

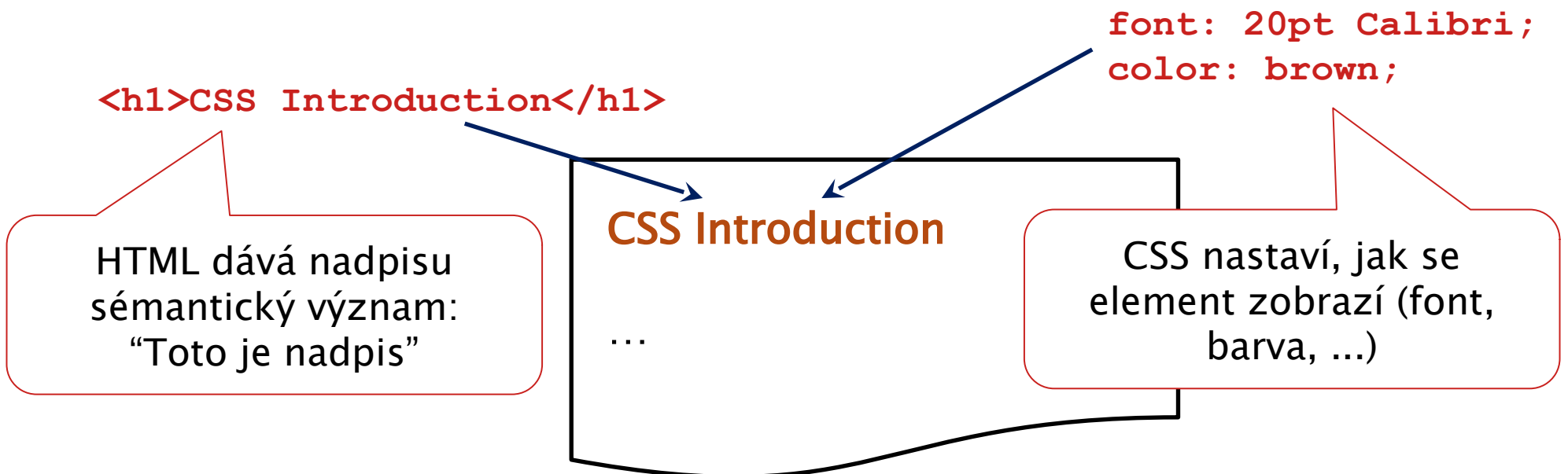


CSS

Klára Pešková, Klara.Peskova@mff.cuni.cz
Katedra softwaru a výuky informatiky, MFF UK
Základy tvorby webu, ZS 2023/24

Co je CSS?

- Cascading Style Sheets
- CSS popisuje, jak budou HTML prvky vypadat na obrazovce, telefonu, na papíře...
- + jedno CSS pro několik stránek



CSS syntax

- Obecný tvar pravidel:

```
selektor {  
    vlastnost: hodnota;  
    vlastnost2: hodnota2  
}
```

- Příklad:

```
p {  
    color: red;  
    text-align: center  
}
```

Tři způsoby vkládání do HTML

- Inline :

```
<p style="text-align: center;">Text</p>
```

- V hlavičce HTML (uvnitř elementu `<head>`), jako obsah tagu `<style>`

```
<head>
  <style>
    p {text-align: center;}
  </style>
</head>
```

- V samostatném .css souboru:

```
<head>
  <link rel="stylesheet"
        type="text/css"
        href="styles.css">
</head>
```

```
styles.css
p { text-align: center;}
body { color: blue; }
```

Druhy CSS selektorů

- **p** styly pro druh HTML elementu (pro všechny odstavce)

- **#mys** styly pro HTML prvek, s `id="mys"`

```
<h1 id="mys">Myš domácí</h1>
```

- **.upozorneni** styly pro prvky s přiřazenou CSS třídou

 - Jednomu HTML tagu může být přiřazeno více tříd

```
<p class="upozorneni cervena">Pozor, pes!  
</p>
```

- ***** univerzální selektor (všechno)

Používání selektorů

- Agregace
 - `s1, s2 {...styly...}` stejné styly pro více selektorů
- Kombinování selektorů
 - `p.aktualne` vybere všechny odstavce se třídou `aktualne`
 - Využití relativní pozice prvku v rámci stromové struktury HTML
 - `E F` vybere prvky `F`, které mají předka `E`
 - `E > F` vybere prvky `F`, které mají přímého rodiče `E`
 - `E + F` vybere prvky `F`, které jsou hned za `E`
 - `E ~ F` vybere prvky `F`, které jsou za `E`

CSS Selectors Example

`E F` vybere prvky `F`, které mají předka `E`
`E > F` vybere prvky `F`, které mají přímého rodiče `E`
`E + F` vybere prvky `F`, které jsou hned za `E`
`E ~ F` vybere prvky `F`, které jsou za `E`

```
div.info { ... }
```

```
<section>  
  <div class="info">...</div>  
  <p class="info">...</p>  
  <ul>  
    <li>first item</li>  
    <li>second item</li>  
    <li>third item</li>  
    <li>fourth item</li>  
  </ul>  
</section>  
<section class="small">  
  <section>  
    <p>Paragraph of smaller text</p>  
  </section>  
  <p>Another one of smaller text</p>  
</section>
```

CSS Selectors Example

`E F` vybere prvky `F`, které mají předka `E`
`E > F` vybere prvky `F`, které mají přímého rodiče `E`
`E + F` vybere prvky `F`, které jsou hned za `E`
`E ~ F` vybere prvky `F`, které jsou za `E`

```
div.info { ... }
```



```
<section>
  <div class="info">...</div>
  <p class="info">...</p>
  <ul>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
    <li>fourth item</li>
  </ul>
</section>
<section class="small">
  <section>
    <p>Paragraph of smaller text</p>
  </section>
  <p>Another one of smaller text</p>
</section>
```


