



Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko

CSS



CSS: Kaskadne Slogovne Predloge



CSS - Cascading Style Sheets

- ▶ HTML je prvenstveno zadolžen za vsebino dokumenta
- ▶ CSS je sintaktični mehanizem za specifikacijo informacije o slogih
- ▶ CSS je način kontrole **predstavitve** HTML dokumentov
- ▶ je sintaksa, ločena od HTML, vendar lahko vsebovana v HTML dokumentih
 - validator: <http://jigsaw.w3.org/css-validator/>
- ▶ omogoča:
 - enoten slog **celega dokumenta** ali **množice dokumentov**
 - **ločevanje dela** razvijalcev in oblikovalcev
- ▶ verzije:
 - CSS1 razvit leta 1996
 - CSS2 razvit leta 1998
 - CSS 2.1 uporabljajo današnji brskalniki
 - CSS3 je v fazi razvoja

CSS₃ TEXT LAST CALL FOR COMMENTS

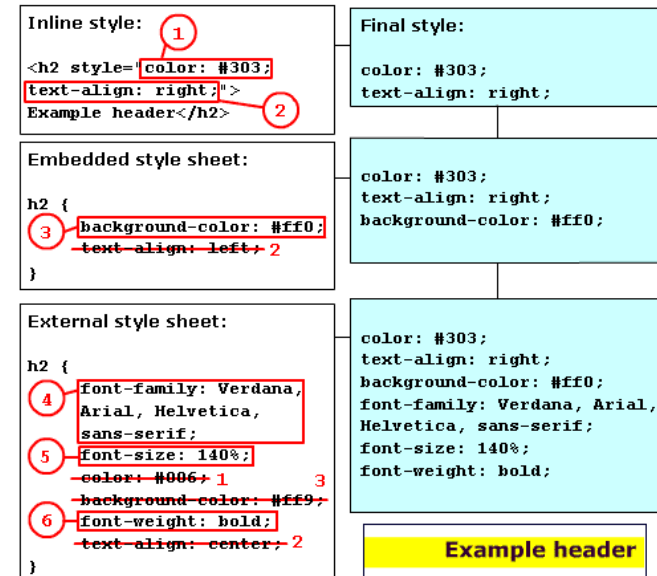
By fantasai [October 12, 2013](#) ([Permalink](#))

Categories: **publications**

The W3C CSS Working Group has published a Last Call Working Draft of the **CSS Text Module Level 3**. This module covers various aspects of text layout including white space processing, text transformations, line breaking, justification, and indentation, including additions to level 2 to address internationalization concerns, allowing better typography in non-Western (and Western) scripts. Significant changes since the previous are listed in the [changes section](#)

CSS - Cascading Style Sheets

- ▶ CSS = *Kaskadne slogovne pole (predloge, prekrivni slogi)*
- ▶ kaskada treh nivojev definiranja sloga
 - **vrstične definicije** (inline): določajo slog za posamezen element
 - **na nivoju dokumenta** (document-level): za elemente znotraj dokumenta
 - **zunanje slogovne pole** (external style sheets): za uporabo na množici dokumentov
- ▶ kadar je uporabljenih več nivojev definicij sloga, brskalnik uporabi **bolj specifično**: vrstična ima prednost pred definicijo na nivoju dokumenta; ta ima prednost pred zunanjo slogovno polo
- ▶ če slog za element ni določen, brskalnik uporabi privzeti slog zanj



Implementacija nivojev slogov

1. vrstične definicije (inline)

atribut značke `style` = "lastnost_1: vrednost_1; ... ;"

- zameglijo prednost uporabe slogovnih pol: uniformen izgled elementov v vsaj enem dokumentu
- dejansko ne ločijo predstavitev od vsebine, saj so vsebovane v dokumentu HTML
- to povzroči tudi mešanje dveh različnih jezikov oziroma sintaks: HTML in CSS

```
<p style = "font-size: 20pt;">This text has the  
  <em>font-size</em> style applied to it, making it 20pt.  
</p>
```

