

BALTIC TALENTS ACADEMY

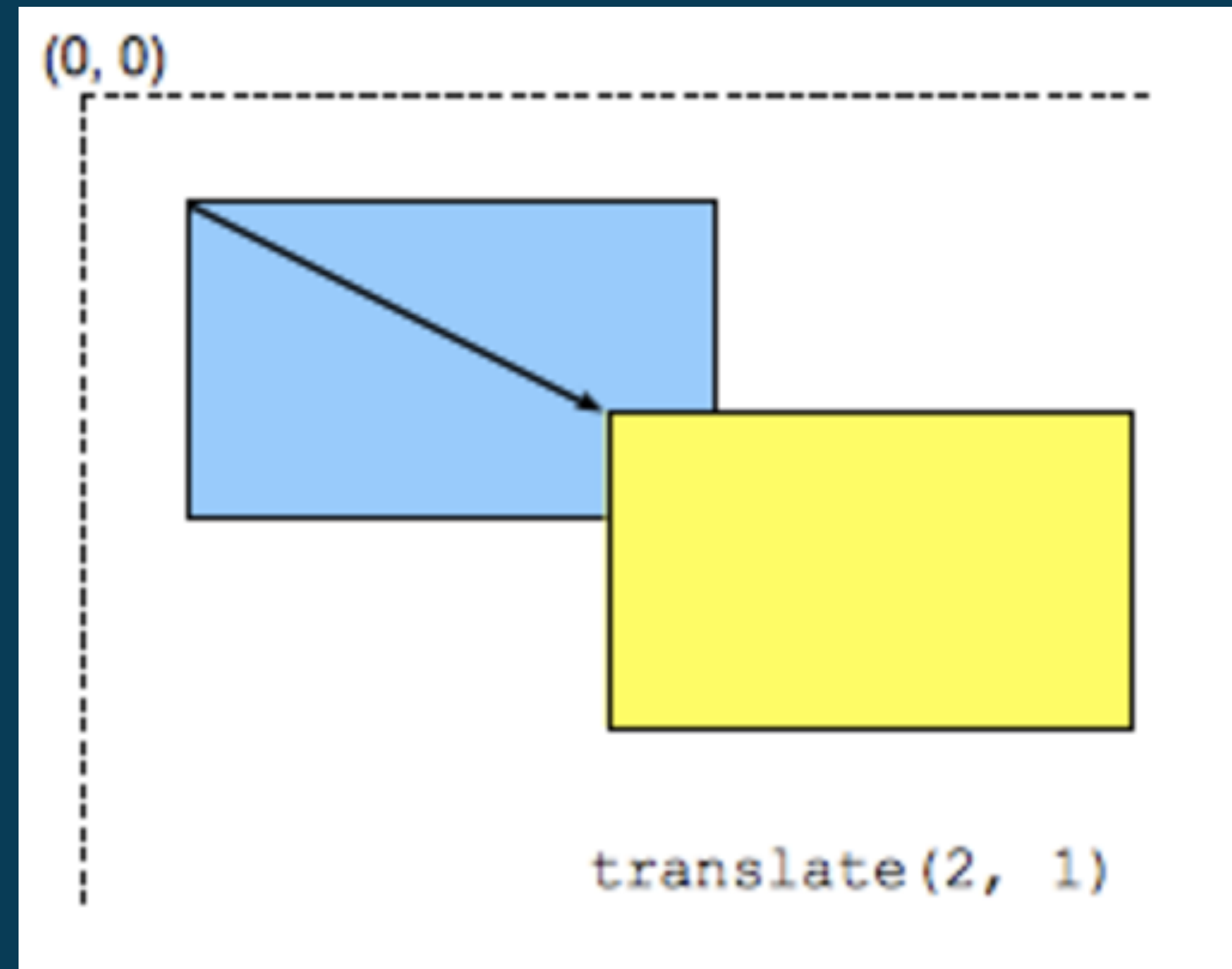
CSS FLEXIBLE BOXES

KARTOJIMAS

- ▶ transform: translate
- ▶ transform: rotate, transform-origin
- ▶ transform: scale
- ▶ transition
- ▶ @keyframes, animation