CSS Selectors Example

`E F` vybere prvky `F`, které mají předka `E`
`E > F` vybere prvky `F`, které mají přímého rodiče `E`
`E + F` vybere prvky `F`, které jsou hned za `E`
`E ~ F` vybere prvky `F`, které jsou za `E`

```
div.info { ... }
```

```
p.info { ... }
```

```
<section>
```

```
  <div class="info">...</div>
```

```
    <p class="info">...</p>
```

```
    <ul>
```

```
      <li>first item</li>
```

```
      <li>second item</li>
```

```
      <li>third item</li>
```

```
      <li>fourth item</li>
```

```
    </ul>
```

```
</section>
```

```
<section class="small">
```

```
  <section>
```

```
    <p>Paragraph of smaller text</p>
```

```
  </section>
```

```
  <p>Another one of smaller text</p>
```

```
</section>
```

CSS Selectors Example

`E F` vybere prvky `F`, které mají předka `E`
`E > F` vybere prvky `F`, které mají přímého rodiče `E`
`E + F` vybere prvky `F`, které jsou hned za `E`
`E ~ F` vybere prvky `F`, které jsou za `E`

```
div.info { ... }
p.info { ... }
```

```
<section>
  <div class="info">...</div>
  <p class="info">...</p>
  <ul>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
    <li>fourth item</li>
  </ul>
</section>
<section class="small">
  <section>
    <p>Paragraph of smaller text</p>
  </section>
  <p>Another one of smaller text</p>
</section>
```

CSS Selectors Example

`E F` vybere prvky `F`, které mají předka `E`
`E > F` vybere prvky `F`, které mají přímého rodiče `E`
`E + F` vybere prvky `F`, které jsou hned za `E`
`E ~ F` vybere prvky `F`, které jsou za `E`

```
div.info { ... }
p.info { ... }
li+li { ... }
```

```
<section>
  <div class="info">...</div>
  <p class="info">...</p>
  <ul>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
    <li>fourth item</li>
  </ul>
</section>
<section class="small">
  <section>
    <p>Paragraph of smaller text</p>
  </section>
  <p>Another one of smaller text</p>
</section>
```

CSS Selectors Example

E **F** vybere prvky **F**, které mají předka **E**
E **>** **F** vybere prvky **F**, které mají přímého rodiče **E**
E **+** **F** vybere prvky **F**, které jsou hned za **E**
E **~** **F** vybere prvky **F**, které jsou za **E**

```
div.info { ... }
p.info { ... }
li+li { ... }
```

```
<section>
  <div class="info">...</div>
  <p class="info">...</p>
  <ul>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
    <li>fourth item</li>
  </ul>
</section>
<section class="small">
  <section>
    <p>Paragraph of smaller text</p>
  </section>
  <p>Another one of smaller text</p>
</section>
```

CSS Selectors Example

E F vybere prvky F , které mají předka E
 $E > F$ vybere prvky F , které mají přímého rodiče E
 $E + F$ vybere prvky F , které jsou hned za E
 $E \sim F$ vybere prvky F , které jsou za E

```
div.info { ... }
p.info { ... }
li+li { ... }
li+li+li { ... }
```

```
<section>
  <div class="info">...</div>
  <p class="info">...</p>
  <ul>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
    <li>fourth item</li>
  </ul>
</section>
<section class="small">
  <section>
    <p>Paragraph of smaller text</p>
  </section>
  <p>Another one of smaller text</p>
</section>
```

CSS Selectors Example

E **F** vybere prvky **F**, které mají předka **E**
E **>** **F** vybere prvky **F**, které mají přímého rodiče **E**
E **+** **F** vybere prvky **F**, které jsou hned za **E**
E **~** **F** vybere prvky **F**, které jsou za **E**

```
div.info { ... }
p.info { ... }
li+li { ... }
li+li+li { ... }
```

```
<section>
  <div class="info">...</div>
  <p class="info">...</p>
  <ul>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
    <li>fourth item</li>
  </ul>
</section>
<section class="small">
  <section>
    <p>Paragraph of smaller text</p>
  </section>
  <p>Another one of smaller text</p>
</section>
```

