red;} /* unvisited link */
- a:visited {color: green;} /* visited link */
- a:hover {color: orange;} /* mouse iver the link */
- a:focus {color: purple;} /* link with focus, e.g. tab key*/
- a:active {color: blue;} /* selected link */
- Color formating will work properly if the rule declaration order is preserved (LVFHA nebo LVHFA)

Selecting nth element with pseudoclass selectors

- :nth-child(X) .. Every nth element from all descendents of the parent
- :nth-of-type(X) .. Every nth element of selected type from all descendents of selected type
- Elegible values of the parameter X: number, even, odd, ($an + b$) expression
- Examples:
  - Every 3rd paragraph: p:nth-of-type(3n+0)
  - Every even row of the table: tr:nth-child(even)

Selection of elements by attribute value

- [lang] .. element with attribute with any value specified
- [lang=“cs“] .. With exact value
- [title~=flower] .. With a word from a list (before and after a space is allowed only)
- [href*=“http“] .. With a part of value
- [class$=“test“] .. Ending with a part of value
- [class^=“top“] .. Starting with a part of value
- [class|=“top“] .. With a word at the beginning (can be followed by a space or - only)

- Combination with other selectors: simple concatenation of symbols
  - a[href*=“http“]

Examples of selectors based on attributes

- All images with the alt attribute with a value specified:
  - img[alt]
- All links pointing outside of the page:
  - a [href*=http]
- All elementy with an id starting with my:
  - [id^=my]
- All elements in English (en, en-us)
  - [lang|=en]

Combination of operators by the + operator

„operator of neighbouring sibling“

elementX+elementX

- Every element, which is proceeded by a sibling element of the same type
- Example:
  - li+li will select all items in a list except the first one

elementX+elementY

- Every elementY, which is proceeded by a sibling elementX
- Example:
  - h2+p will select all paragraphs that follow immediately after a second-level heading

Seasons

- Spring
- Summer
- Autumn
- Winter

```
li+li {
    color: yellow;}
```

::before and ::after pseudoelements selectors

- Pseudolement ::before
  - Inserts content before the element
- Pseudoelement ::after
  - Inserts content after the element
- Used together with the content property, which contains the content to be inserted

- Pseudoelement selectors can not be combined into one selector:
  - Not allowed: p::first-letter::before
  - Not allowed: p::first-line::after
  - …

Generated content

- CCS content property
- Generates content which is not included in the HTML document
- Example: displaying attribute of an element (here target of the link)
  - a::before {content: attr(href);}
- Example: quoting the text of an element (here for a paragraph)
  - p::before {content: open-quote;}
  - p::after {content: close-quote;}
- Example: adding text content
  - p::before {content: "New!";}
- Example: adding non-text content
  - p::before {content: url(w3css.gif);}

```
ul {list-style: none;}
li {display: inline;}
li+li::before {content: ‘***‘;}
```

spring *** summer *** autumn *** winter

Generated indexing of items

- Content property
- Value of the counter function
- The counter has to be initialized by the counter-inkrement property
- Example: numbering all paragraphs from the class message on the page
  1. p.message {counter-increment: myCounter;}
  2. p.message::before {content: counter(myCounter);}
- Any valid selector can be used

Combination of + selector with pseudoclass selector

li+li::before

- All but first items of a list will be selected
- A content (e.g. ***) will be added before each item selected (selector ::before)
- The result: a list with visually separated items

spring *** summer *** autumn *** winter

CSS functions and „variables “

- CSS functions: rgb(), hsl(), rgba(), hsla()
- User can define his/her own function (calculation)
- User can define custom CSS properties
- Custom properties are used as „variables“
- Example:
  - :root { --myBackgroundColor: coral; }
  - div { background-color: var(--myBackgroundColor);}
- First define the attribute, next use the attribute in a function
- The name of the attribute beginns with **--** and it is case sensitive

Using user-defined functions

- It is a good practice to specify a fallback value (not required):
  - var(custom-name, value)
- Example:
  - .myDiv { background-color: var(-- myBackgroundColor, coral);}

- Remember that attribute values can be inherited
- Improper usage of function may cause invalid values of attributes of other elements
- If var() returns invalid value, initial or inherited value will be used

Prefix attributes

- For elements that are not specified in W3C recommendations yet
- CSS attributes with a partial support from browsers (not implemented by all browsers)
- Testing attribute with a vendor prefix
- Every browser uses its own prefix:
- -webkit- …Safari, Chrome (with WebKit engine)
- -moz- .. Firefox
- -ms- .. Internet Explorer
- -o- .. Opera
- Support can be checked at Can I use (supported versions, known bugs, browser usage shares, etc.)