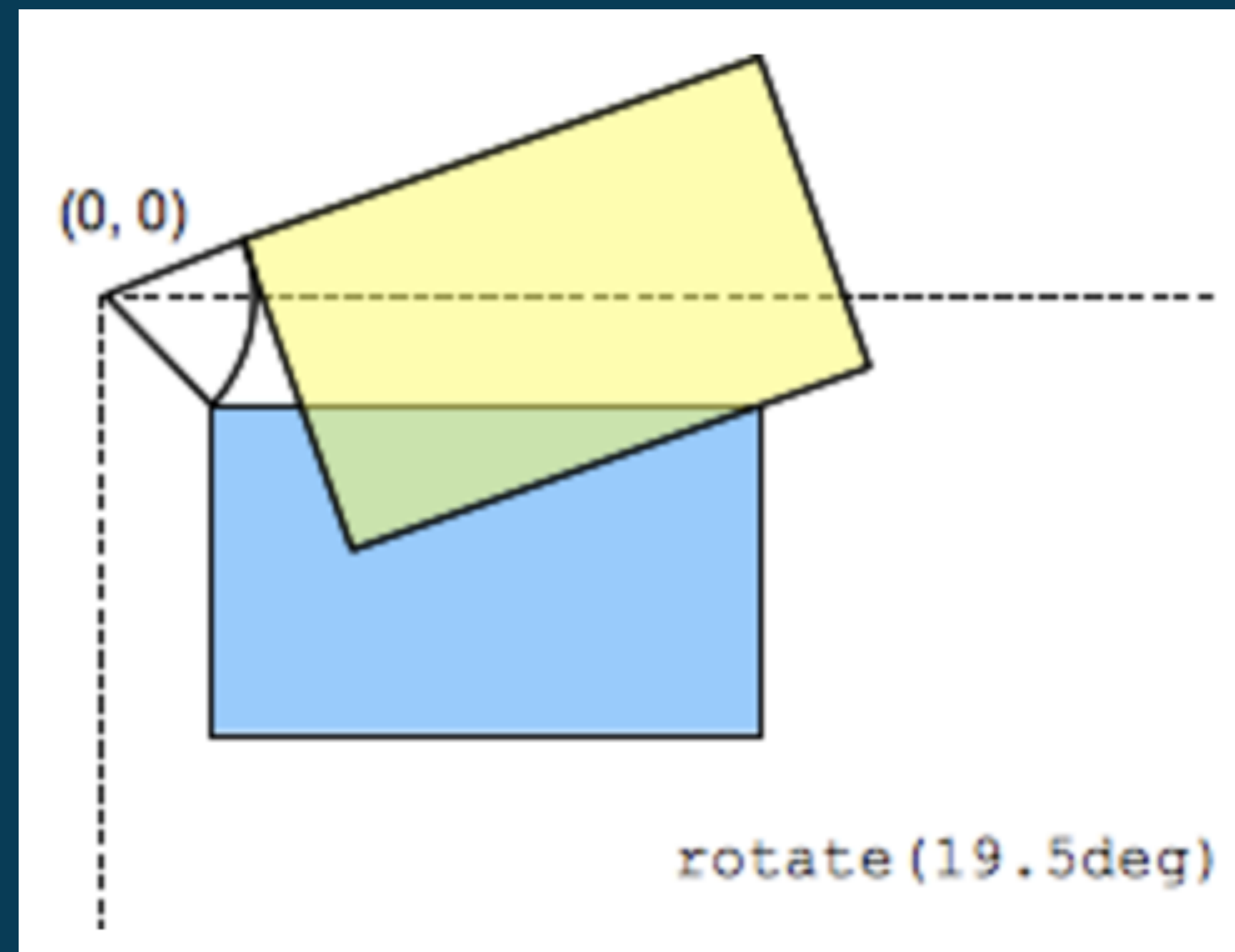
TRANSFORMACIJOS 2D - PERKELTI (TRANSLATE)

- ▶ `transform: translate(tx, ty);`
- ▶ `transform: translateX(tx);`
- ▶ `transform: translateY(ty);`



TRANSFORMACIJOS 2D - PASUKTI (ROTATE)

► `transform: rotate(19.5deg);`



TRANSFORMACIJOS 2D - CENTRAS

- ▶ Sukimo transformacijai būtina žinoti kur yra transformacijos centras, t.y. taškas aplink kurį sukamas (transformuojamas) objektas. Toks taškas pagal nutylėjimą yra objekto viduryje, bet jį galima apibrėžti ir kitur:
- ▶ `transform-origin: x-axis y-axis;`
 - ▶ *x-axis*: left | center | right | *length* | %
 - ▶ *y-axis*: top | center | bottom | *length* | %

TRANSFORMACIJOS 2D - DIDINTI / MAŽINTI (SCALE)

- ▶ `transform: scale(sx, sy);`
- ▶ `transform: scaleX(sx);`
- ▶ `transform: scaleY(sy);`

sx, sy - tai didinimo/mažinimo koeficientai pagal x ir y ašį atitinkamai, pvz.:

- 1 - niekas nesikeičia,
- 3 - tris kartus didinamas,
- 0.5 - du kart mažinamas

PERĖJIMAI (TRANSITIONS)

Animacija - tai elemento savybių pasikeitimas per nustatytą laiką.