CSS Selectors Example

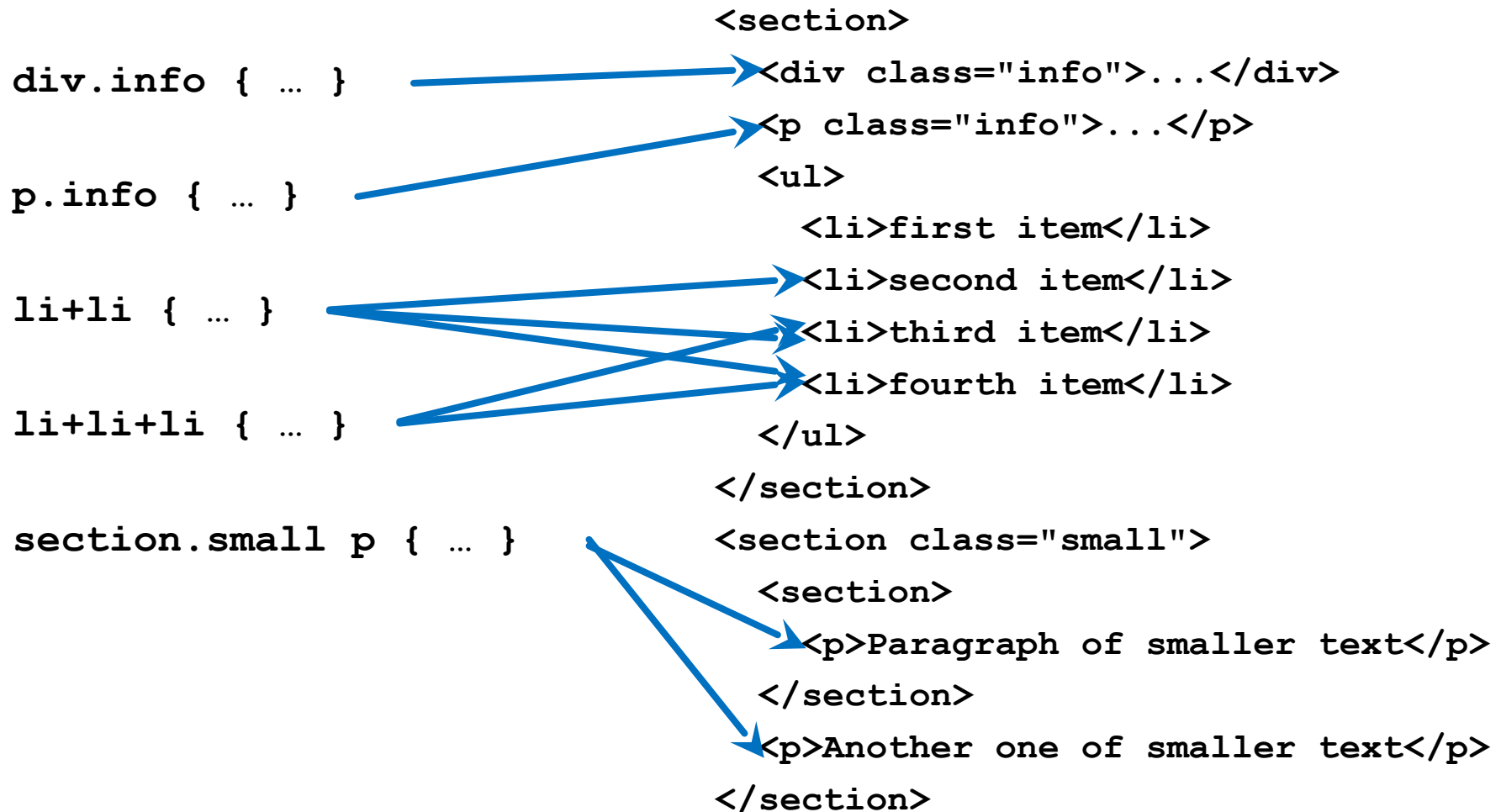
E **F** vybere prvky **F**, které mají předka **E**
E **>** **F** vybere prvky **F**, které mají přímého rodiče **E**
E **+** **F** vybere prvky **F**, které jsou hned za **E**
E **~** **F** vybere prvky **F**, které jsou za **E**

```
div.info { ... }
p.info { ... }
li+li { ... }
li+li+li { ... }
section.small p { ... }
```

```
<section>
  <div class="info">...</div>
  <p class="info">...</p>
  <ul>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
    <li>fourth item</li>
  </ul>
</section>
<section class="small">
  <section>
    <p>Paragraph of smaller text</p>
  </section>
  <p>Another one of smaller text</p>
</section>
```