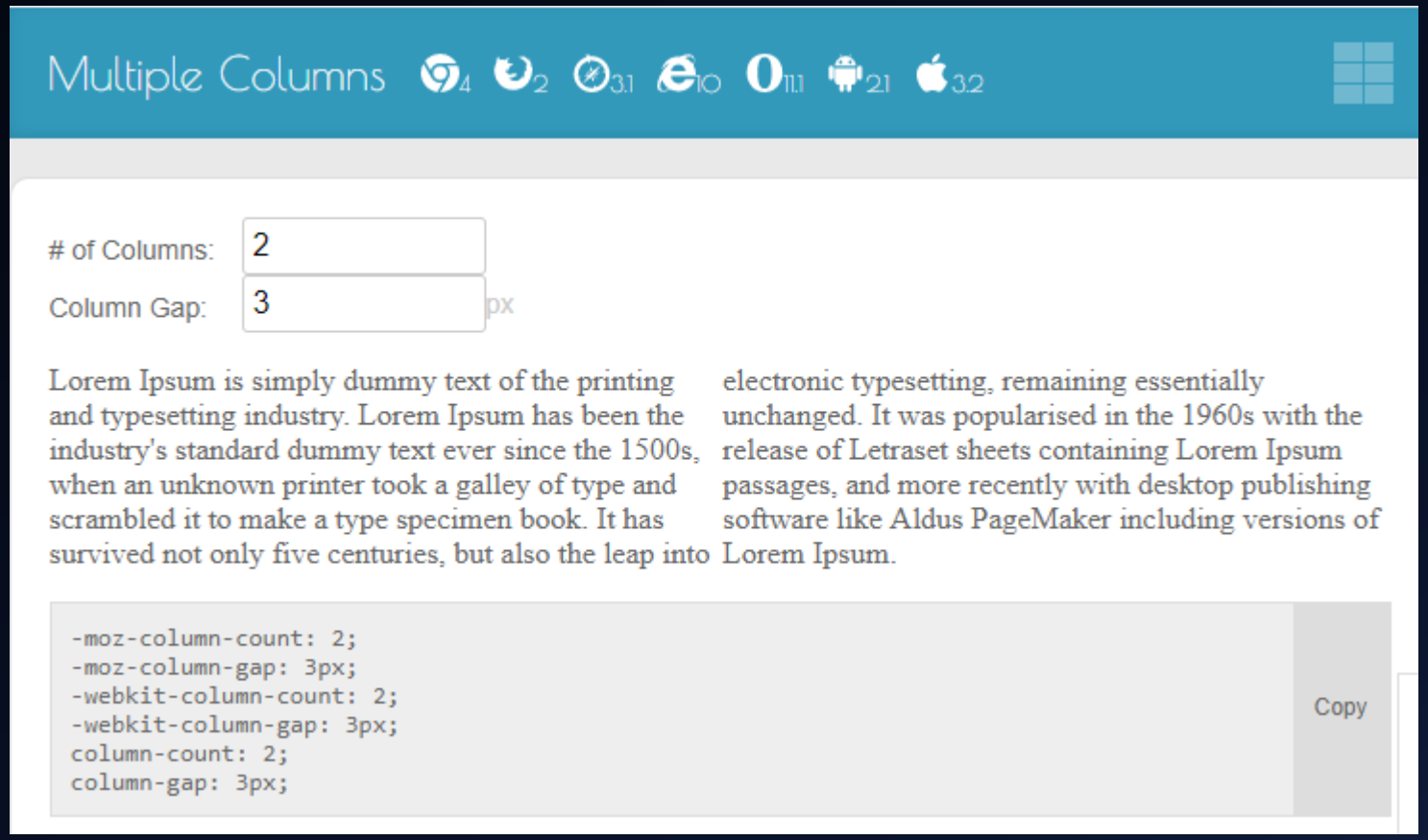
Example of usage of prefix attributes

```
div {
-moz-border-radius: 10px;        /* older Firefox versions */
-webkit-border-radius: 10px;     /* older Safari and Chrome versions */
border-radius: 10px;             /* new versions */
}
```

- Every browser ignores those parts it cannot understand
- Every browser interprets those part it can understand
- The last declaration wins, therefore the newest version (with no prefices) is at the end

Automatic generation of rules with vendor-prefices

- Special rule generators can be used, e.g. CSS3Generator



Animation definition in CSS

- The @keyframes rule defines an animation and gives it a name
- Animation is created as a gradual change of one setting set to some other setting set
- The setting set can be changed in several steps during the animation
- In one step, several attributes can be changed simultaneously
- Changepoints can be specified as % or "from" (identical with 0% value) and "to" (identical with 100% value) keywords can be used
- 0% is state at the beginning of the animation
- 100% is state at the end of the animation

Example of animation definition

- Several points of change, simultaneous change of several attributes

```
@keyframes myAnimation {
   0%   {top: 0px; left: 0px; background: red;}
   25%  {top: 0px; left: 100px; background: blue;}
   50%  {top: 100px; left: 100px; background: yellow;}
   75%  {top: 100px; left: 0px; background: green;}
   100% {top: 0px; left: 0px; background: red;}
}
```

Using animation with the animation property

- Before animation can be used, it is necessary to define it by the @keyframes rule
- Then, it is necessary to relate it to the element for which it was defined
- Animation property
  - Subproperties name, duration, timing-function, delay, iteration-count, direction, fill-mode, play-state
- Details at w3schools

Example:

```
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: myAnimation;
    animation-duration: 4s;
}
```

Which properties can be animated?