Implementacija nivojev slogov

2. na nivoju dokumenta

- shranjene so v znački `<style>` v glavi dokumenta
- omogočajo specifikacijo uniformnega sloga za cel dokument

```
<!-- Embedded style sheet. -->
<html>
  <head>
    <meta charset = "utf-8">
    <title>Embedded Style Sheet</title>

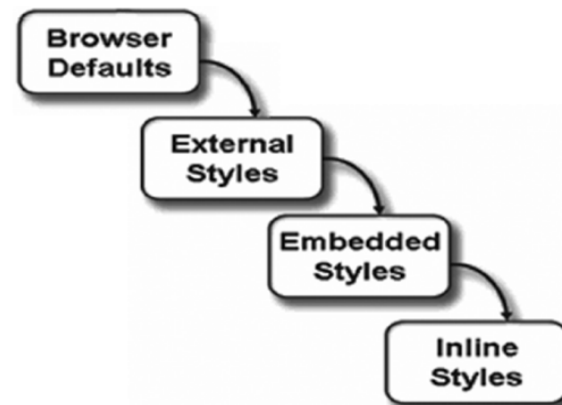
    <!-- this begins the style sheet section -->
    <style type = "text/css">
      em    { font-weight: bold;
              color: black; }
      h1    { font-family: tahoma, helvetica, sans-serif; }
      p     { font-size: 12pt;
              font-family: arial, sans-serif; }
      .special { color: purple; }
    </style>
  </head>
```

Implementacija nivojev slogov

3. zunanje slogovne pole (external style sheets)

- za uporabo na množici dokumentov
- seznam pravil v ločenih datotekah, lahko na poljubnem strežniku
- povezava na slogovno polo se mora pojaviti v glavi dokumenta
- dobra ločitev med vsebino (HTML) in slogom (CSS)

```
<head>  
  <meta charset = "utf-8">  
  <title>Linking External Style Sheets</title>  
  <link rel = "stylesheet" type = "text/css"  
    href = "styles.css">  
</head>
```



Format podajanja slogov

- Format je odvisen od nivoja slogovne pole

1. vrstične definicije (inline):

splošna oblika:

```
style = "property_1: value_1; property_2: value_2; ...  
        property_n: value_n; "
```

2. na nivoju dokumenta:

splošna oblika:

```
<style>  
    selector {property_1: value_1; property_2: value_2; ...  
              property_n: value_n; }  
</style>
```

3. zunanje slogovne pole (external style sheets):

splošna oblika:

```
selector {property_1: value_1; property_2: value_2; ...  
          property_n: value_n; }
```

Določanje slogovnih pravil

- ▶ s seznamom elementov
 - `h1, h3 {font-size: 20pt;}`
- ▶ s seznamom elementov v kontekstu (gnezdenju) - naslednikov
 - `form em {font-size: 14pt;}`
- ▶ s seznamom elementov v kontekstu - neposrednih naslednikov
 - `form > p > em {font-size: 14pt;}`
- ▶ glede na vrstni red (zaporedje) – za oziroma pred elementom
 - `div + p {font-size: 14pt}`
 - `div ~ p {font-size: 14pt}`
- ▶ z definiranjem razredov sloga za določene elemente
 - `p.normal {color: blue;}`
`p.warning {color: red;}`
 - znački določimo razred z atributom class, npr. `<p class="warning">`

Selector

Declaration
Property

Declaration
Value

`body { color: blue }`



Določanje slogovnih pravil

- ▶ z definiranjem generičnega razreda sloga
 - definiramo razred kot npr. `<h3 class="poudarjeno">` in `<p class="poudarjeno">`
 - določimo slog za ta razred `.poudarjeno {font-size: 20px;}`
- ▶ za značko z določenim id-jem
 - določimo id značke npr. `<h2 id="dolocen_id">`
 - določimo slog za značko z nekim id-jem `#dolocen_id {...}`
- ▶ za vse elemente, ki imajo določeno vrednost atributa:
 - `[atribut=vrednost] {font-size: 12px;}`
- ▶ za vse elemente v dokumentu
 - `* {color: red;}`
- ▶ ob dogodkih nad elementi (psevdo-razredi, psevdo-elementi)
 - ker se nekaj zgodi nad elementom/ker je določen del elementa
 - `input:hover {...}` – zgodi se, ko je kurzor nad elementom
 - `input:focus {...}` – zgodi se, ko ima element fokus
 - `link:visited`, `hover` mora priti za ostalima psevdo-razredoma (za element a)
 - `p::after {...}` – doda vsebino za izbrane elemente