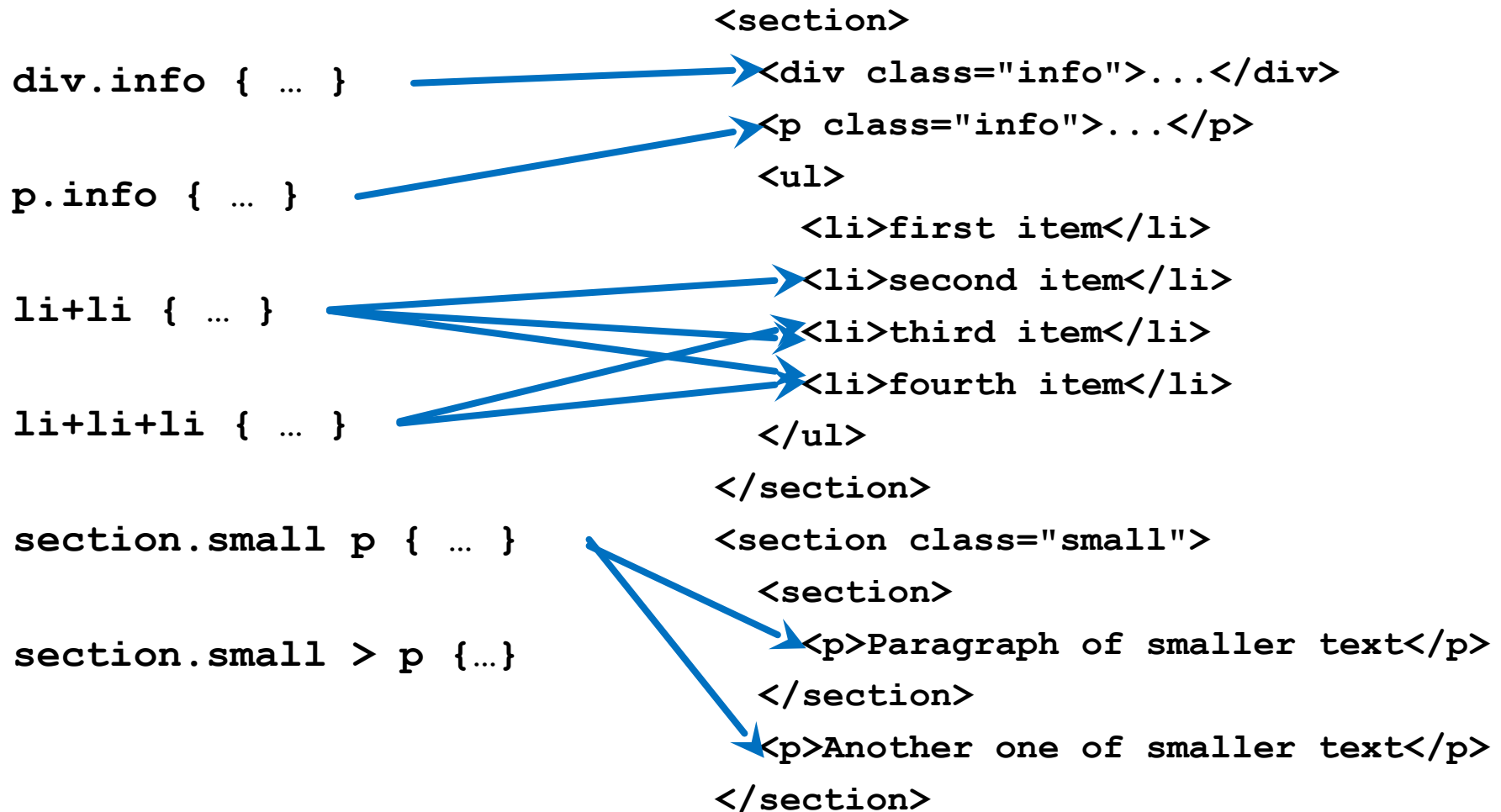
CSS Selectors Example

E F vybere prvky F , které mají předka E
 E $>$ F vybere prvky F , které mají přímého rodiče E
 E $+$ F vybere prvky F , které jsou hned za E
 E \sim F vybere prvky F , které jsou za E



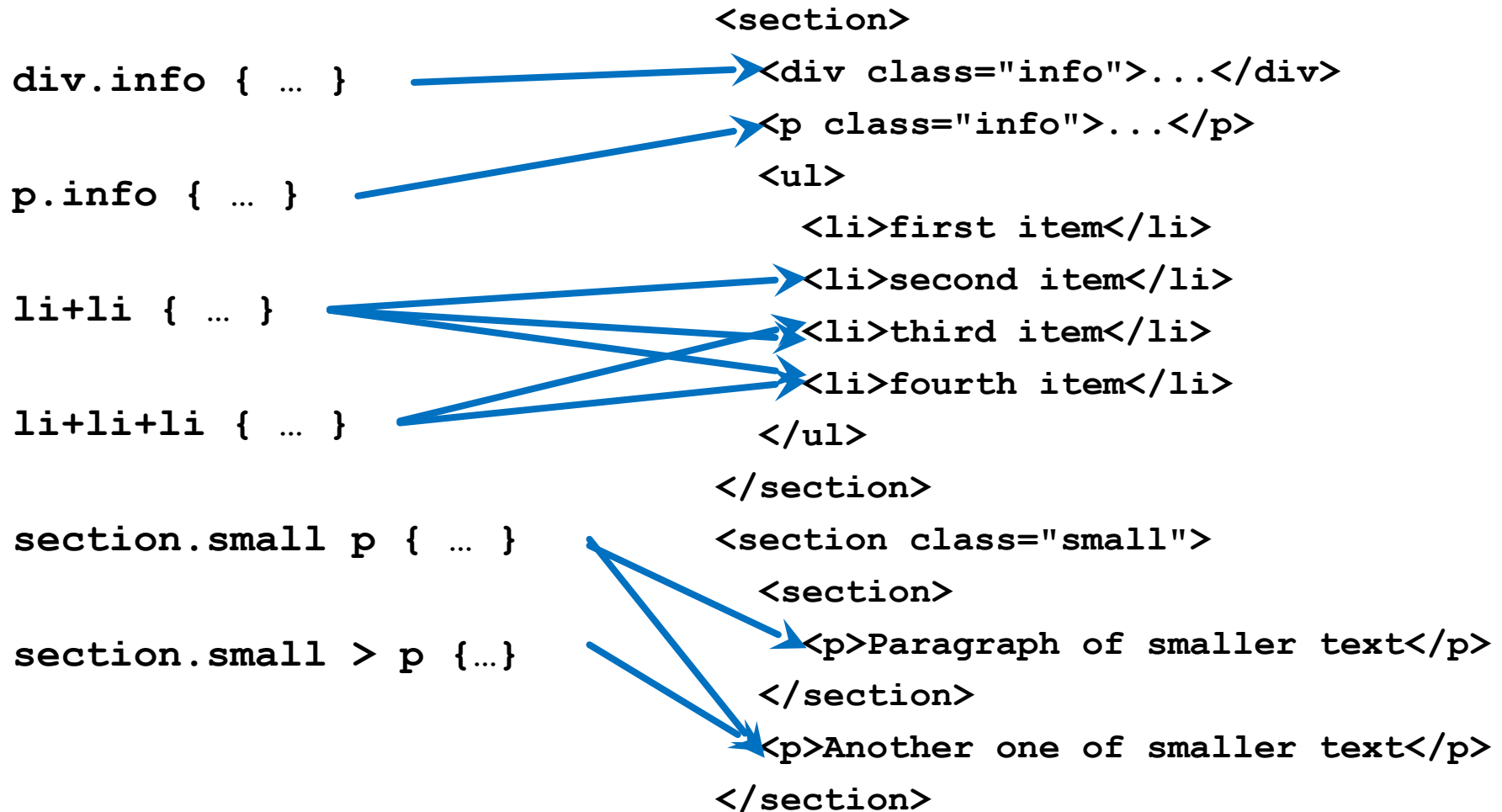
CSS Selectors Example

E F vybere prvky F , které mají předka E
 $E > F$ vybere prvky F , které mají přímého rodiče E
 $E + F$ vybere prvky F , které jsou hned za E
 $E \sim F$ vybere prvky F , které jsou za E



CSS Selectors Example

E **F** vybere prvky **F**, které mají předka **E**
E **>** **F** vybere prvky **F**, které mají přímého rodiče **E**
E **+** **F** vybere prvky **F**, které jsou hned za **E**
E **~** **F** vybere prvky **F**, které jsou za **E**



Kombinování selektorů - pozor!