- Not all properties can be animated
- Animatable properties (complete list)
- The value can be other than numeric
- Examples: background (image), color, border (style)

Inserted content

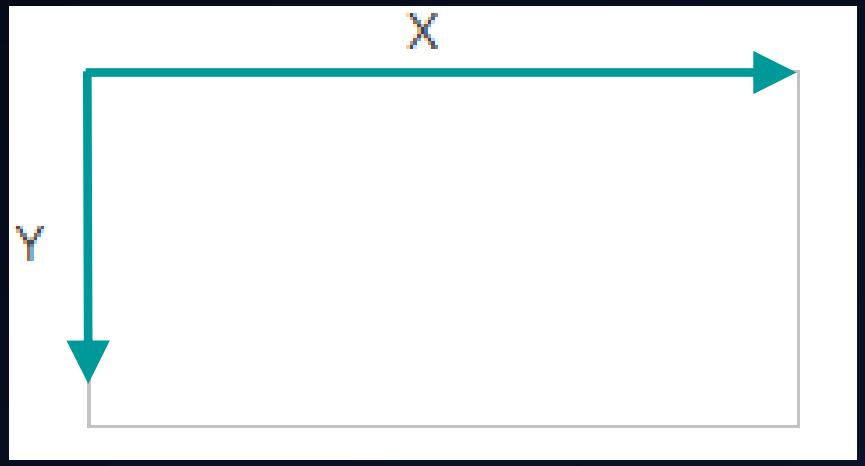
- Visual content defined in its own (= non HTML) language and inserted in the HTML document in a suitable HTML element
- svg and canvas HTML elements

Canvas

- Dynamic drawing of grid graphics and animations
- The <canvas> element is a container only
- JavaScript is used for the drawing itself
- Basic functions for graphical basics drawing (curves, rectangles, text etc.)
- Example – creation of canvas:
  - <canvas id="myCanvas" width="200" height="100">This text is shown because your browser does not support the canvas element.</canvas>

Canvas

- A system of two-dimensional coordinates
- Left top corner has (0,0) coordinates
- Canvas enables to draw other graphics:
  - Transitions, moving object, objects respoding to user actions
- Result of the drawing can be saved as a common image



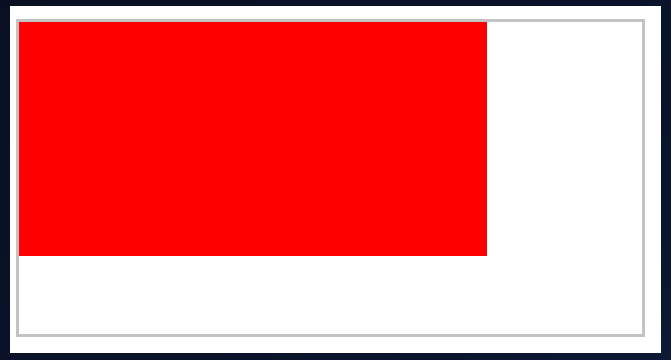
Drawing in the canvas

A script is needed, with drawing commands

Example:

```
<canvas id="myCanvas" width="200" height="100">…</canvas>

<script>
        var canvas = document.getElementById("myCanvas");
        var ctx = canvas.getContext("2d");
        ctx.fillStyle = "#FF0000";
        ctx.fillRect(0,0,150,75);
</script>
```



More information about Canvas

- Can ne used for games in HTML (W3Schools game tutorial)
- More details:
  - W3Schools canvas tutorial
  - W3Schools example - clock

SVG

- SVG = Scalable Vector Graphics
- Image in SVG format is placed in the <svg> HTML element in the document (can be nested)
- Image is defined in the SVG language (sublanguage of XML), in this format it is also stored in browser‘s memory
- Size change does not cause quality loss
- Every object/property can be animated
- Recommended by W3 consorcium
- Printable in any resolution and size

SVG - example

```
< svg height="190">
    <polygon points="100,10   40,180   190,60   10,60   160,180"
        style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;">
< /svg>
```

- SVG drawings can be created in any text editor
- More effective and convenient: in vector graphics editor (e.g. INKSCAPE)

Canvas and SVG (comparison)

CANVAS

1. Depends on resolution
2. No event support
3. Well suitable for games
4. After rendering no reference to the element
5. Element modification requires redrawing of all elements
6. Text rendering with problems
7. Can be saved (in.png or .jpg formats)

SVG

1. Independent on resolution
2. Event support
3. Not suitable for games
4. Reference to the element
5. Element modification is simple
6. Very suitable for applications with big canvas
7. Slow, if DOM is complex