



VILNIUSCODINGSCHOOL

Flexbox

Turinio formatavimas



start

center

end



Gaming Paradise!

Xbox will change your life.

Buy Now

Kai naudojame flexbox, turinys yra formatuojamas pagal ašį, kuri turi pradžią, pabaigą ir vidurį.

```
display: flex;
```

justify-content



justify-content naudojamas tarpų/erdvės formatavimui pagal ašies kryptį.

```
display: flex;  
justify-content: flex-start;
```

start

center

end



Gaming Paradise!

Xbox will change your life.

Buy Now

justify-content



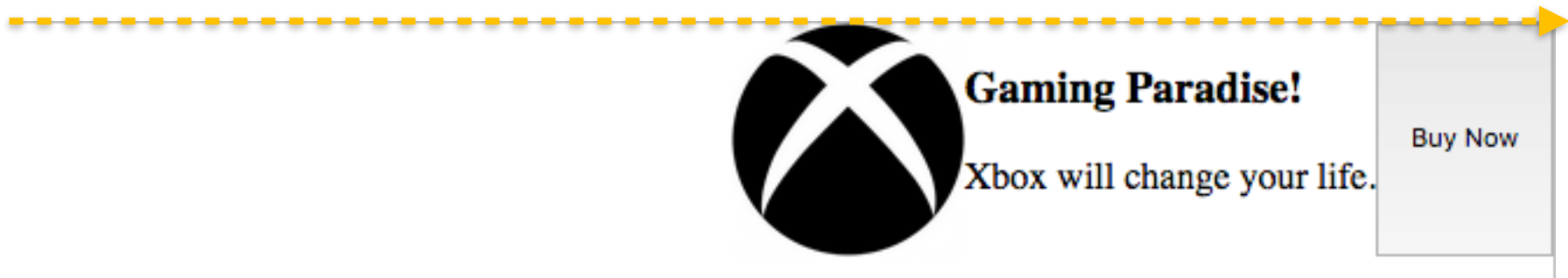
justify-content naudojamas tarpų/erdvės formatavimui pagal ašies kryptį.

```
display: flex;  
justify-content: flex-end;
```

start

center

end



justify-content



justify-content naudojamas tarpų/erdvės formatavimui pagal ašies kryptį.

```
display: flex;  
justify-content: space-between;
```

start

center

end



Gaming Paradise!

Xbox will change your life.

Buy Now

justify-content



justify-content naudojamas tarpų/erdvės formatavimui pagal ašies kryptį.

```
display: flex;  
justify-content: space-around;
```

start

center

end



Gaming Paradise!

Xbox will change your life.

Buy Now

justify-content



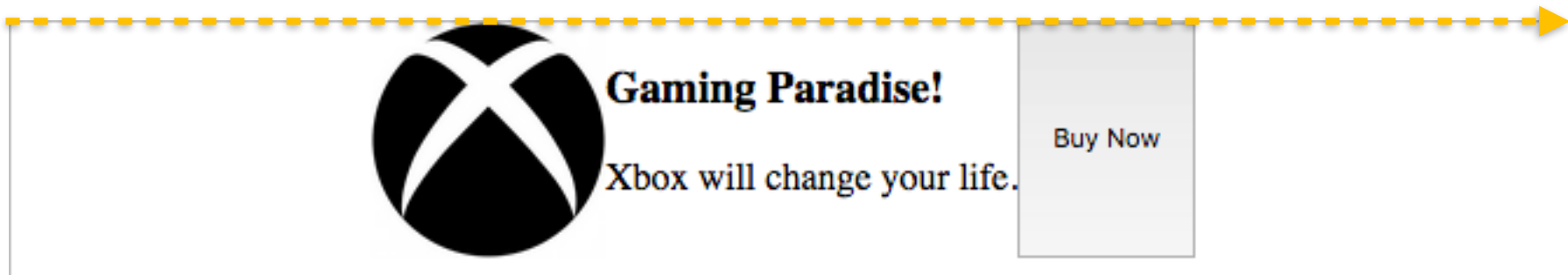
justify-content naudojamas tarpų/erdvės formatavimui pagal ašies kryptį.

```
display: flex;  
justify-content: center;
```

start

center

end



justify-content



flex elementams galime pridėti margin ir padding:

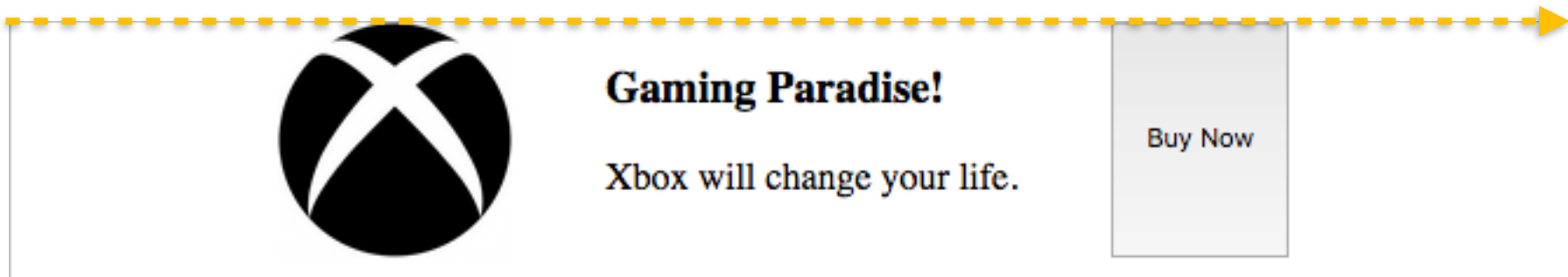
```
display: flex;  
justify-content: center;
```

```
img, div, button {  
  margin: 0 20px;  
}
```

start

center

end



order



order naudojamas eiliškumui nustatyti pagal ašies kryptį. Default yra 0, skaičiai gali būti teigiami ir neigiami. Kai order: 0, flex elementai rikiuojami pagal jų užrašymo eiliškumą.

```
nav {  
  display: flex;  
  justify-content: center;  
}
```

Kai visi order: 0



Gaming Paradise!