- `ul li` – funguje i pro vnořené seznamy, které opět mají `li`

```
<ul>
  <li>
    <ol>
      <li> styly se aplikují
        i na toto li
      </li>
    </ol>
  </li>
</ul>
```

- `p.aktualne` VS. `p .aktualne` – pozor na mezery!
- `div ul, ol` – `div` patří pouze k `ul`

CSS pseudotřídy

- Odkazy

`a:link` normální odkaz

`a:visited` navštívený

`a:hover` při přejetí myší

`a:active` právě kliknutý

`:nth-child(even)`

`:first-child` (pokud je daný prvek první potomek)

`:last-child, :only-child`

- př. první `<i>` ve všech `<p>`

`p i:first-child`

- př. všechna `<i>` v prvním `<p>`

`p:first-child i`

Pseudotřídy – příklad

Jméno	Příjmení	Body
Petr	Smutný	100
Ludvík	Veselý	150
Jiří	Černý	300
Klement	Nový	250

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

CSS pseudo elementy

- Styly pro části HTML prvků

```
::first-letter
```

```
::first-line
```

```
::selection
```

```
::after
```

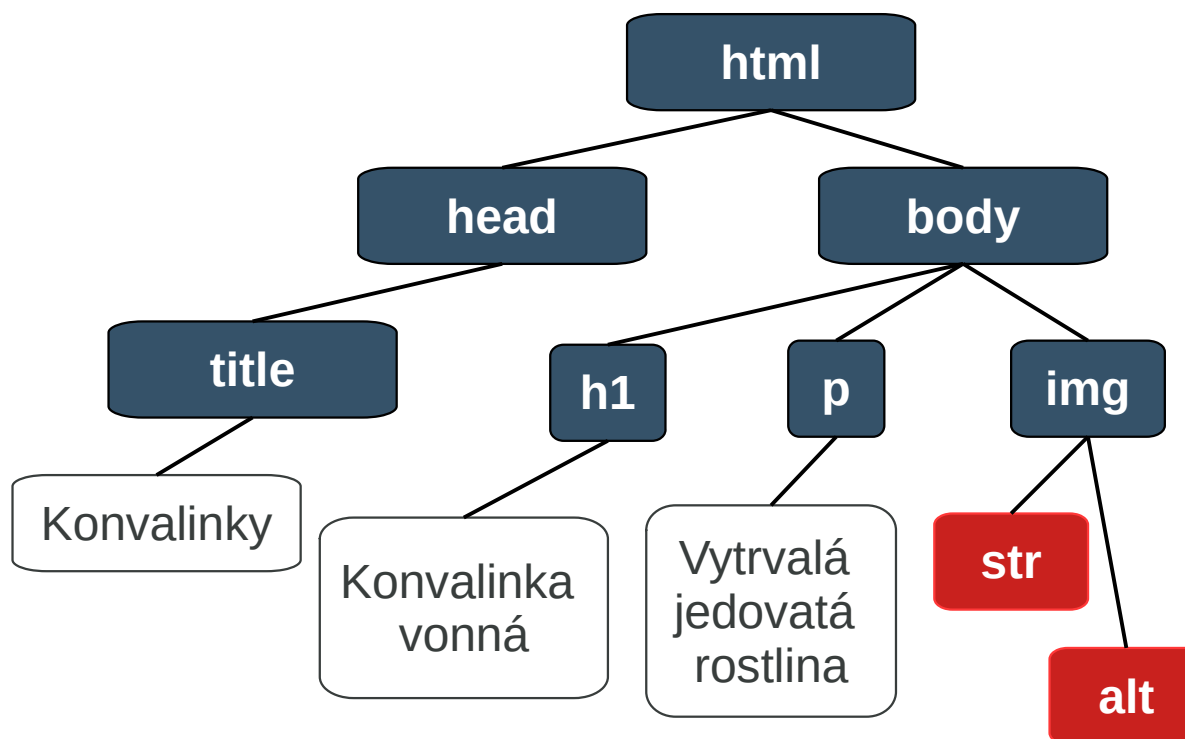
```
::before
```

```
h1::after {content: url(smiley.gif) }
```

```
p::first-letter {font-size: xx-large; }
```

Dědičnost

- Některé vlastnosti se dědí od nadřazených prvků v HTML stromě, např. font (`<body>` → `<h1>`, `<p>`...)



Kaskády

- Pro jeden HTML prvek může platit více stylů
- Poměrně složitá pravidla, v jakém pořadí se styly se aplikují
 - 1)vložené styly
 - 2)ID
 - 3)třídy
 - 4)elementy, pseudo elementy
 - 5)*
- Přednost:

color: blue !important;

Kaskády - podrobněji

- Váha je určena (podrobněji na dalším slidu):
 1. Podle toho, kde jsou styly zapsané – záleží na “blízkosti” definice stylu HTML elementu (např. inline styly mají větší prioritu než styly v externím souboru)
 2. Podle specifacity selektorů
 3. Podle pořadí, ve kterém jsou styly zapsány (předchozí přepíše následující)

Kaskády - ještě podrobněji

- Specificita selektorů
 - Definuje pořadí – prioritu selektorů
 - Výpočet:
 - Připočtete 1000 za deklaraci v atributu style
 - 100 za ID
 - 10 za třídu nebo pseudo-třídu
 - 1 za název HTML elementu nebo pseudo-elementu
 - Zjednodušeně: atribut style > ID > třídy > elementy
 - Tedy např. `a: hover` má vyšší prioritu než samotné `a`
 - Pokud je specificita stejná, vítězí naposledy definovaný (přepíše předchozí)

