2D a 3D transformace, Flexible Boxes

2D transformace

- CSS vlastnosti pro afinní geometrické transformace HTML prvků
- aplikuje se po vykreslení → neovlivňují layout ostatních prvků
- mohou být akcelerované → rychlejší než absolutní pozicování
- občas nutnost použití vendor prefixů

```
1. div {
2.     transform: translate(100%, 50px);
3.     transform: rotate(45deg);
4.     transform: scale(0.2);
5.     transform: skewX(30deg);
6.     transform: matrix(a, b, c, d, tx, ty);
7. }
```

- vlastnost „**transform-origin**“ určuje počátek souřadného systému
    - výchozí 50% 50

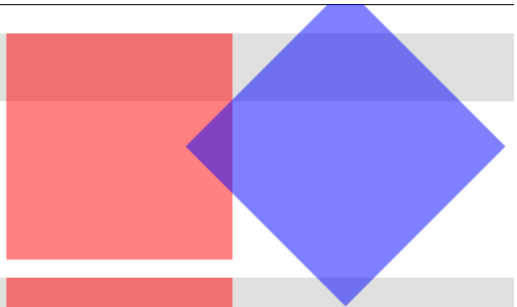
```
1. div {
2.     transform-origin: 0 0;
3.     transform: rotate(45deg);
4. }
```

- transformace lze kombinovat

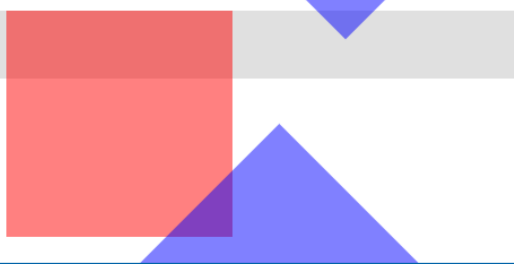
```
1. div {
2.     transform: rotate(45deg) translate(50px, 50px);
3. }
```

Pořadí

```
transform: translate(100%, 0) rotate(45deg)
```

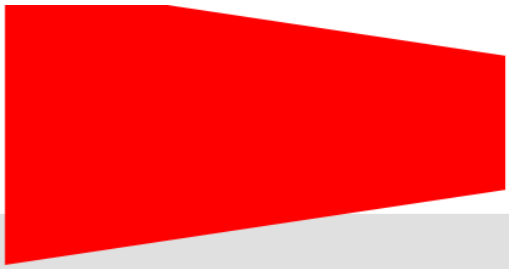


```
transform: rotate(45deg) translate(100%, 0)
```



3D transformace

- rozšíření transformačních funkcí
- translateZ, translate3d, scaleZ, scale3d
- rotate3d, rotateX, rotateY, rotateZ
- nejistá podpora prohlížečů, závisí i na HW
- pro perspektivní zkreslení → nutno definovat parametry 3D prostoru
    - perspective = na rodičovském prvku určuje míru zkreslení
    - transform-style = určuje, jak se chovají 3D transformace na potomcích již trans. prvků



```
#parent {
    perspective: 500px;
}

#parent div {
    transform: rotateY(45deg);
}
```

CSS Flexible Box Module

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

http://the-echoplex.net/flexyboxes/

- nový pozicovací algoritmus do CSS
- pro polohování prvků vedle sebe / pod sebe
- náhrada za (slabý) float

```
<div id="parent">
    <div id="child1"></div>
    <div id="child2"></div>
    <div id="child3"></div>
</div>
```

```
#parent {
    display: flex;
    flex-direction: row;
    justify-content: space-between;
}
```

- komplexní systém mnoha CSS vlastností
- popisuje vždy jen rozložení několika sourozenců v rámci rodiče

- **relativní rozměry boxů:**
  - **flex-grow** = relativní síla růstu
  - **flex-shrink** = relativní síla zmenšení
  - **flex-basic** = základní rozměr (výška/šířka)

- **pořadí**:
  - vlastnost „**order**“ pro změnu pořadí prvků
  - podobný princip jako u z-index
  - dovoluje zachovat obsah první
  - často v kombinaci s media queries

- **další vlastnosti:**
  - **align-items**, **align-self** = zarovnání v kolmé ose (cross axis)
  - **flex-wrap** = povolení/zakázání zalamování

Písma - @font-face

- definice vlastního písma
- podporováno od IE4
- nutnost dodat soubor(y) s definicí písma
- pozor na CORS
- vhodné pro typografii, dříve pro ideogramy (nyní lépe SVG)

```
@font-face {
    font-family: MujFont;
    src: local("Muj Font"), url(MujFont.ttf);
}

body {
    font-family: MujFont;
}
```

- **formáty**:
  - EOT proprietární od Microsoftu
  - TTF, OTF
  - WOFF / WOFF2: Web Open Font Format

- **data URI:**

```
@font-face {
    font-family: "MujFont";
    src: url(data:application/x-font-woff;charset=utf-8;base64,d09......AAA==) format("woff"),
         url("MujFont.ttf") format("truetype");
}
```

- **FOUT (Flash of Unstyled Text):**
    - Co má prohlížeč udělat, než se písmo načte?
    - čekat vs. zkusit jiné?
    - velký problém u HTML canvasu
    - budoucnost: Font Loading API

- **poznámky**:
    - vlastní písma → vyšší objem přenášených dat
    - kvalita písma je většinou subjektivní
    - odlišné renderování v různých prohlížečích a OS
    - Google Web Fonts