Določanje slogovnih pravil

```
<!DOCTYPE HTML>
<html>
  <head> <title>Prikaz selektorjev</title> <meta charset="utf-8"/>
    <style type="text/css">
      h1 {font-size: 10px;}
      #prvi {font-size: 22px;}
      .drugi {color: orange;}
      [name=tretji] {font-size: 12px;}
      div>span {font-size: 30px;}
      div span {color: green;}
      a:hover { font-size: 30px;}
      p::first-letter{font-size: 25px; color: magenta;}
      div::first-letter{font-size: 55px; color: blue;}
      p::after{ content: " - Vsebina, ki jo dodamo na konec";}
      div::before{ content: " Dodatek na začetek ";}
    </style>
  </head>
  <body> <h1>Test CSS selektorjev</h1>
    <p id="prvi">Zivel je moz</p>
    <p class="drugi">imel je psa</p>
    <p name="tretji">lepo ga je ucil</p>
    <div> Nekdo je <span>nekje nekaj</span> naredil</div>
    <div> <p>Nekdo je <span>nekje nekaj</span> naredil</p></div> <hr>
    <address><a href="mailto:nekdo@nekje.si">Nekdo, kdo pa drug</a></address>
  </body> </html>
```

Test CSS selektorjev

Zivel je moz - Vsebina, ki jo dodamo na konec

imel je psa - Vsebina, ki jo dodamo na konec

lepo ga je ucil - Vsebina, ki jo dodamo na konec

Dodatek na začetek Nekdo je nekje nekaj naredil

Dodatek na začetek

Nekdo je nekje nekaj naredil - Vsebina, ki jo dodamo na konec

Nekdo, kdo pa drug

Vrste slogovnih pravil

- ▶ 60 lastnosti v (približno) sedmih kategorijah (CSS1):
 - fonts (vrste pisav), lists (seznam), colours (barve), text alignment (poravnava besedila), margins (robovi), borders (obrobe), background (ozadje)
- ▶ dodatnih 34 lastnosti v prejšnjih sedmih in dodatnih šestih kategorijah (CSS2):
 - writing modes (načini pisanja), tables (tabele), basic user-interface (osnovni vmesnik), multicolumn layout (večstolpična postavitve), paged media (ostranjeni mediji), generated content for paged media (generirana vsebina za ostranjene medije)
- ▶ dodatnih več kot 100 lastnosti v predhodnih kategorijah in dodatnih devetih kategorijah (CSS3):
 - animation (animacija), transform (transformacije), transitions (prehodi), filter eff. (učinki filtriranja), image vals. (slikovne vrednosti), speech (govor), ...



Vrste slogovnih pravil

- ▶ CSS1: 60 lastnosti v sedmih kategorijah
 - te lastnosti so najpogostejše uporabljane
- ▶ vrednosti določamo s ključnimi besdami (npr. `left`), merami:
`px`, `in`, `cm`, `mm`, `pt`, `pc` (npr. `100px`) – absolutne vrednosti
`em`, `ex` – relativne vrednosti
odstotki (npr. `50%`) – odstotek predhodne/privzete vrednosti atributa
- ▶ barve določamo z imenom ali paleto RGB
- ▶ URL naslove podamo kot `url(...)`, npr.
`body { background-color: #FFFFFF url('smiley.gif') no-repeat right top; }`



1. Vrste pisav

- ▶ `font-family`: Arial, Helvetica, Futura, sans-serif
brskalnik uporabi enega izmed teh fontov glede na podan vrstni red

- ▶ generične družine:

Generic Name	Examples
serif	Times New Roman, Garamond
sans-serif	MS Arial, Helvetica
cursive	Caflisch Script, Zapf-Chancery
fantasy	Critter, Cottonwood
monospace	Courier, Prestige

- ▶ `font-size`: velikost (`xx-small` | `x-small` | `small` | `medium` | `large` | `x-large` | `xx-large`)
velikost je lahko 1.2em, 12pt, 14pc
- ▶ `font-variant`: `small-caps` | `normal`
- ▶ `font-style`: `italic` | `oblique` | `normal`
- ▶ `font-weight`: `bold` (`normal`, `bolder`, `lighter`; številsko 100-900, 400=`normal`, 700=`bold`)
- ▶ skrajšano (vrstni red je pomemben): `font`: `bolder 14pt Arial Helvetica`
- ▶ `text-decoration`: `line-through` | `overline` | `underline` | `none`
- ▶ `letter-spacing`, `word-spacing`, `line-height`: `normal` | `1px` | `-1em` ...