Kaip turi vykti elemento kažkokios savybės pakeitimas yra nurodoma perėjimo (transitions) savybėse:

- ▶ *transition: property duration function delay, ...;*

pvz.: *transition: color 1s;*

- ▶ *transition-property: name;*

- ▶ *transition-duration: time;*

- ▶ *transition-timing-function: function-name;*

- ▶ *transition-delay: time;*

Daugiau info: https://www.w3schools.com/cssref/css3_pr_transition.asp

ANIMACIJOS

- ▶ `@keyframes name {`
 `from { background-color: red; width: 10px; }`
 `to { background-color: blue; width: 200px; }`
 `}`
- ▶ `@keyframes name {`
 `0% { background-color: red; width: 10px; }`
 `50% { background-color: yellow; width: 100px; }`
 `100% { width: 200px; }`
 `}`

ANIMACIJOS

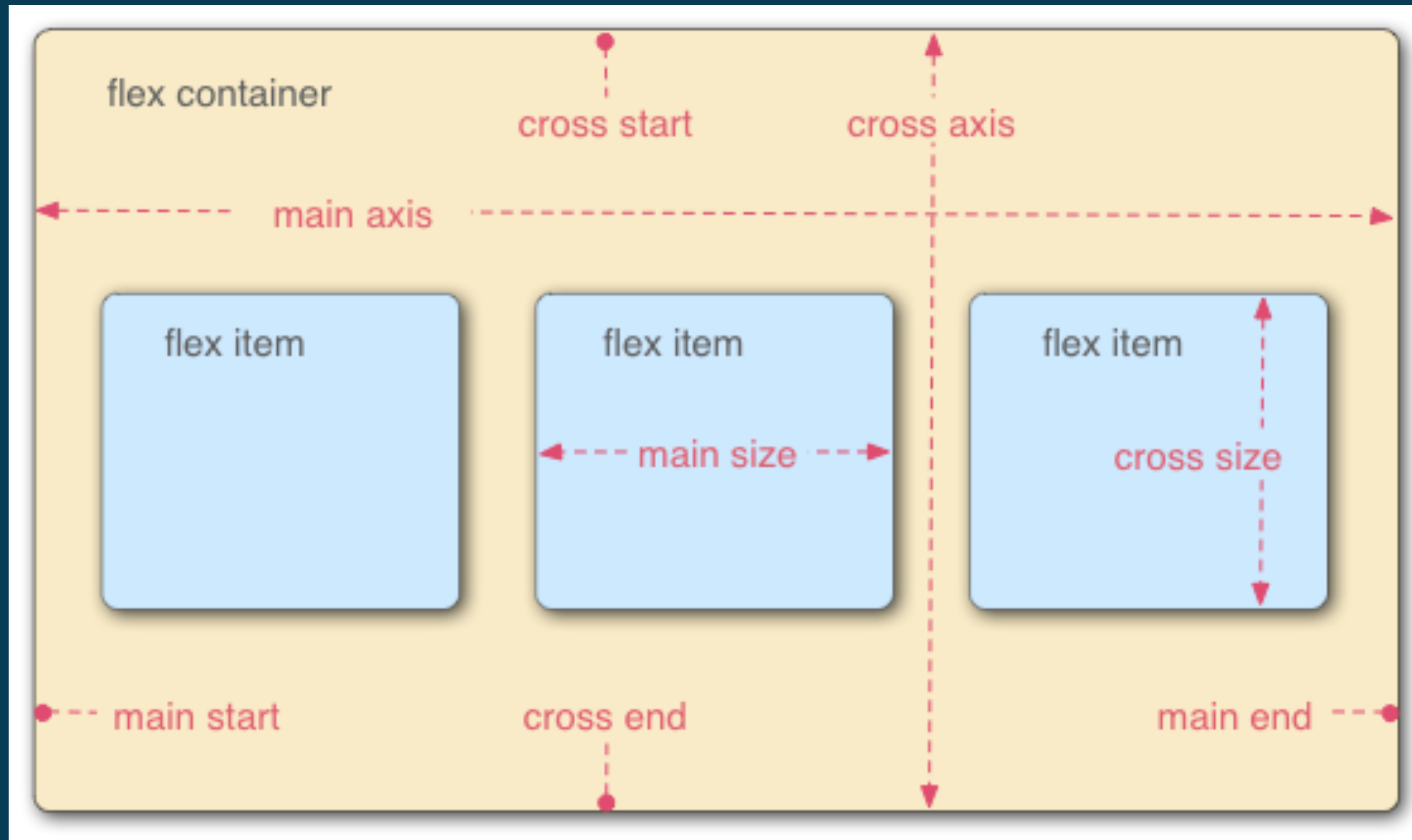
- ▶ *animation: name duration timing-function delay iteration-count direction fill-mode play-state*
- ▶ *animation-name: name*
- ▶ *animation-duration: time*
- ▶ *animation-timing-function: ease | linear | ease-in | ease-out | ease-in-out*
- ▶ *animation-delay: time* - reikšmė pagal nutylėjimą **0s**
- ▶ *animation-iteration-count: number | infinite* - reikšmė pagal nutylėjimą **1**
- ▶ *animation-play-state: paused | running*

