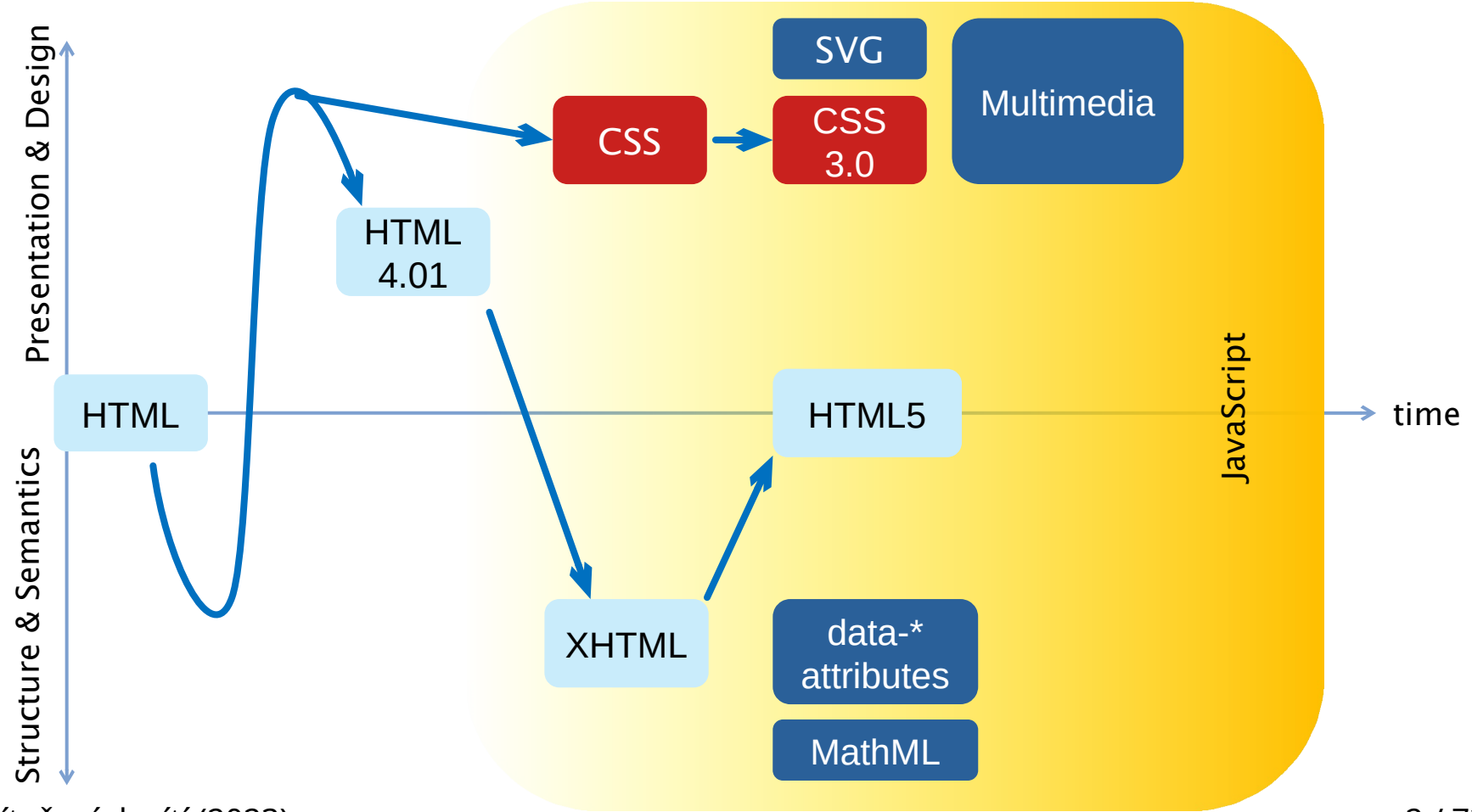
Cascading Style Sheet – kaskádové styly

Úvod od počítačových sítí

Mgr. Klára Pešková, Ph.D., Klara.Peskova@mff.cuni.cz
Katedra softwaru a výuky informatiky

Vývoj webových prezentací



Vývoj CSS

- CSS 1 (1996)
  - Základní styly pro text – fonty, zarovnání, …
  - Okraje (padding, margin), rámečky
  - Barva textu a pozadí
- CSS 2 (1998)
  - Nové typy pozicování
  - Koncept @media
- CSS 2.1 (2004–2011)
  - Řešení různých problémů CSS 2

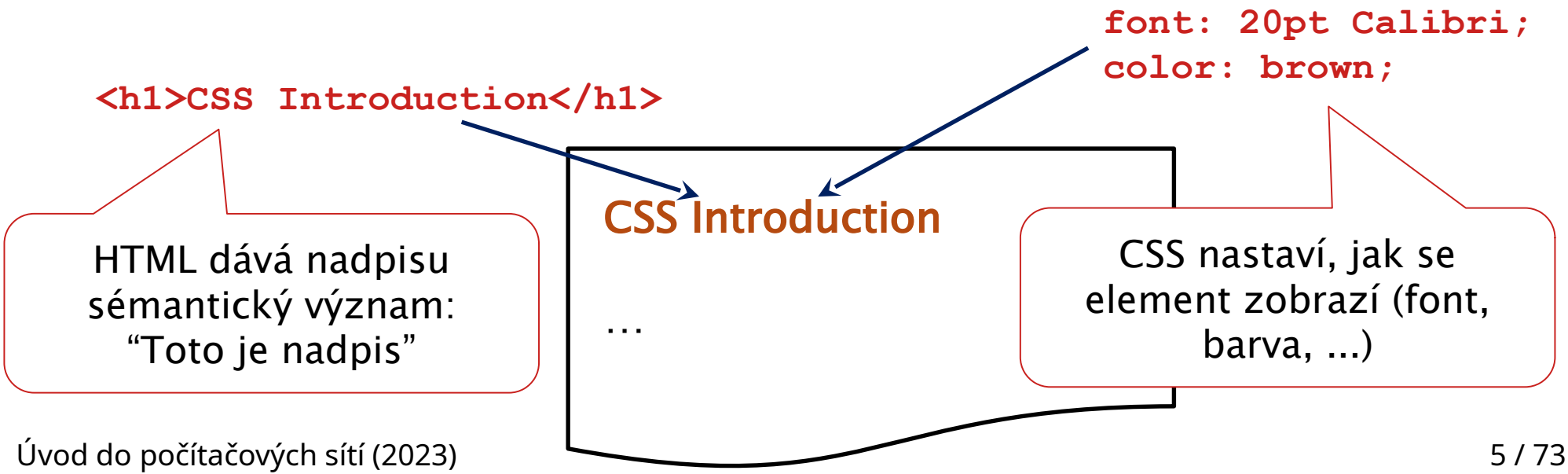
Vývoj CSS

- CSS 3 (1999–současnost)
  - Vylepšení stávajících vlastností – složitější pozadí, vylepšené rámečky, …
  - Další moderní efekty - kulaté rohy, stíny
  - Povoluje vlastní fonty
  - Přidány tranzice a animace

Co je CSS?