2. Slogi seznamov

- ▶ znački `` ali `` lahko določimo slog ali sliko
`list-style-type: disc | circle | square | none`
`list-style-image: url('slika.gif')`
- ▶ številčeni sezname imajo lahko različne sloge številčenj:
`list-style-type:`
- ▶ sezname uporabimo za definiranje menujev

Property Values	Sequence Type
decimal	Arabic numerals starting with 1
decimal-leading-zero	Arabic numerals starting with 0
lower-alpha	Lowercase letters
upper-alpha	Uppercase letters
lower-roman	Lowercase Roman numerals
upper-roman	Uppercase Roman numerals
lower-greek	Lowercase Greek letters
lower-latin	Same as lower-alpha
upper-latin	Same as upper-alpha
armenian	Traditional Armenian numbering
georgian	Traditional Georgian numbering

- ▶ določimo bodisi z:

- ### RGB Colour Codes

#000000	#830000	#460000	#990000	#CC0000	#FF0000	#110000	#001000
#000300	#833000	#463000	#993000	#CC3000	#FF3000	#113000	#001300
#000600	#836000	#466000	#996000	#CC6000	#FF6000	#116000	#001600
#000900	#839000	#469000	#999000	#CC9000	#FF9000	#119000	#001900
#000C00	#83C000	#46C000	#99C000	#CCC000	#FFC000	#11C000	#001C00
#000F00	#83FF00	#46FF00	#99FF00	#CCFF00	#FFF000	#11FF00	#001FF0
#000033	#830033	#460033	#990033	#CC0033	#FF0033	#110033	#001033
#000066	#830066	#460066	#990066	#CC0066	#FF0066	#110066	#001066
#000099	#830099	#460099	#990099	#CC0099	#FF0099	#110099	#001099
#0000CC	#8300CC	#4600CC	#9900CC	#CC00CC	#FF00CC	#1100CC	#0010CC
#0000FF	#8300FF	#4600FF	#9900FF	#CC00FF	#FF00FF	#1100FF	#0010FF
#000333	#830333	#460333	#990333	#CC0333	#FF0333	#110333	#001033
#000366	#830366	#460366	#990366	#CC0366	#FF0366	#110366	#001066
#000399	#830399	#460399	#990399	#CC0399	#FF0399	#110399	#001099
#0003CC	#8303CC	#4603CC	#9903CC	#CC03CC	#FF03CC	#1103CC	#0010CC
#0003FF	#8303FF	#4603FF	#9903FF	#CC03FF	#FF03FF	#1103FF	#0010FF
#000666	#830666	#460666	#990666	#CC0666	#FF0666	#110666	#001066
#000699	#830699	#460699	#990699	#CC0699	#FF0699	#110699	#001099
#0006CC	#8306CC	#4606CC	#9906CC	#CC06CC	#FF06CC	#1106CC	#0010CC
#0006FF	#8306FF	#4606FF	#9906FF	#CC06FF	#FF06FF	#1106FF	#0010FF
#000999	#830999	#460999	#990999	#CC0999	#FF0999	#110999	#001099
#0009CC	#8309CC	#4609CC	#9909CC	#CC09CC	#FF09CC	#1109CC	#0010CC
#0009FF	#8309FF	#4609FF	#9909FF	#CC09FF	#FF09FF	#1109FF	#0010FF
#000CCC	#830CCC	#460CCC	#990CCC	#CC0CCC	#FF0CCC	#110CCC	#0010CC
#000CFF	#830CFF	#460CFF	#990CFF	#CC0CFF	#FF0CFF	#110CFF	#0010FF
#003333	#833333	#463333	#993333	#CC3333	#FF3333	#113333	#001333
#003366	#833366	#463366	#993366	#CC3366	#FF3366	#113366	#001366
#003399	#833399	#463399	#993399	#CC3399	#FF3399	#113399	#001399
#0033CC	#8333CC	#4633CC	#9933CC	#CC33CC	#FF33CC	#1133CC	#0013CC
#0033FF	#8333FF	#4633FF	#9933FF	#CC33FF	#FF33FF	#1133FF	#0013FF
#006666	#836666	#466666	#996666	#CC6666	#FF6666	#116666	#001666
#006699	#836699	#466699	#996699	#CC6699	#FF6699	#116699	#001699
#0066CC	#8366CC	#4666CC	#9966CC	#CC66CC	#FF66CC	#1166CC	#0016CC
#0066FF	#8366FF	#4666FF	#9966FF	#CC66FF	#FF66FF	#1166FF	#0016FF
#009999	#839999	#469999	#999999	#CC9999	#FF9999	#119999	#001999
#0099CC	#8399CC	#4699CC	#9999CC	#CC99CC	#FF99CC	#1199CC	#0019CC
#0099FF	#8399FF	#4699FF	#9999FF	#CC99FF	#FF99FF	#1199FF	#0019FF
#00CCCC	#83CCCC	#46CCCC	#99CCCC	#CCCCCC	#FFCCCC	#11CCCC	#001CCC
#00CCFF	#83CCFF	#46CCFF	#99CCFF	#CCCCFF	#FFCCFF	#11CCFF	#001CFF
#00FFFF	#83FFFF	#46FFFF	#99FFFF	#CCFFFF	#FFFFFF	#11FFFF	#001FFF
#00FF33	#83FF33	#46FF33	#99FF33	#CCFF33	#FFF033	#11FF33	#001F33
#00FF66	#83FF66	#46FF66	#99FF66	#CCFF66	#FF6066	#11FF66	#001F66
#00FF99	#83FF99	#46FF99	#99FF99	#CCFF99	#FF9099	#11FF99	#001F99
#00FFCC	#83FFCC	#46FFCC	#99FFCC	#CCFFCC	#FFCC99	#11FFCC	#001FCC
#00FFFF	#83FFFF	#46FFFF	#99FFFF	#CCFFFF	#FFFFFF	#11FFFF	#001FFF