CSS variables

- funkce `var()`
- globální (v celém dokumentu)

```
:root {  
  --blue: #1e90ff;  
}  
  
.container {  
  color: var(--blue);  
}
```

- nebo lokální (uvnitř selektoru)

```
div {  
  --color: red;  
  color: var(--color);  
}
```

CSS variables - pokračování

- proměnné je možné nastavovat pomocí javascriptu v průběhu existence stránky

```
document.documentElement.style.setProperty('--varName', 'propValue')
```

- **podpora:** <https://caniuse.com/css-variables>

```
@supports(--color: red) {  
    // code here implementing variables  
}
```

- **záložní hodnota**

```
div {  
    --color: red;  
    color: red;  
    color: var(--color);  
}
```



CSS vlastnosti

Stylování textu – fonty

- Typ fontu: `font-family: Arial CE, sans-serif`
- Velikost: `font-size: 12px, 12pt, 1.2em` (relativní k velikosti textu aktuálního prvku)

`font-style: normal, italic`

`font-weight: normal, bold`

- Zkrácený zápis:

`font: italic bold 20px Arial`

Stylování textu

- Zarovnání textu

```
text-align: left, right, center, justify
```

- Podtržení atd.

```
text-decoration:  
  none,  
  underline,  
  line-through
```

Barvy

- `color` – barva popředí (písmo)

```
Red  
Tomato  
MediumSeaGreen,  
#161616,  
rgb(0, 0, 100)
```

- **Průhlednost** `opacity` nebo `rgba`

```
opacity: 0-1  
rgba(255, 0, 0, 0.5)
```


Pozadí

- `background-color` – barevná výplň pozadí
- Obrázek na pozadí

```
background-image: url("obrazek.gif");  
background-repeat: repeat-x, no-repeat  
background-position: right top;  
background-attachment: fixed;
```

- Nastavení pozadí jednou vlastností

```
background: #ffffff url("tree.png") no-repeat right top
```

- Barevný přechod

```
background: linear-gradient(to bottom right, red, yellow)
```

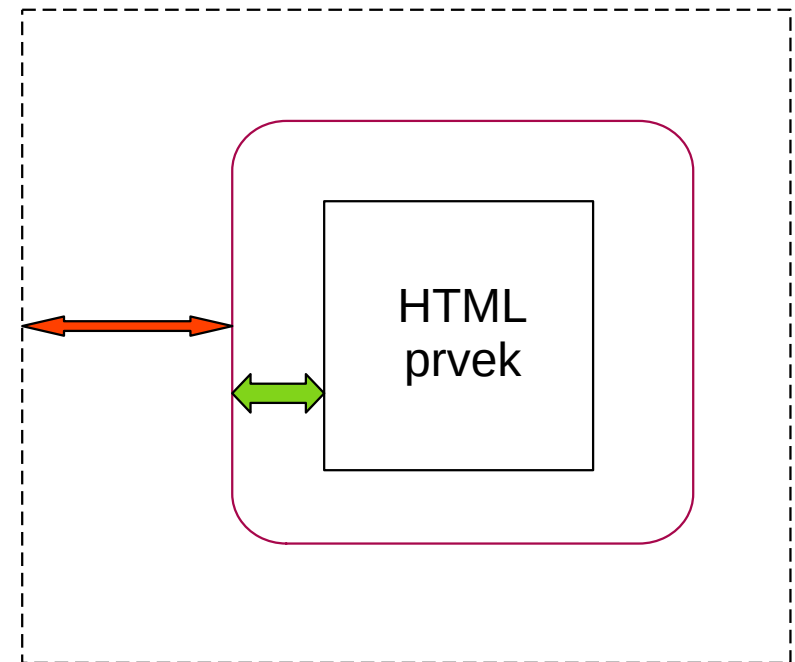
Okraje

- **Padding**
 - 10px (jedna hodnota) – všechny strany
 - 10px 0px (dvě hodnoty) – nahoře-dole, vlevo-vpravo
 - 0px 5px 10px 5px (čtyři hodnoty) – nahoře, vpravo, dole, vlevo