- 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:

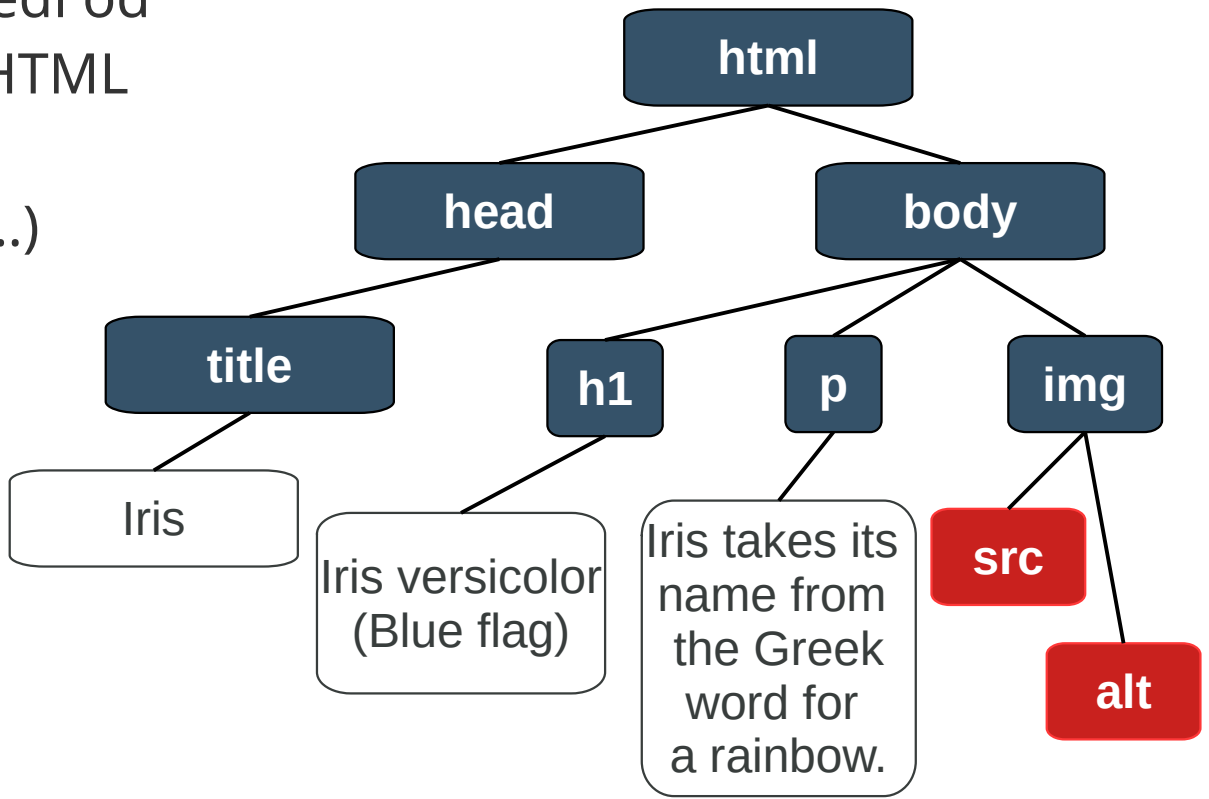
```
<head>
   <link rel="stylesheet"
         type="text/css"
         href="styles.css">
</head>
```

styles.css

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

body { color: blue; }
```

Dědičnost

- Některé vlastnosti, např. font, nebo barva textu se dědí od nadřazených prvků v HTML stromě (`<body>` → `<h1>`, `<p>`...)
- Vlastnosti, které se dědí mají nastavenou defaultní hodnotu `inherit`



CSS selektory

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`
  - `h1#hlavni` vybere `<h1 id="hlavni">`
  - 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`

Kombinování selektorů – příklad

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>
```

Kombinování selektorů – příklad

```
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>
```

Kombinování selektorů – příklad

```
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>
```

Kombinování selektorů – příklad

```
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>
```

Kombinování selektorů – příklad

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>
```

Kombinování selektorů – příklad

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>
```

Kombinování selektorů – příklad

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

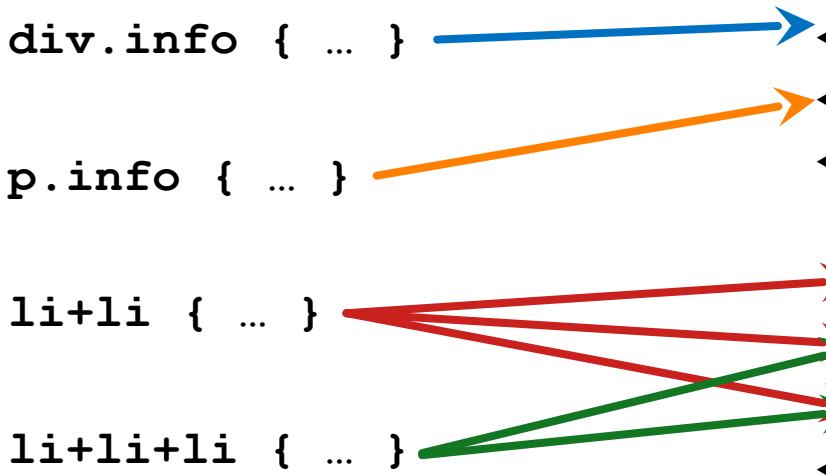
```
div.info { … }

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>
```

Kombinování selektorů – příklad

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

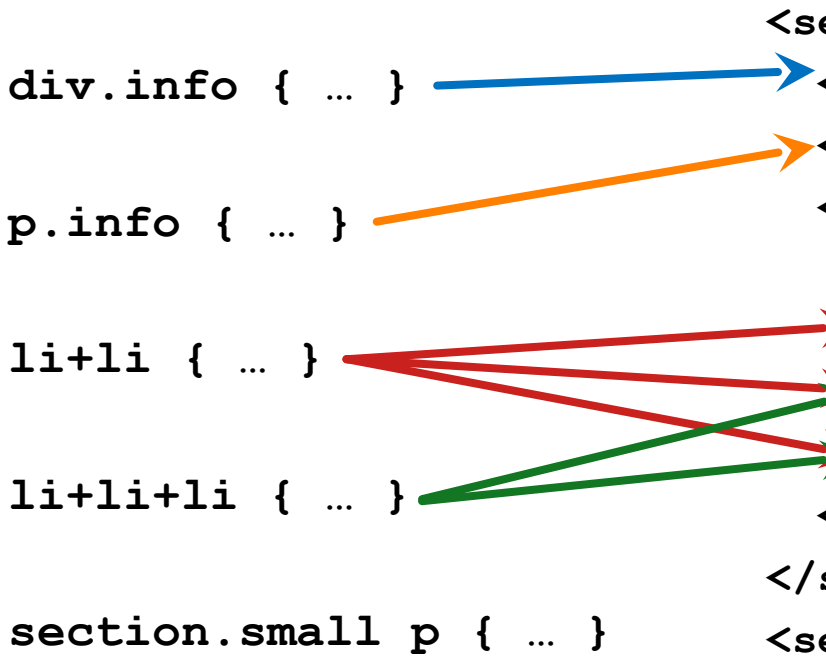
```
div.info { … }

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>
```