Web-safe Colours

Black

#000000

Silver

#C0C0C0

White

#FFFFFF

Maroon

#8B0000

Red

#FF0000

Purple

#800080

Fuchsia

#FF00FF

Green

#008000

Lime

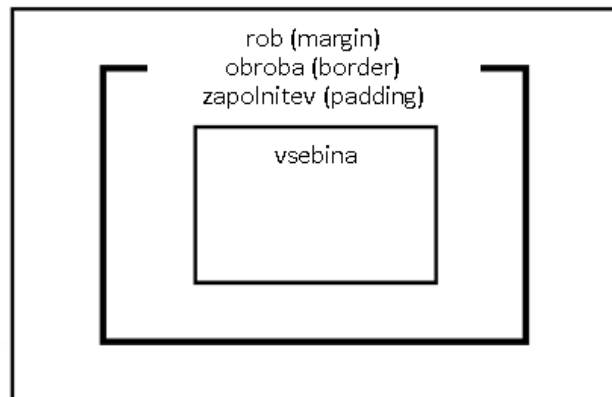
4. Poravnava besedila

- `text-indent`: zamik prve vrstice odstavka
- `text-align`: `left` | `center` | `right` | `justify`
- `float`: `left` | `right` kadar želimo ovijanje besedila okoli elementa
- ``

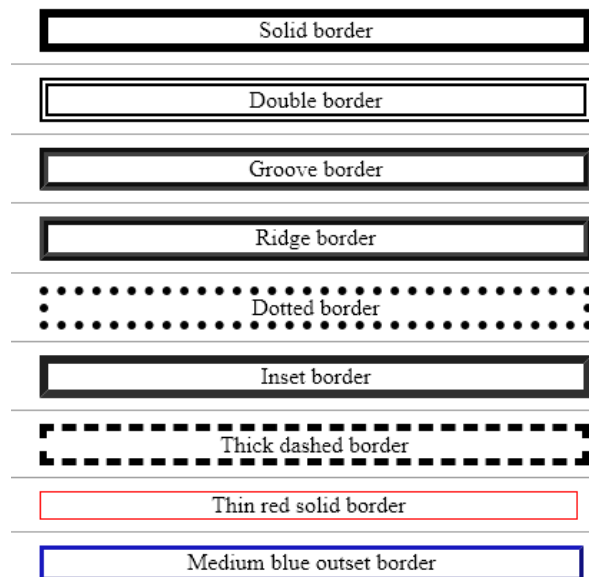
This is a picture of a Cessna 210. The 210 is the flagship single-engine Cessna aircraft. Although the 210 began as a four-place aircraft, it soon acquired a third row of seats, stretching it to a six-place plane. The 210 is classified as a high performance airplane, which means its landing gear is retractable and its engine has more than 200 horsepower. In its first model year, which was 1960, the 210 was powered by a 260 horsepower fuel-injected six-cylinder engine that displaced 471 cubic inches. The 210 is the fastest single-engine airplane ever built by Cessna.