ANIMACIJOS

- ▶ `animation-direction`: **normal** | `reverse` | `alternate` | `alternate-reverse` | ...
 - ▶ `alternate` - animuojama pirmyn, o po to atgal į pradinės reikšmės
 - ▶ `alternate-reverse` - animuojama atgal, o po to pirmyn
- ▶ `animation-fill-mode`: **none** | `forwards` | `backwards` | `both` | ...
 - ▶ **none** - savybės prieš ir po animacijos nesikeičia
 - ▶ `forwards` - savybės įgyja naujas animacijos pabaigos reikšmes
 - ▶ `backwards` - pirskiriamos pradinės savybės prieš pradedant animaciją laiko intervalui nurodytam `animation-delay`
 - ▶ `both` - veikia ir `forwards` ir `backwards` taisyklės

Daugiau info: https://www.w3schools.com/cssref/css3_pr_animation.asp

LANKSČIOS DĖŽUTĖS :)



PAGRINDAI

- ▶ ***flex container*** - konteineris kurio **display** reikšmė yra **flex** arba **inline-flex**
- ▶ **flex-direction** nurodo pagrindinę ašį (***main axis***). Kryžminė ašis (***cross axis***) visada statmena pagrindinei
- ▶ **justify-content** - nurodo kaip elementai dëllojami pagal pagrindinę ašį
- ▶ **align-items** - nurodo kaip elementai dëllojami pagal kryžminę ašį
- ▶ **align-self** - nusirodo elemente jei reikia pakeisti **align-items** nurodytas reikšmes
- ▶ **order** - nusirodo elemente jei reikia pakeisti dëllojimo eilę. Reikšmė pagal nutylėjimą 0. Jei elementų **order** reikšmės vienodos tai dëllojami pagal aprašymo tvarką

KRYPTYS

flex-direction nurodo pagrindinę ašį (***main axis***). Kryžminė ašis (***cross axis***) visada statmena pagrindinei:

- ▶ **flex-direction: row** | row-reverse | column | column-reverse;
- ▶ **flex-wrap: nowrap** | wrap | wrap-reverse;

DĖSTYMAI PAGAL AŠIS

justify-content - nurodo kaip elementai dėliojami pagal pagrindinę ašį:

► **justify-content: flex-start | flex-end | center | space-between | space-around;**

align-items - nurodo kaip elementai dėliojami pagal kryžminę ašį:

► **align-items: stretch | flex-start | flex-end | center;**

ELEMENTŲ DYDŽIAI

flex - nurodo kaip keičiasi elementų dydžiai:

- ▶ **flex:** *flex-grow flex-shrink flex-basis*;
- ▶ **flex-grow:** *number* - kaip didėja elementas kitų elementų atžvilgiu, reikšmė pagal nutylėjimą 0
- ▶ **flex-shrink:** *number* - kaip mažėja elementas kitų elementų atžvilgiu, reikšmė pagal nutylėjimą 1
- ▶ **flex-basis:** **auto** | *number* - pradinis elemento dydis

ELEMENTŲ DYDŽIAI - FLEX-GROW

Jei elementai užima mažiau vietos negu konteineris, tai laisva vieta paskirstoma pagal elemento **flex-grow** savybę. Kiekvienas elementas gauna savo dalį, jei vieno elemento **flex-grow** reikšmė yra 1, o kito 3, tai pastarasis gauna 3 kart daugiau.

ELEMENTŲ DYDŽIAI - FLEX-SHRINK

Jei elementai užima daugiau vietos negu konteineris, tai elementai mažinami pagal **flex-shrink** savybę. Kiekvienas mažinamas ir jei vieno **flex-shrink** reikšmė yra 1, o kito 3, tai pastarasis praranda 3 kart daugiau.

PRATIMAS / UŽDAVINYS

Sukurkite puslapio išdėstymą (layout) pagal paveiksluką šį kartą panaudojant flex konteinerius.