Kombinování selektorů – příklad

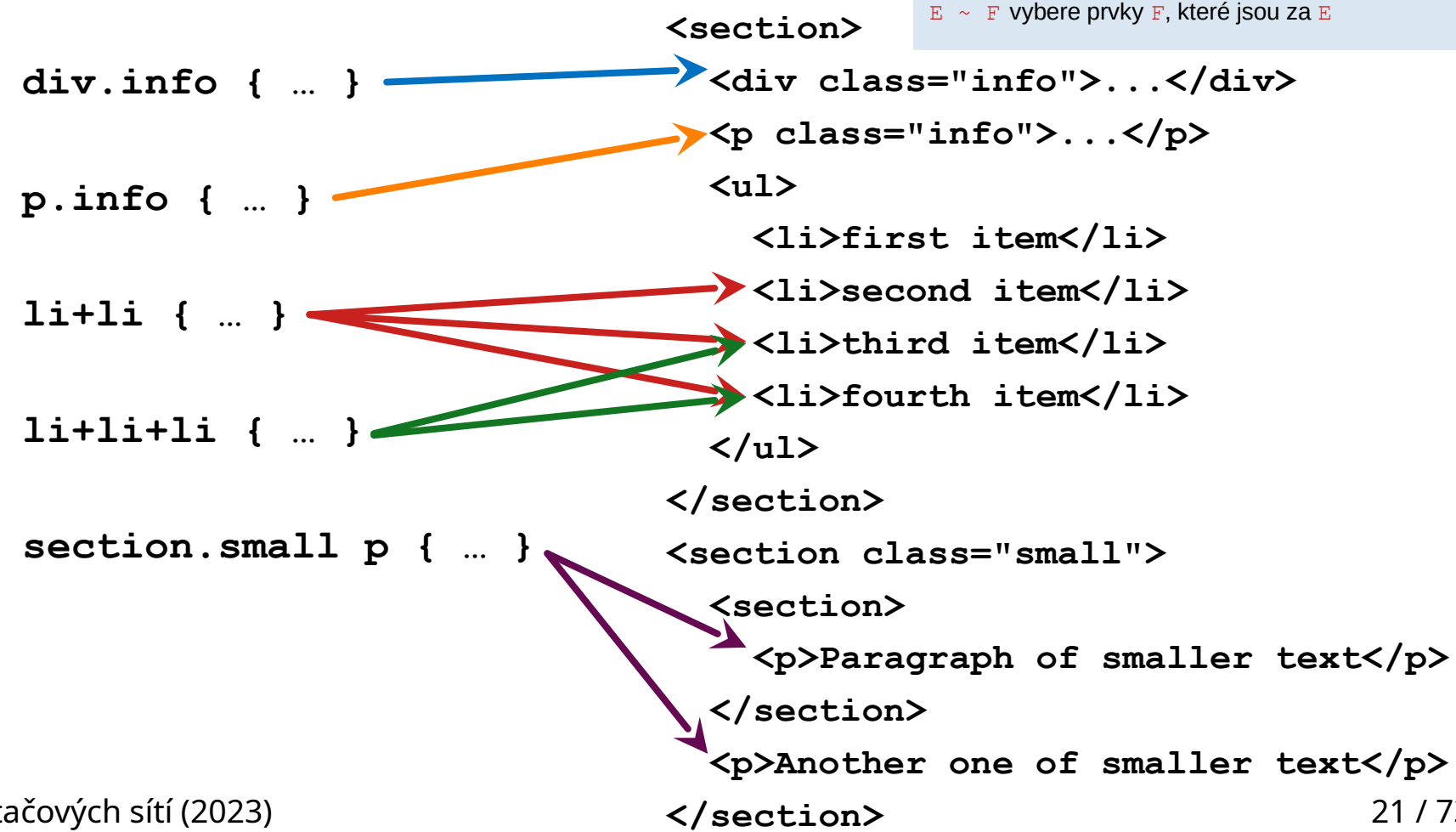
`E`	`F`	vybere prvky `F`, které mají předka `E`
`E`	`> F`	vybere prvky `F`, které mají přímého rodiče `E`
`E`	`+ F`	vybere prvky `F`, které jsou hned za `E`
`E`	`~ F`	vybere prvky `F`, které jsou za `E`




```
<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>
```