5. Obrobe (borders) in 6. Robovi (margins)

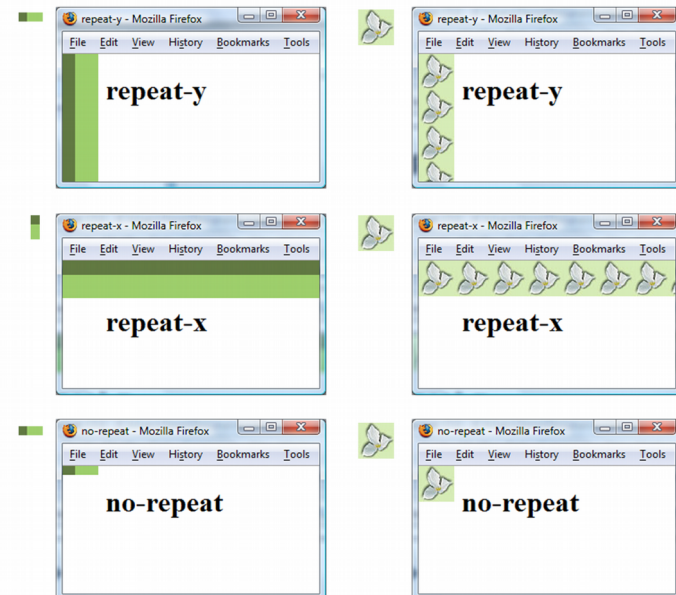


- ▶ `border-style: dotted | dashed | double | ...`
`border-width: thin | medium | thick`
`border-color`
- ▶ obstajajo tudi različice
`border-{top|bottom|left|right}-{style|width|color}`
`margin, margin-{top|bottom|left|right}` razmik med obrobo in okolico
`padding, padding-{top|bottom|left|right}` razmik med vsebino in obrobo



7. Ozadje

- ▶ Poleg barv lahko ozadju določimo tudi sliko
- ▶ `background-image: url('smilie.gif')`
- ▶ `background-size: x-size y-size`
- ▶ `background-repeat: repeat | no-repeat | repeat-x | repeat-y`
- ▶ `background-position:`
 `top | center |`
 `bottom | left | right`
 `[xposPX yposPX]`
- ▶ možnih tudi več slik ozadja
in ločeno pozicioniranje vsake



Pozicioniranje elementov

- ▶ **absolutno** - merjeno od zgornjega levega oglišča zunanega elementa

`<p style = "position: absolute; left: 50px; top: 100px;">`

- ▶ **relativno** - merjeno glede na naravno lokacijo elementa v dokumentu

`<p style = "position: relative; left: 50px; top: 100px;">`

- ▶ **statično** - privzeto, naravna lokacija elementa

`<p style = "position: static; ">`

- ▶ **fiksno** - fiksirano glede na prikazno okno (brskalnik)

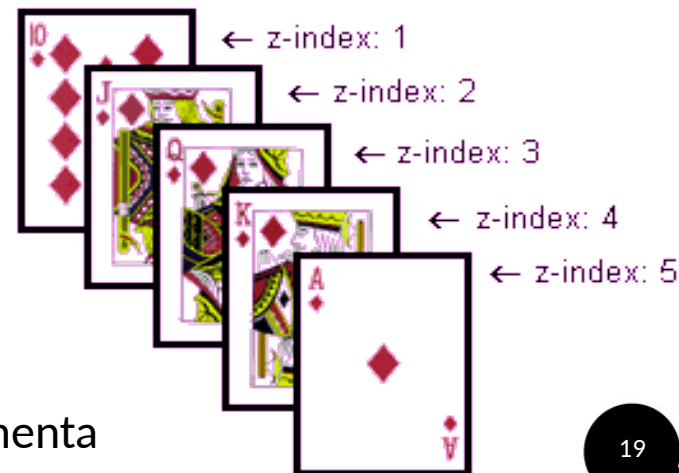
`<p style = "position: fixed; ">`

- ▶ **lepljivo** - pozicionirano glede na položaj drsnikov

`<p style = "position: sticky; top: 0px">`

- ▶ prekrivanje elementov:

višja vrednost atributa z-index pomeni "višjo" prekrivno lego elementa



Znački `` in `<div>`

- ▶ `` - za določanje posebnega sloga dela dokumenta, ki je manjši od odstavka (`<p>`)
- ▶ `<div>` - za določanje posebnega sloga odsekov dokumenta, ki obsegajo več odstavkov (`<p>`)