- **Margin**

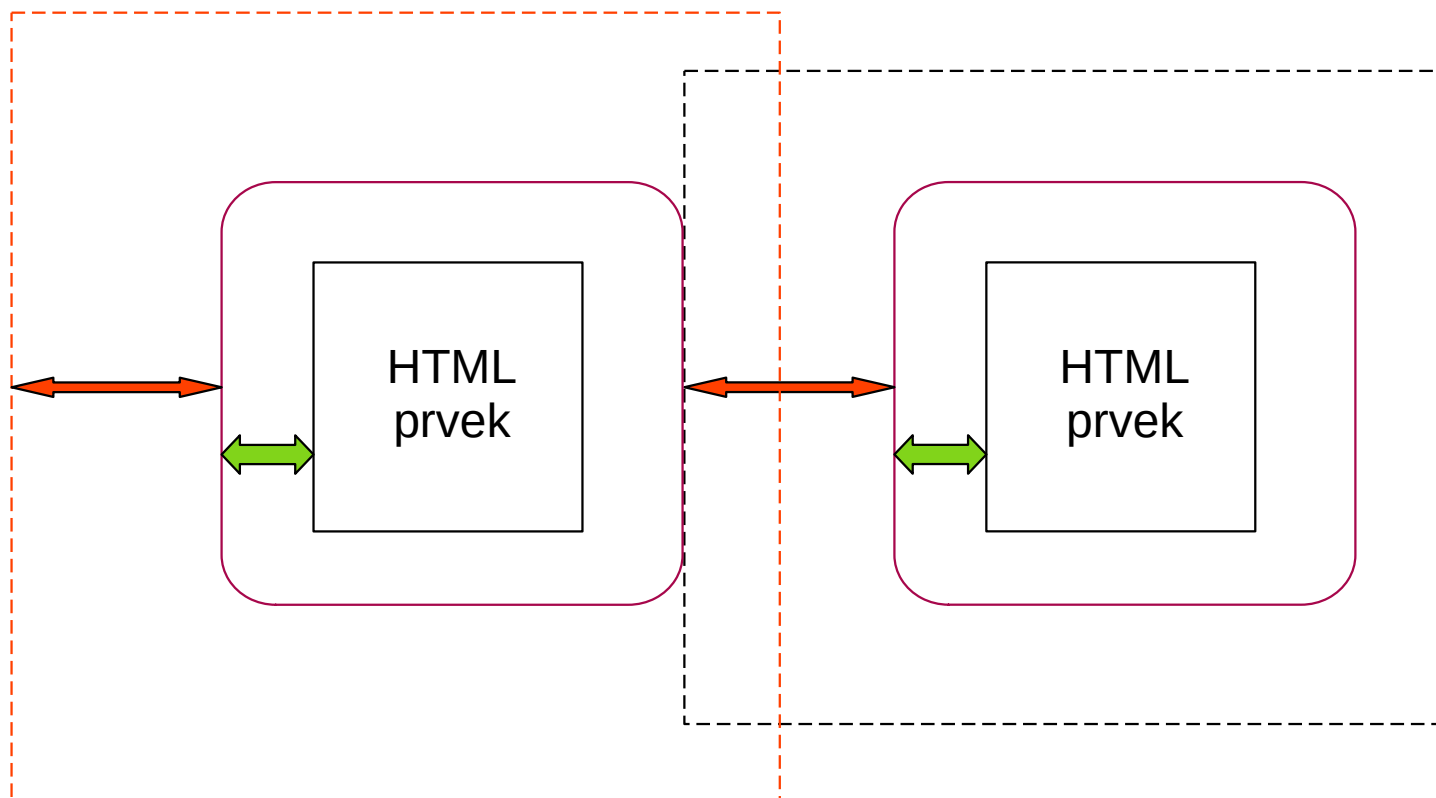
- **Border**

`border-radius:10px`



Okraje

- Margin dvou prvků se může překrývat



Border

- `border: 2px solid blue;`

- `border-style:`

`dotted` – tečkovaně

`dashed` – čárkovaně

`solid` – normální čára

`double` – dvojité

`groove` – 3D vyrytě

`ridge` – 3D vystouple

`inset` – zamáčknutě

`outset` – vystouple

`none` – bez rámečku

`hidden` – skrytý rámeček

- Smíšený rámeček

`border-style: dotted dashed solid double;`

- Tabulka bez rámečku

`border-collapse: collapse;`

Stíny

- text-shadow

```
text-shadow: v_posun h_posun r_rozmazání barva  
color: white; text-shadow: 2px 2px 4px #000000;
```

- box-shadow

```
box-shadow: v_posun h_posun r_rozmazání  
            r_rozšíření barva  
box-shadow: 3px 10px 10px 5px #555
```

Text se stínem



Transformace

- **2D transformace:**

```
translate(x px, y px)
```

```
rotate(20deg)
```

```
scale(2, 3) – 2x širší, 3x delší
```

```
skewX(20deg), skewY(20deg) – zmáčknutí podle osy x, y
```

```
skew(X, Y)
```

```
matrix(scaleX(), skewY(), skewX(), scaleY(),  
        translateX(), translateY())
```

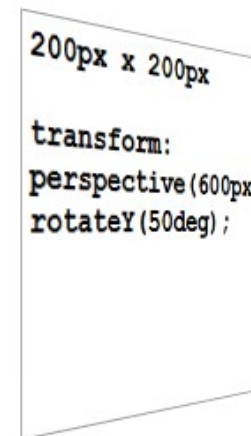
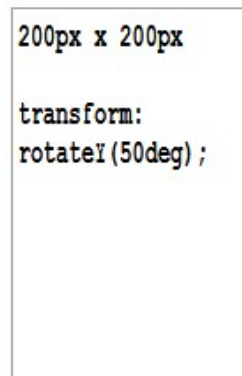
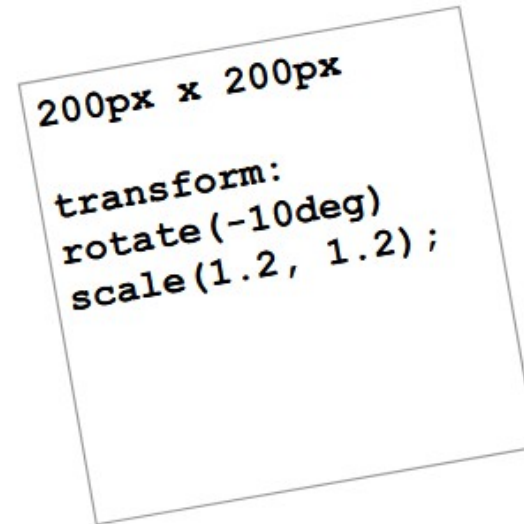
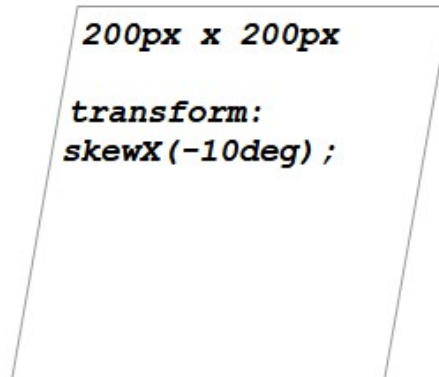
- **3D transformace:**

```
rotateX(90deg)
```

```
rotateY(90deg)
```

```
rotateZ(90deg)
```