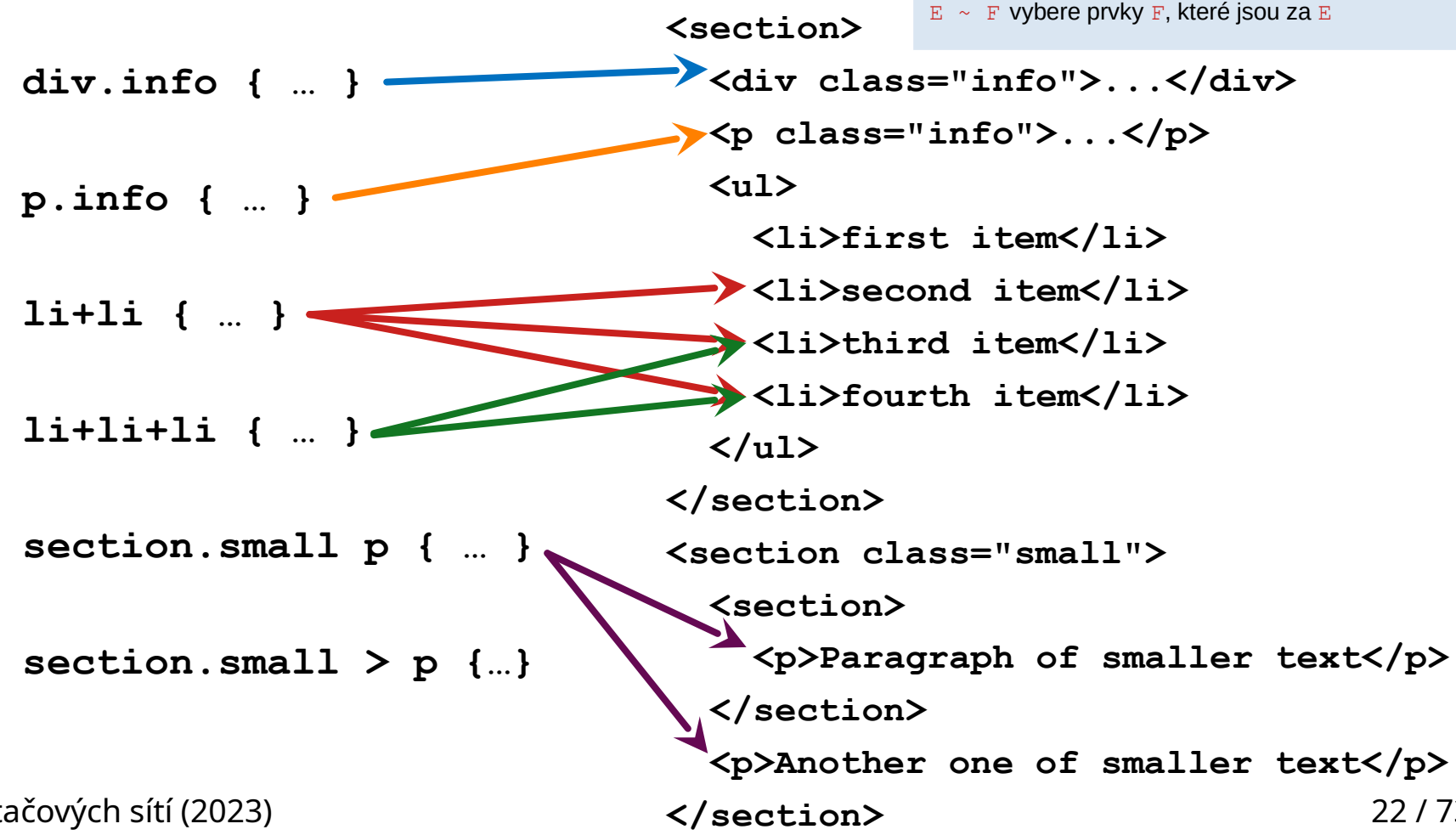
Kombinování selektorů – příklad

`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 – příklad

`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`

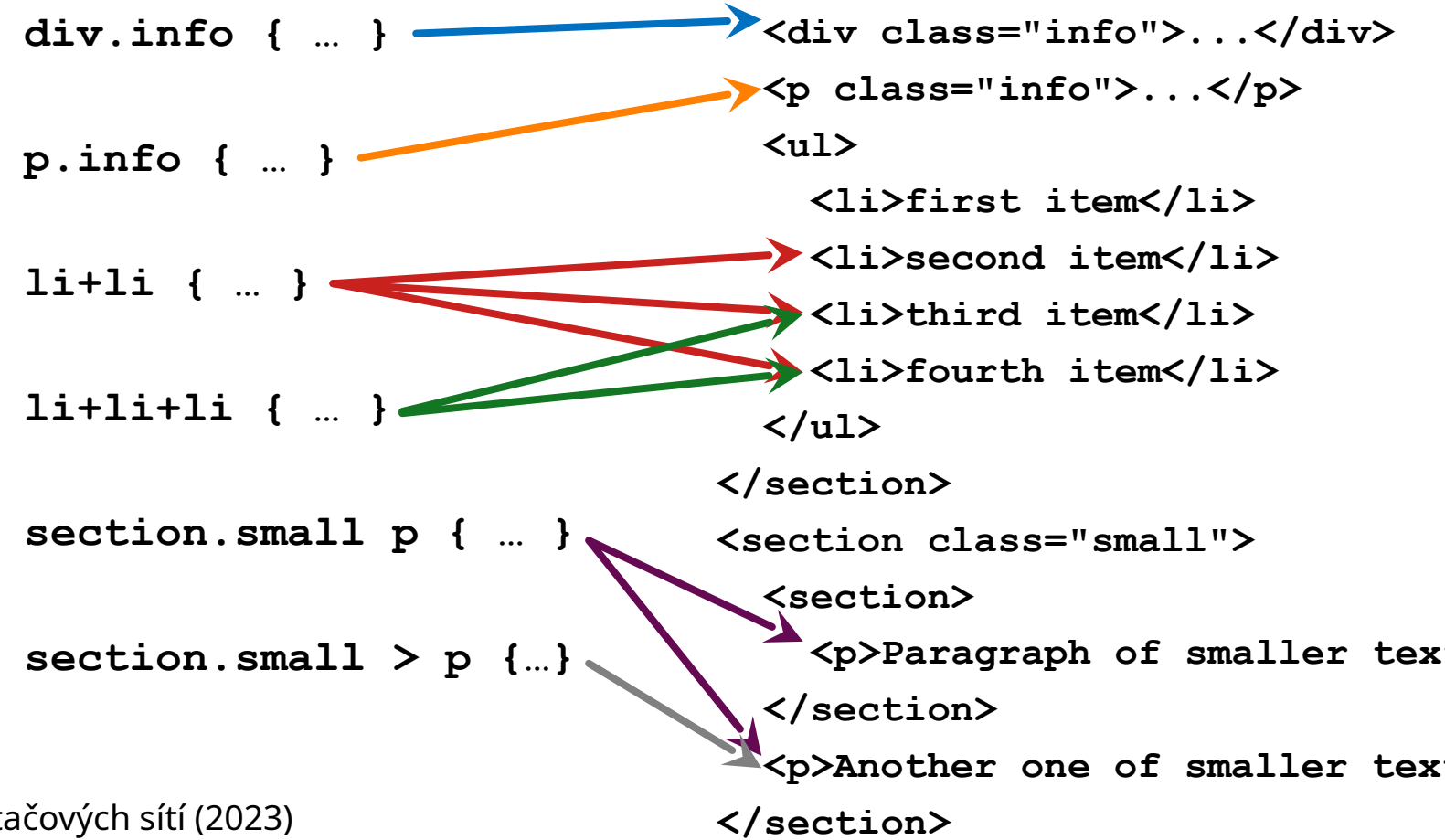


Kombinování selektorů – příklad

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.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>
```



Kombinování selektorů

- `ul li`
  - `li` kdekoliv uvnitř `ul`
- `p.info` vs. `p .info`
  - Mezera má význam!
- `main ul, ol`
  - main patří jen k ul

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

Pseudotřídy

- Odkazy

`a:link`	normální odkaz
`a:visited`	navštívený
`a:hover`	při přejetí myší
`a:active`	právě kliknutý

- `:first-child` (`:first-of-type`) – styl se aplikuje, pokud je daný prvek první potomek / první potomek daného typu
- `:last-child` (`:last-of-type`), `:only-child` (`:only-of-type`)
- `:nth-child(even)` / `:nth-of-type(3n+1)`
  - př. první `<i>` ve všech `<p>` — `p i:first-child`
  - př. všechna `<i>` v prvním `<p>` — `p:first-child i`

Pseudotřídy – příklad

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

Jméno	Příjmení	Body
Petr	Smutný	100
Ludvík	Veselý	150
Jiří	Černý	67
Klement	Nový	250

Pseudo elementy

- Styly pro části HTML prvků