- ▶ Primer:

```
<style type = "text/css">
```

```
    .bigred {font-size: 24pt; font-family: Ariel; color: red}
```

```
</style>
```

```
...  
<p> Zdaj pa znamo že <span class = "bigred"> skoraj </span> vse! </p>
```

Zdaj pa znamo že **skoraj** vse!



`<div>`

Meniji na spletnih straneh

- ▶ za kreiranje menijev lahko uporabimo HTML kodo v kombinaciji z zunanjimi programskimi knjižnicami v JavaScript-u
- ▶ menije lahko ustvarimo tudi samo s HTML kodo in kaskadnimi slogovnimi predlogami
- ▶ uporabimo gnezdene sezname, kjer v posameznem (pod)seznamu združimo možnosti, ki sodijo skupaj, vse podseznime skrijemo in po potrebi prikazujemo

Prikaz uporabe in gnezdenja seznamov

- i. [Prvi](#)
- ii. [Drugi](#)
 - o [Drugi prvi](#)
 - o [Drugi drugi](#)
 - o [Drugi tretji](#)
 - A. [Drugi tretji prvi](#)
 - B. [Drugi tretji drugi](#)
 - C. [Drugi tretji tretji](#)
 - a. [Drugi tretji tretji prvi](#)
 - b. [Drugi tretji tretji drugi](#)
 - c. [Drugi tretji tretji tretji](#)
- iii. [Tretji](#)
- iv. [Cetrtri](#)



Prikaz uporabe in gnezdenja seznamov



Meniji na spletnih straneh

```
<style>
  ul,ol {list-style: none;}
  ul ul, ul ol, ol ul, ol ol {display: none;}
  ul li:hover > ul, ul li:hover > ol, ol li:hover > ul, ol li:hover > ol {display: block;}
  ul,ol {background: #efefef;
    background: -moz-linear-gradient(top, #efefef 0%, #bbbbbb 100%);
    background: -webkit-linear-gradient(top, #efefef 0%,#bbbbbb 100%);
    background: linear-gradient(top, #efefef 0%, #bbbbbb 100%);
    box-shadow: 0px 0px 9px rgba(0,0,0,0.15); padding: 0 20px; border-radius: 10px;
    list-style: none; position: relative; display: inline-table; }
  ul:after, ol:after {content: ""; clear: both; display: block;}
  ol li, ul li {float: left;}
  ul li:hover, ol li:hover { background: #555555;}
  ul li:hover a, ol li:hover a {color: #fff;}
  ul li a, ol li a {display: block; padding: 15px 10px; color: #757575; text-decoration: none;}
  ul ul, ol ul, ul ol, ol ol {background: #5f6975; border-radius: 0px; padding: 0;
    position: absolute; top: 100%; }
  ul ul li, ul ol li, ol ul li, ol ol li {float: none; border-top: 1px solid #6b727c;
    border-bottom: 1px solid #575f6a; position: relative;}
  ul ul li a, ul ol li a, ol ul li a, ol ol li a {padding: 15px 40px; color: #fff;
    white-space: nowrap;}
  ul ul li a:hover, ul ol li a:hover, ol ul li a:hover, ol ol li a:hover { background: #4b545f;}
  ol ul ol {position: absolute; left: 100%; top:0;}
</style>
```



Reševanje konfliktov pri določitvah sloga

- ▶ do konfliktov pride, kadar obstaja več določil za isti slog za isti element
- ▶ razreševanje v primeru konflikta pri vrstični definiciji, na nivoju dokumenta in z zunanjimi slogovnimi polami
- ▶ konflikt se načeloma razrešuje glede na specifičnost pravila
- ▶ uporabnikova določila imajo prednost pred avtorjevimi če se pravila označijo z bolj pomembnimi z oznako `!important`
npr. `font-style: italic !important;`