Contact Us

order



```
a.gaming {  
  order:-1;  
}
```

Kai gaming turi order: -1, o visi kiti turi order: 0

Gaming Paradise!



Contact Us

order



```
<nav>
  <a class="home" href="#"></a>
  <p class="motto">Gaming forever!</p>
  <a class="gaming" href="#">Gaming Paradise!</a>
  <a class="contact" href="#">Contact Us</a>
</nav>
```

```
p.motto {
  order:1;
}
```

Nors .motto eina po nuotraukos, order: 1 nustumia .motto į galą, kadangi visi kiti elementai turi order: 0



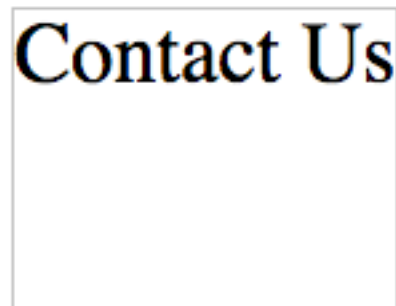
Gaming Paradise!
Contact Us
Gaming forever!

align-items



align-items naudojamas turinio išdėstymui pagal kryptį, kuri yra stačiai nuo ašies krypties, t.y. jei ašies kryptis yra horizontali, tuomet align-items naudojamas vertikaliam turinio išdėstymui, o jei ašies kryptis yra vertikali, tuomet align-items naudojamas horizontaliam išdėstymui.

Default yra align-items:stretch, kiti variantai: flex-start, flex-end, baseline ir center



align-items:stretch;

align-items



Gaming Paradise!

Contact Us

`align-items: flex-start;`

align-items



Gaming Paradise!

Contact Us

`align-items:flex-end;`

`align-items:baseline;`

align-items



Gaming Paradise!

Contact Us

`align-items:center;`

flex-grow



flex-grow naudojamas nustatymui santykio, kurį turėtų užimti flex elementas. Default yra 0, priima skaičius.

```
.vienas, .trys {  
  flex-grow:1;  
}  
  
.du {  
  flex-grow:2;  
}
```



```
.vienas, .du, .trys {  
  flex-grow:1;  
}
```



flex-shrink



flex-shrink naudojamas susiaurinimui nustatyti. Default yra 1, priima skaičius. Jei norime, kad ekranui mažėjant, mažėtų tik tekstas, bet ne nuotrauka:

```
.flex1 {  
  flex-grow: 0;  
  flex-shrink: 0;  
  width: 200px;  
}
```

```
.flex2 {  
  flex-grow: 2;  
  flex-shrink: 2;  
}
```



Šiuo atveju 1 konteineris neauga ir nemažėja, jo dydis išlieka 200px, o 2 konteinerio dydis užima likusį ekrano plotį. Mažėjant ekrano pločiui, siaurėja tik 2 konteineris.

flex-shrink ir flex-basis



flex-shrink naudojamas susiaurinimui nustatyti. Default yra 1, priima skaičius. Jei norime, kad ekranui mažėjant, mažėtų tik tekstas, bet ne nuotrauka:

```
.flex1 {  
  flex-grow: 0;  
  flex-shrink: 0;  
  flex-basis: 200px;  
}  
  
.flex2 {  
  flex-grow: 2;  
  flex-shrink: 2;  
}
```

1

2

Šiuo atveju 1 konteineris neauga ir nemažėja, jo dydis išlieka 200px, o 2 konteinerio dydis užima likusį ekrano plotį. Mažėjant ekrano pločiui, siaurėja tik 2 konteineris.

Dydį nustatėme su **flex-basis**, kuris naudojamas pločiams nustatyti, ir priima px, em, rem, %

align-self ir align-content



align-self naudojamas individualių flex elementų išdėstymui, neatsižvelgiant į align-items parametrus. Default yra stretch, taip pat priima flex-start, flex-end, center ir baseline.

align-content naudojamas flex elementų, kuriems buvo pritaikytas wrap, išdėstymui. Default yra stretch, taip pat priima flex-start, flex-end, space-between ir space-around.

Išsamus pavyzdys:

<https://codepen.io/HugoGiraudel/pen/1caa1ef0857a023d0561b5223abef75d>

Sutrumpinimai: flex



flex naudojamas **flex-grow**, **flex-shrink** ir **flex-basis** rašymui vienoje vietoje.

```
.akcija {  
  flex: 1 0 50%;  
}
```

```
.akcija {  
  flex-grow: 1;  
  flex-shrink: 0;  
  flex-basis: 50%;  
}
```

Sutrumpinimai: flex-flow



flex-flow naudojamas flex-direction ir flex-wrap rašymui vienoje vietoje.

```
.akcija {  
    flex-flow: column wrap;  
}
```

```
.akcija {  
    flex-direction: column;  
    flex-wrap: wrap;  
}
```