```
::first-letter
::first-line
::selection
::after
::before
```

```
h1::after {content: url(smiley.gif)}
p::first-letter {color: #f00; font-size: xx-large;}
```

Kaskády

- Pro jeden HTML prvek může platit více stylů
- Poměrně složitá pravidla, v jakém pořadí se styly se aplikují
- 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 specificity selektorů
  3. Podle pořadí, ve kterém jsou styly zapsány (předchozí přepíše následující)
- Absolutní přednost `color: blue !important;`

Specificita selektorů

- Definuje pořadí – prioritu selektorů
- Výpočet:
  - Připočtěte 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 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 - další

- Zarovnání textu

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

- Podtržení atd.

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

- Text a prostor (mezi, kolem)

  `text-indent, line-height, letter-spacing, word-spacing`

Barvy

- barvu je možné nastavit různými způsoby:

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

- Průhlednost `opacity` nebo `rgba`

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

- `color` – barva popředí (např. písmo)

Nastavení pozadí

- `background-color` – barvená výplň pozadí
- Obrázky na pozadí

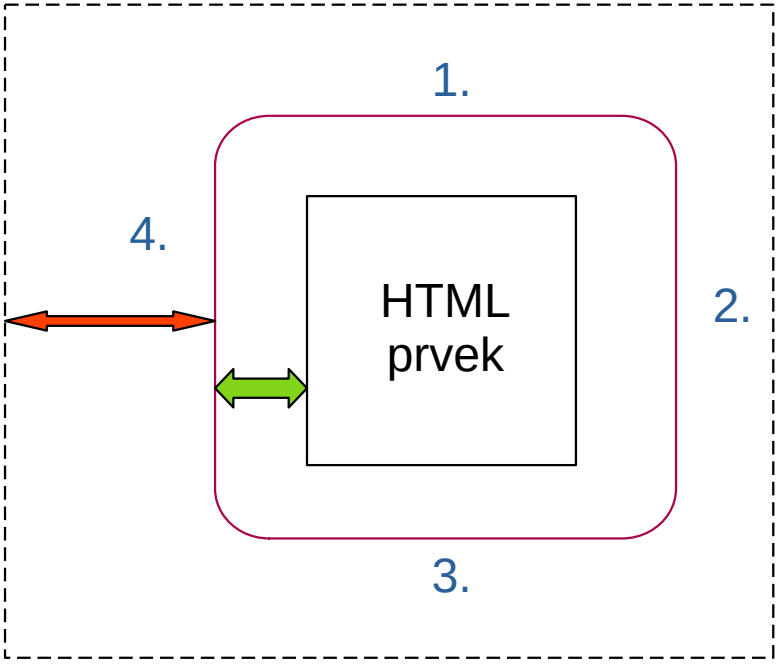
  `background-image: url("obrazek.gif");`
  `background-position: right top;` – umístění v rámci elementu
  `background-repeat: repeat-x | no-repeat` – dá se použít na různé textury
  `background-attachment: fixed;` – nastaví, zda je pozice pozadí svázaná s dokumentem nebo oknem prohlížeče
- Nastavení pozadí jednou vlastností

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

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

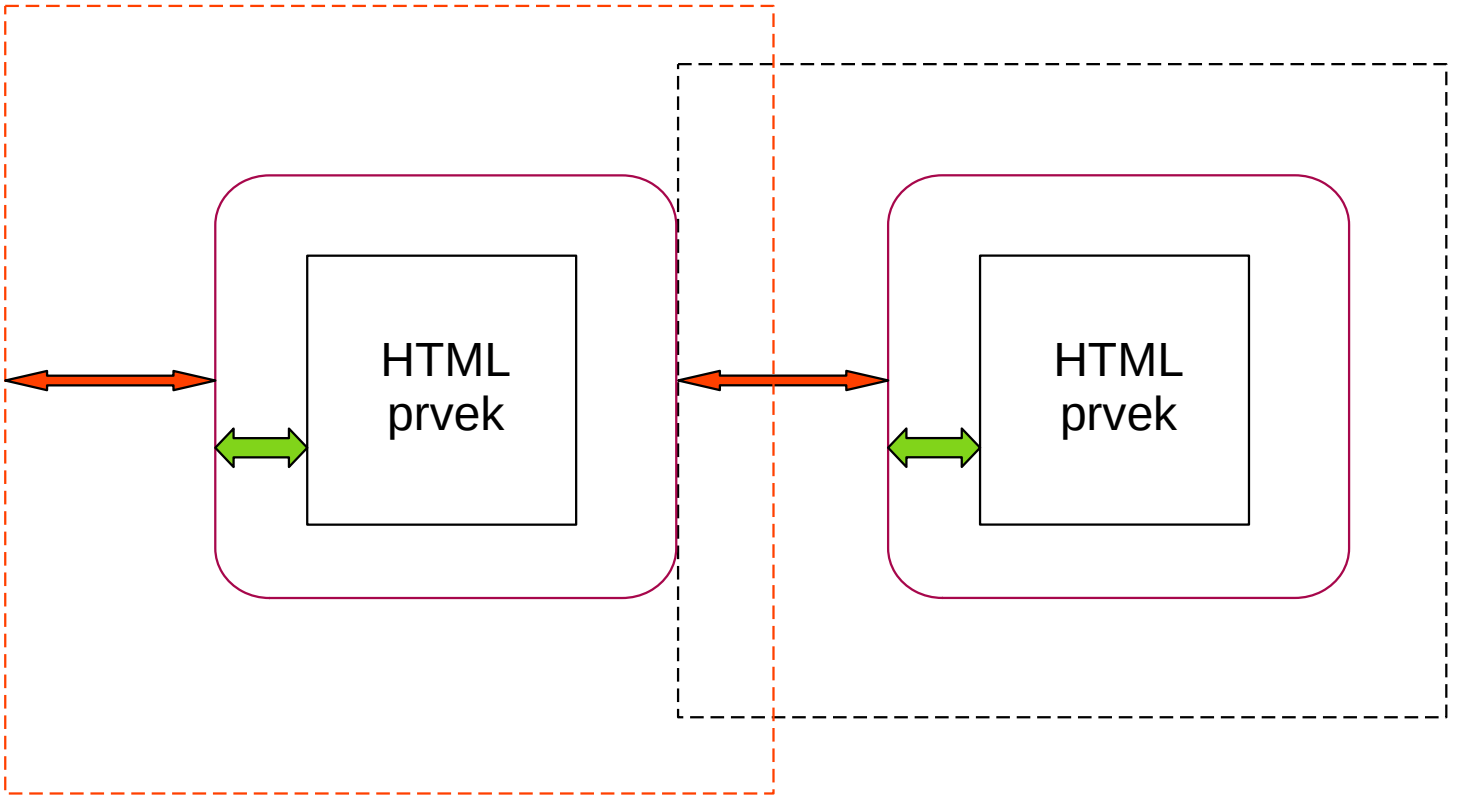
Box Model (Okraje a rámeček)

- 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`



Box Model (Okraje a rámeček)

- Margin dvou prvků se může překrývat



Rámeček (border)

`border: 2px solid blue;` (nastavení všech vlastností najedou)

- `border-style:`
  - `dotted` – tečkovaně
  - `dashed` – čárkovaně
  - `solid` – normální čára
  - `double` – dvojitě
  - `groove` – 3D vyrytě
  - `ridge` – 3D vystouple
  - `inset` – zamáčknutě
  - `outset` – vystouple
  - `none` – bez rámečku
  - `hidden` – skrytý rámeček
- Smíšený rámeček (nahore vpravo dole vlevo)

```
border-style: dotted dashed solid double;
```

- Tabulka bez rámečku `border-collapse: collapse;`

Stíny

- Stín textu (text-shadow)

```
text-shadow: v_posun h_posun rozmazání barva

color: white; text-shadow: 2px 2px 4px #000000;
```

- Stín rámečku (box-shadow)

```
box-shadow: v_posun h_posun rozmazání
            rozšíření barva

box-shadow: 3px 10px 10px 5px #555
```

Text se stínem

Transformace

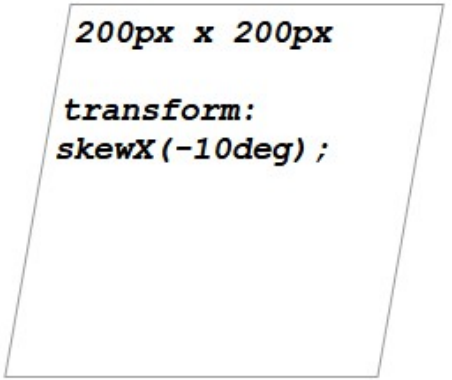
- 2D transformace:

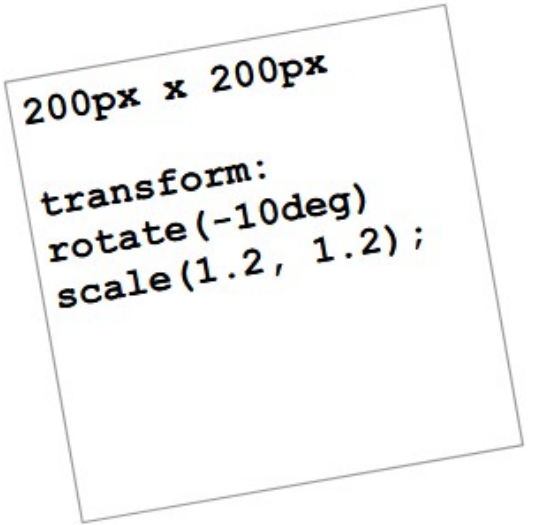
```
translate(x px, y px)
rotate(20deg)
scale(2,3) – 2x širší, 3x delší
skewX(20deg), skewY(20deg) – zmáčknutí podle osy x, y
skew(X,Y)
matrix(scaleX(), skewY(), skewX(), scaleY(),
       translateX(), translateY())
```

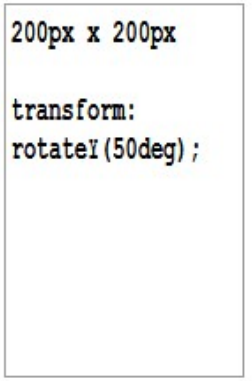
- 3D transformace:

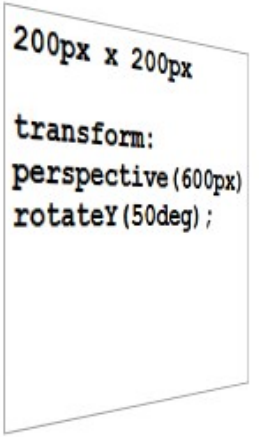
```
rotateX(90deg)
rotateY(90deg)
rotateZ(90deg)
```

Příklady transformací









Tranzice

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

`transition: width 2s, height 3s`

`transition-timing-function:`

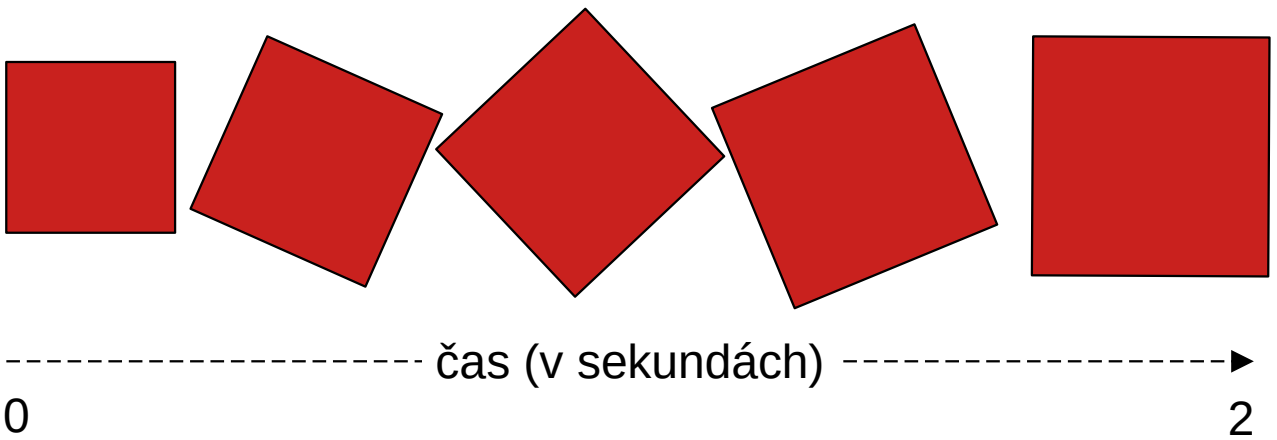
- `ease` – default
- `linear`
- `ease-in`
- `ease-out`

`transition-delay: 1s`

`transition: width 2s linear 1s`

Tranzice – příklad

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

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



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 (1in = 2.54cm)
`px`	Pixels (1px = 1/96 in)
`pt`	Typographical points (1pt =1/72 in)
`pc`	Picas (1pc = 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;
```