Příklady transformací



Tranzice

- jeden styl se postupně mění v jiný
- je možné měnit více vlastností

`transition: width 2s` – jaká vlastnost se mění, trvání

`transition: width 2s, height 3s`

`transition-timing-function:`

`ease` – default

`linear`

`ease-in`

`ease-out`

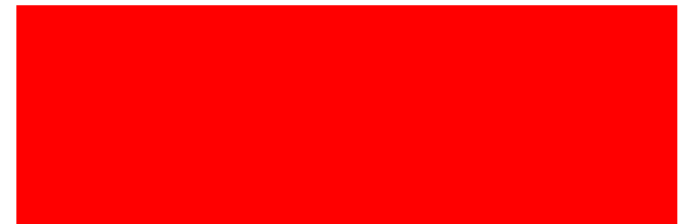
`transition-delay: 1s`

`transition: width 2s linear 1s`

Tranzice - příklad 1

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
}
```

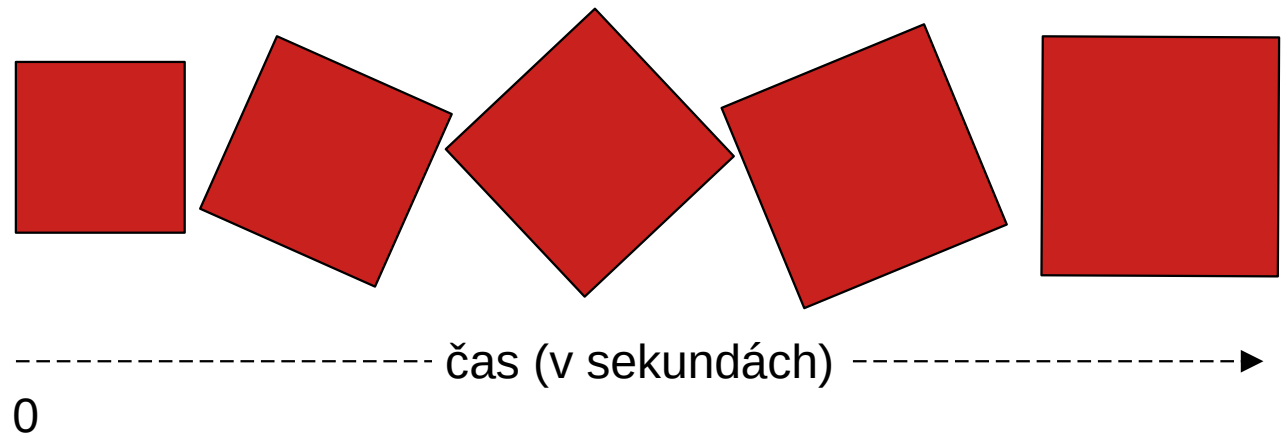
```
div:hover {  
  width: 300px;  
}
```



Tranzice – příklad 2

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s, height 2s, transform 2s;  
}
```

```
div:hover {  
  width: 300px;  
  height: 300px;  
  transform:  
  rotate(90deg);  
}
```



CSS Animace

- je potřeba
 - provázat animaci s HTML elementem
 - nastavit trvání animace (jinak trvá 0 s)
- @keyframes - definují styl prvků v konkrétním čase

Animace - příklad 1

- přebarvující se čtvereček

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

```
@keyframes example {  
  0%   {background-color: red;}  
  25%  {background-color: yellow;}  
  50%  {background-color: blue;}  
  100% {background-color: green;}  
}
```

https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation2

Animace - příklad 2

```
.tecky {  
    width: 5px;  
    height: 20px;  
    border-bottom: 5px dotted;  
    animation-name: tecky;  
    animation-duration: 8s;  
    animation-iteration-count: infinite;  
}  
  
@keyframes tecky {  
    0%    {width: 5px;}  
    50%   {width: 200px;}  
    100%  {width: 5px;}  
}
```

@media

- All, screen, print, speech

- @media

```
@media print { styly pro tisk }
```

```
@media screen { styly pro zobrazení na  
obrazovce }
```

Obecné – co všechno může být hodnota

- Numerické hodnoty (velikost, úhel, trvání)
 - `font-size: 12pt;`
- Barva
 - `background-color: #00ff00;`
- Odkaz na externí zdroj (např. obrázek)
 - `background-image: url("paper-texture.png")`
- Znakový řetězec
 - `font-family: "Courier New";`
- Hodnota z výčtu
 - `border-style: solid;`

Obecné – jednotky

- Všechny číselné hodnoty kromě 0 musí mít zadanou jednotku

cm, mm, in	Centimeters, Millimeters, Inches (1 in = 2.54cm)
px	Pixels (1 px = 1 / 96 in)
pt	Typographical points (1 pt = 1 / 72 in)
pc	Picas (1 pc = 12pt)
em	Relative to the font-size of current element
ex	Relative to the height of 'x' in current font size
%	Special – relative to some existing/inherited value
vh, vw	Relative to 1% of width/height of the viewport
deg	Degrees (rotation)
s	Seconds

Obecné - zkrácený zápis vlastností

- Mnoho CSS vlastností je možné zapsat v “úsporné” podobě
- Např. tři vlastnosti rámečku, je možné nastavit buď každou zvlášť:

```
border-width: 2px;
```

```
border-style: solid;
```

```
border-color: blue;
```

- Nebo pomocí jedné vlastnosti:

```
border: 2px solid blue;
```