Reševanje konfliktov pri določitvah sloga

- ▶ prednost pravil v naslednjem vrstnem redu (od najvišje k najnižji):
 - pomembne določitve uporabnika
 - pomembne določitve avtorja
 - navadne določitve avtorja
 - navadne določitve uporabnika
 - določitve brskalnika
- ▶ naslednji korak je določitev po specifičnosti:
 - selektorji z uporabo id-jev
 - selektorji z uporabo razredov in psevdo-razredov
 - kontekstualni selektorji (več imen tipov pomeni, da so bolj specifični)
 - univerzalni selektorji
- ▶ če so konflikti še vedno prisotni, obvelja zadnja vid(e)na določitev

Slogovne pole za različne medije

- ▶ slogovna pola, vezana na napravo, ki prikazuje stran (**all**, handheld, braille, **speech**, **print**, **screen**, tv, projection, ...)
- ▶ definiramo jo z grupiranjem oblikovnih stavkov v @media _naprava_ {...}

```
<!DOCTYPE HTML>
<html>
<head> <title>Test CSS medijev</title>
<meta charset="utf-8" />
<style type="text/css" >
  body { background-color: green; font-size: 15px; }
  @media screen and (min-width: 480px) { body { background-color: yellow; } }
  @media screen and (min-height: 300px) { body { font-size: 25px; } }
  @media print{ body{ background-color: white; font-size: 40px; } }
</style>
</head>
<body>
  <h1>Test CSS medijev</h1>
  <p>Sedaj pa imamo tukaj neko besedilo, ki se bo prikazalo v brskalniku.
    S spreminjanjem dimenzij okna, pa bomo lahko spreminjali velikost pisave.
  </p>
</body>
</html>
```



Slogovne pole za različne medije

- ▶ zaradi preglednosti je včasih dobro, da so slogovne pole ločene po različnih datotekah, na primer za različne vrste naprav in podobno
- ▶ slogovne pole je, poleg uporabe značke `<style>` možno uvoziti z direktivo `@import`
- ▶ direktive za uvoz morajo biti prve direktive v definiciji sloga za dokument

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head> <title>Test CSS medijev</title>
```

```
  <meta charset="utf-8" />
```

```
  <style type="text/css" >
```

```
    @import 'custom.css';
```

```
    @import url('widescreen.css') screen and (min-width: 480px);
```

```
    @import "highscreen.css" screen and (min-height: 300px);
```

```
    @import url("printing.css") print;
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>Test CSS medijev</h1>
```

```
  <p>Sedaj pa imamo tukaj neko besedilo, ki se bo prikazalo v brskalniku.
```

```
    S spreminjanjem dimenzij okna, pa bomo lahko spreminjali velikost pisave.
```

```
  </p>
```

```
</body>
```

```
</html>
```

body {background-color: green;
font-size: 15px; }

body { font-size: 25px; }

body { background-color: yellow; }

body { background-color: white;
font-size: 60px; }



Napredna oblikovanja

- ▶ težava: eksperimentalna in neenotna implementacija različnih brskalnikov
 - vsak ima svojo sintakso
 - vsake funkcionalnosti ne podpira vsak brskalnik
 - potrebno je testiranje za posamezen brskalnik

Vendor prefix	Browsers
-ms-	Internet Explorer
-moz-	Mozilla-based browsers, including Firefox
-o-	Opera and Opera Mobile
-webkit-	WebKit-based browsers, including Google Chrome, Safari (and Safari on the iPhone) and Android



Napredna oblikovanja

- ▶ možnosti:
 - senčenja besedila in objektov
 - zaobljeni vogali
 - gradienti
 - obrobe črk
 - več slik ozadja
 - odsevi
- ▶ z implementacijo posameznih naprednih oblikovanj v brskalnikih se predpone lahko izpustijo (ne, če želimo prikaz v starejših brskalnikih):

```
-moz-border-radius: 10px 5px 10px 5px;  
-webkit-border-radius: 10px 5px 10px 5px;  
OZ.  
-webkit-border-top-left-radius: 10px;  
-webkit-border-top-right-radius: 5px;  
-webkit-border-bottom-left-radius: 5px;  
-webkit-border-bottom-right-radius: 10px;  
.....
```

```
border-radius: 10px 5px 10px 5px;
```

- slikovne obrobe
- animacije
- prehodi
- uporaba javno dostopnih pisav
- uporaba stolpcev