Layout stránek

Umístění a zobrazení prvků na stránce

- Základní nástroje:
  - Vlastnost `float`
  - Pozicování
  - Vlastnost `display`

Vlastnost float a obtékání elementů

- Element se “vznáší” – vlevo nebo vpravo, ostatní elementy kolem něj “obtékají”

```
float: left|right|none
```

- Ostatním prvkům je možné nastavit, aby kolem float elementů neobtékaly, pomocí vlastnosti clear:

```
clear: left; clear: right; clear: both;
```

  - Na jedné (nebo obou) stranách se nesmí “vznášet” žádný element (obsah elementu s nastavenou vlastností `clear` je posunut pod “vznášející” se element)

Vlastnost float – příklad

Libějovické Svobodné Hory

Libějovické Svobodné Hory jsou malá vesnice, část obce Stožice v okrese Strakonice. Nachází se asi 2,5 km na jihozápad od Stožic, pod Svobodnou horou. Je zde evidováno 22 adres. V roce 2011 zde trvale žilo 43 obyvatel.

Libějovické Svobodné Hory leží v katastrálním území Křepice u Vodňan o výměře 3,99 km^2.

První písemná zmínka o vesnici pochází z roku 1840.

Pamětihodnosti: Zemědělský dvůr Jarov (kulturní památka ČR), Socha svatého Jiří v lese jižně od vsi, Dva křížky v jižní části vesnice

Bavorovské Svobodné Hory

`float:left`

`float:right`

`clear:both`

Zobrazení prvků

- Pro každý HTML element je dáno, jestli se zobrazuje v toku textu – v řádku (inline) nebo jako samostatný blok (block)
- Toto můžeme změnit pomocí vlastnosti `display`

  `display: block|inline`

  `display: none`

  `display: inline-block` – jako inline, ale je možné nastavit šířku, výšku a okraje

- `visibility: hidden|visible` – element zabírá místo na stránce

Pozicování

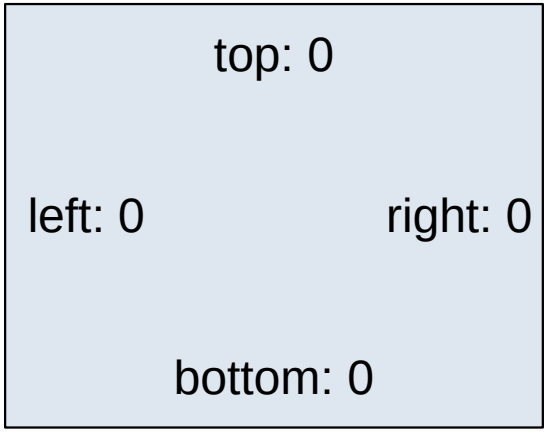
- Prvky se vykreslují v pořadí, ve kterém jsou definované ve zdrojovém kódu stránky
  - Kromě prvků s nastavenou vlastností `float`
- Toto můžeme změnit pomocí pozicování

```
position: static|relative|fixed|absolute|sticky
```

- `z-index` vrstva, ve které se zobrazí prvky, které se překrývají

Pozicování

- `static` – defaultní hodnota
- `relative` – relativně vzhledem k pozici, ve které by byl prvek normálně vykreslen
  - Je možné nastavit vlastnosti `top`, `right`, `bottom`, `left`



- `fixed` – relativně vzhledem k viewportu – zařízení, na kterém se stánka zobrazuje; t.j. když uživatel scrolluje, prvek zůstává na stejném místě
  - Nastavuje se pomocí `top`, `right`, `bottom`, `left`

. . .

Pozicování

...

- `absolute` – pozice je relativní vůči nejbližšímu napozicovanému předku

- `sticky` – něco mezi `relative` a `fixed`, pozice prvku je relativní; pokud ale uživatel scrolluje a prvek by nebyl vidět, zůstane “viset” v dané pozici (jako při `position: fixed`)
  - Fixní pozice se nastavuje pomocí `top`, `right`, `bottom`, `left`

CSS vlastnosti související s pozicováním

- `width`, `height` – šířka, výška prvku

- `min-width`, `max-width` – minimální a maximální šířka
- `min-height`, `max-height` – minimální a maximální výška

- `top`, `right`, `bottom`, `left` – vzdálenost od příslušného okraje

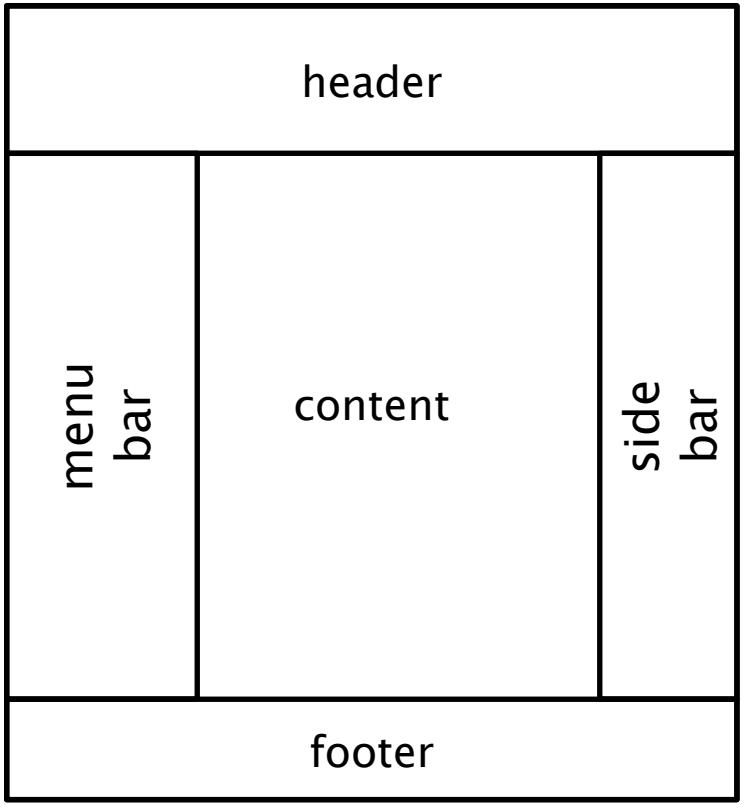
Přetékání, scrollování

- vlastnost `overflow` – funguje pro elementy s nastavenou výškou
- `overflow:`
  - `visible` - default; obsah, který přetekl, se zobrazí i vně elementu
  - `hidden` - obsah je oříznut
  - `scroll` - obsah je oříznut, dá se v něm ale scrollovat pomocí scrollbar
  - `auto` - scrollbar jenom když je potřeba

Layout webové stránky

= rozložení jednotlivých prvků na stránce

- DŘÍVE:
  - Web designer musí vyřešit:
    - Scrollování stránky a jejích částí
    - Co se stane, když obsah nějakého prvku “přetéká”
    - …
- DNES:
  - většinou volnější layout



Layout webové stránky – přístupy

- (Historicky) existují různé způsoby, jak vytvořit rozložení prvků s postranními panely
  - Tabulky – NEPOUŽÍVAT
  - Využití CSS stylů
    - Pomocí vlastnosti float (obtékání elementů textem)
    - Pomocí pozicování
    - Pomocí flexbox a grid – moderní přístup – ANO, používat

Flexbox, grid

- Flex
  - Jednorozměrný layout
  - Flexibilně přizpůsobené rozměry prvků

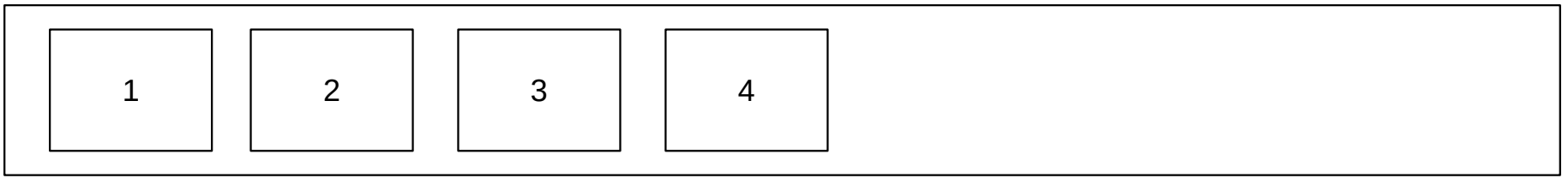
- Grid
  - 2D layout

Flexbox

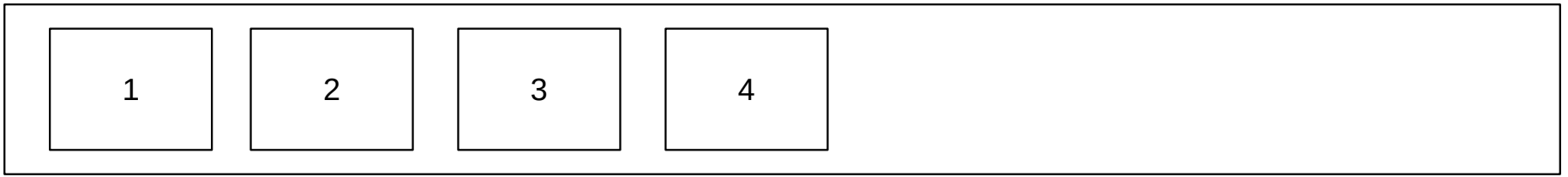
- Layout je tvořen jedním rodičovským elementem s jedním nebo více potomků
- container - obal pro prvky
- items - jednotlivé prvky, které chceme rozložit

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

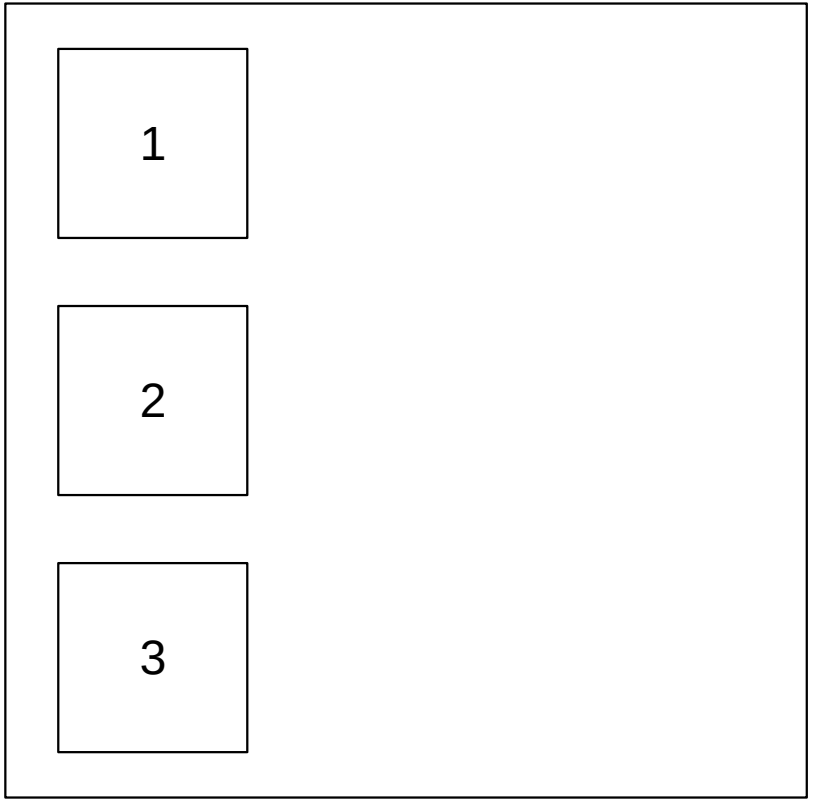
```
.flex-container {
  display: flex;
}
```



Flexbox



- Flexbox určuje, jak budou prvky naskládány uvnitř kontejneru
- `flex-direction`
  - `row, column`
- … zalamování, zarovnání

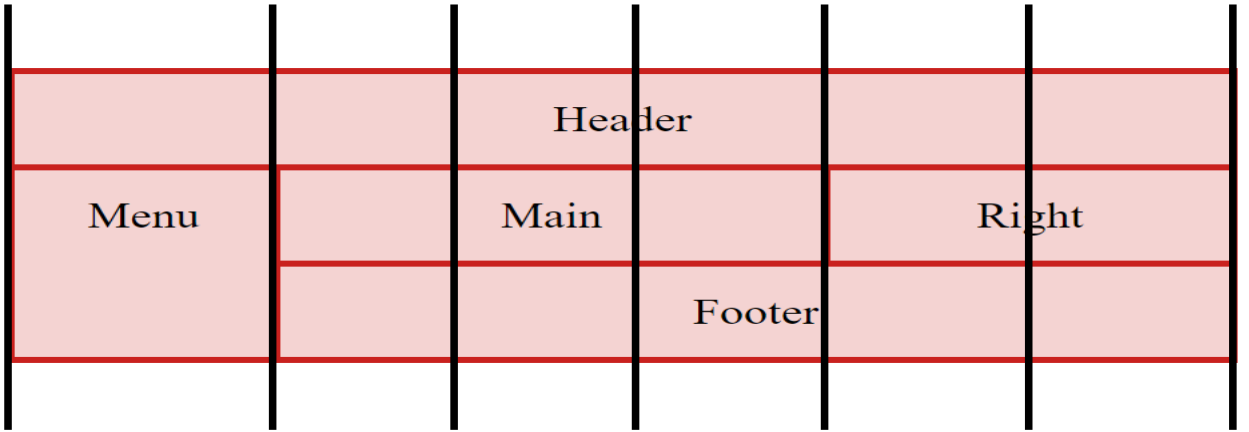


grid

- rozložení prvků na stránce do mřížky - řádků a sloupců
- moderní přístup (místo floatů a pozicování)

grid

- rozložení prvků na stránce do mřížky - řádků a sloupců
- moderní přístup (místo floatů a pozicování)
- celý layout je rozdělen na sloupce



grid

- Layout je tvořen jedním rodičovským elementem s jedním nebo více potomků
  - potomci jsou automaticky součástí mřížky

1	2
3	4
5	6

grid

- Layout je tvořen jedním rodičovským elementem s jedním nebo více potomků
  - potomci jsou automaticky součástí mřížky

1	2
3	4
5	6

```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
</div>
```

grid

- Layout je tvořen jedním rodičovským elementem s jedním nebo více potomků
  - potomci jsou automaticky součástí mřížky

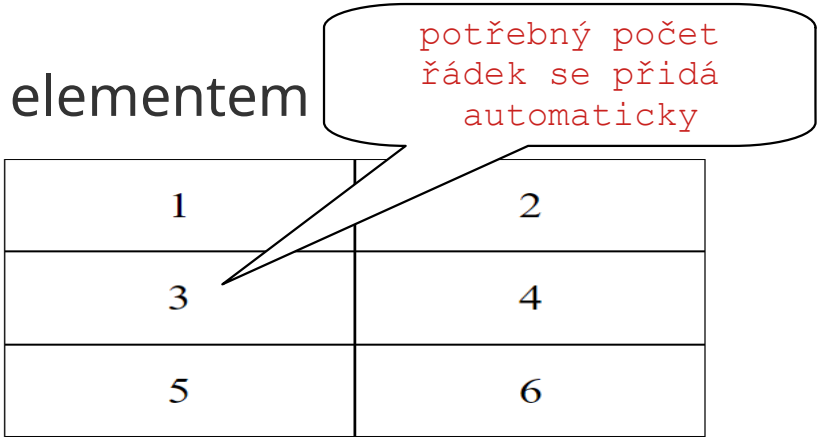
1	2
3	4
5	6

```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
</div>
```

```
.grid-container {
  display: grid;
  grid-template-columns: auto auto;
}
.grid-item {
  border: 1px solid black;
  ...
}
```

grid

- Layout je tvořen jedním rodičovským elementem potomků
  - potomci jsou automaticky součástí mřížky

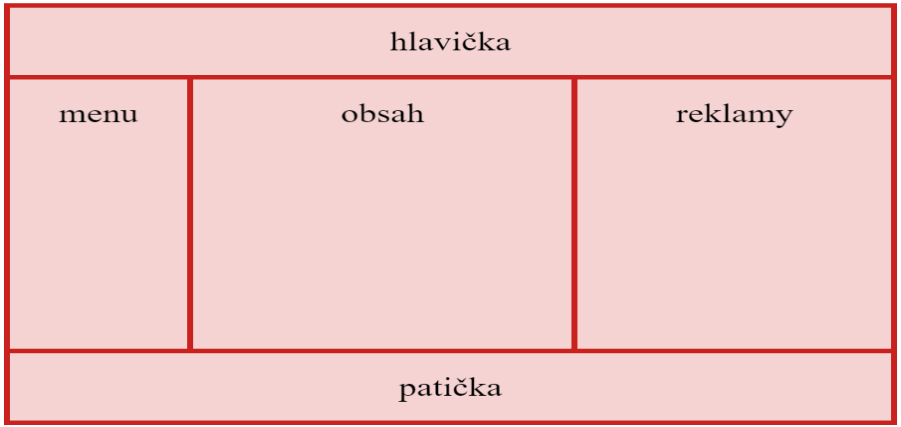



```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
</div>
```

```
.grid-container {
  display: grid;
  grid-template-columns: auto auto;
}
.grid-item {
  border: 1px solid black;
  ...
}
```

"svatý grál" pomocí `grid`

```
.item1 { grid-area: header; }
.item2 { grid-area: menu; }
.item3 { grid-area: main; }
.item4 { grid-area: right; }
.item5 { grid-area: footer; }

.grid-container {
  grid-template-areas:
    'header header header header header header'
    'menu main main main right right'
    'footer footer footer footer footer
footer';
}
```



@media

- Umožňují developerovi reagovat na typ zobrazení stránky a další vlastnosti zařízení, na kterém si uživatel stránku prohlíží
  - Typ zobrazení: all, screen, print, speech

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

  - Další vlastnosti zařízení:
    - Velikost displeje (viewportu), orientace, barevná hloubka, ...
- Použití
  - Přímo v CSS souboru
  - Při připojování externího .css souboru v `<link>` uvnitř `<head>`

@media

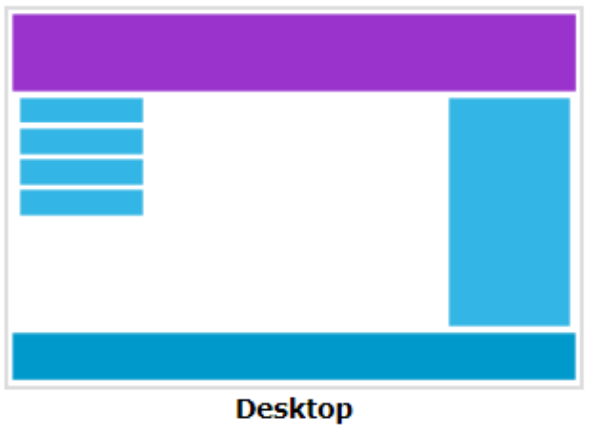
- Podmínky je možné spojovat pomocí and nebo čárky (která funguje jako or)

```
@media screen and (min-width: 480px) {
    CSS rules...
}
```

- Styly uvnitř složených závorek se aplikují, pokud je šířka okna na obrazovce, na které se stránka zobrazuje, větší než 480px

Responsivní design

- Cíl: aby stránka vypadala dobře na všech zařízeních (počítač, notebook, tablet, telefon)
- Jak? Stejný obsah, změna velikostí prvků, přesun prvku na jiné místo
- Jak konkrétně?
  - Rozměry zadané relativně, v %, nebo pomocí jednotek `vh`, `vw`
  - Různá rozložení prvků (CSS styly) pro různá zařízení (pomocí `@media`)



Viewport

- Oblast pro webovou stránku na zařízení uživatele
  - Např. velikost displeje telefonu, velikost okna prohlížeče

- V HTML5 můžeme viewport nastavit v hlavičce stránky

```
<meta name="viewport" content="width=device-width,
                               initial-scale=1.0">
```

- `width=device_width` nastaví stránce stejnou šířku, jako má zařízení, na kterém se stránka zobrazuje
- `initial-scale` nastaví počáteční přiblížení

Nastavení viewportu – příklad

S nastaveným viewportem a šířkou obrázku

```
img {
  max-width: 100%;
}
```

Otázky…

?