



VILNIUSCODINGSCHOOL

HTML 5 ir CSS3

<section>



Grupuojame turinį, kuris turi teminį ryšį

```
<div class="section">
  <h2>The Gallery</h2>
  <!-- ... -->
</div>
```

```
<section>
  <h2>The Gallery</h2>
  <!-- ... -->
</section>
```



<header>



Įprastai naudojamas puslapio viršuje, bet įtakoja turinys, o ne pozicija

```
<div class="header">  
  <!-- ... -->  
</div>
```



```
<header>  
  <!-- ... -->  
</header>
```

<footer>



Įprastai naudojamas puslapio apačioje, bet įtakoja turinys, o ne pozicija

```
<div class="footer">  
  <!-- ... -->  
</div>
```



```
<footer>  
  <!-- ... -->  
</footer>
```

<aside>



Sidebar - gretimai esantis turinys

```
<div class="sidebar">  
  <!-- ... -->  
</div>
```



```
<aside>  
  <!-- ... -->  
</aside>
```

<nav>



Puslapio dalis skirta navigacijai (nuorodos į kitus puslapius)

```
<ul class="nav">  
  <!-- ... -->  
</ul>
```



```
<nav>  
  <ul>  
    <!-- ... -->  
  </ul>  
</nav>
```

<article>



Tekstas, kuris gali būti naudojamas nepriklausomai
(naujiena, blog'as)

```
<div class="article">  
  <!-- ... -->  
</div>
```



```
<article>  
  <!-- ... -->  
</article>
```

<main>



Pagrindinis puslapio turinys (tik vienas puslapyje)

```
<div class="main">  
  <!-- ... -->  
</div>
```



```
<main>  
  <!-- ... -->  
</main>
```


`<input type="search">`



```
<input type="text" />
```



```
<input type="search" />
```



<input type="email">



```
<input type="text" />
```



```
<input type="email" />
```

Email Address



<input type="email">



```
<input type="text" />
```



```
<input type="email" />
```

Email Address



<input type="date">



```
<input type="text" />
```



```
<input type="date" />
```

05/02/2014

February 2014

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2

March	21	2011
April	22	2012
May	23	2013
June	24	2014
July	25	2015

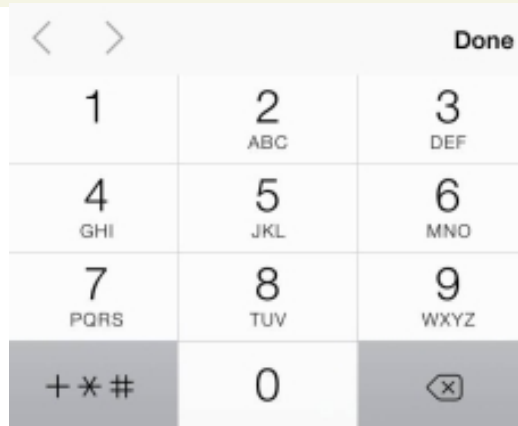
`<input type="tel">`



`<input type="text" />`



`<input type="tel" />`



<datalist>



Iš anksto įvesti pasirinkimai, siūlantys autocomplete

```
<input type="text" list="browsers" />
<datalist id="browsers">
  <option value="Chrome">
  <option value="Firefox">
  <option value="Internet Explorer">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

placeholder



Instrukcija input laukelyje, dingsta, kai rašome ir atsiranda, kai vėl tuščias laukelis

```
<input type="text" value="Enter your email..." />
```



```
<input type="text" placeholder="Enter your email..." />
```



required



Padaro input laukelį privalomą

```
<input type="text" required />
```


pattern



Leidžia priimti įvestą turinį tik nurodytu formatu

```
<input type="text" pattern="[0-9]{3}" />
```

border-radius



Apvalina kampus

```
.box {  
  background: grey;  
  height: 50px;  
  width: 200px;  
}
```



```
.box {  
  border-top-left-radius: 15px;  
  border-top-right-radius: 15px;  
  border-bottom-right-radius: 15px;  
  border-bottom-left-radius: 15px;  
}
```



border-radius



Apvalina kampus

```
.box {  
  border-radius: 4px 15px 12px 10px;  
}
```



```
border-radius: <top left> <top right> <bottom right> <bottom left>
```



border-radius



Apvalina kampus

```
.box {  
  border-radius: 50%;  
}
```



box-shadow



Suteikia šešėlį elementui

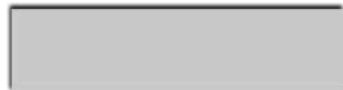
```
.box {  
  box-shadow: 1px 2px 2px #000;  
}
```



`box-shadow: <inset> <offset-x> <offset-y> <blur-radius> <spread-radius> <color>`

```
box-shadow: 1px 2px 0 2px #000;
```

```
.box {  
  box-shadow: -1px -2px 2px #000;  
}
```



text-shadow



Suteikia šešėlį tekstui

```
h1 {  
  text-shadow: 1px 2px 2px #000;  
}
```



I have a shadow!

```
text-shadow: <offset-x> <offset-y> <blur-radius> <color>
```

keli fonai tam pačiam elementui



```
.element {  
  background-image: url(bg1.png), url(bg2.png);  
  background-position: top left, center right;  
  background-repeat: no-repeat, no-repeat;  
}
```

```
.element {  
  background:  
    url(bg1.png) top left no-repeat,  
    url(bg2.png) center right no-repeat;  
}
```



opacity



```
.element {  
  color: rgba(0, 0, 0, 0.75);  
}
```

```
.element {  
  opacity: 0.45;  
}
```

```
.element {  
  background: url(bg.jpg) center no-repeat;  
  opacity: 0.45;  
}
```


linear-gradient



```
.element {  
  background: linear-gradient(to bottom, red, yellow);  
}
```

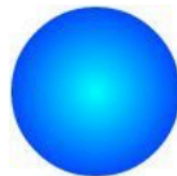


radial-gradient



```
radial-gradient(<shape> <size> at <position>, <color-stop>s)
```

```
.element {  
  background:  
    radial-gradient(aqua, blue);  
}
```



```
.element {  
  background: radial-gradient(circle at top left, aqua, blue);  
}
```





Online šriftai mūsų svetainėje

```
@font-face {  
  font-family: 'OpenSansRegular';  
  src: url('OpenSansRegular-webfont.eot');  
  font-style: normal;  
  font-weight: normal;  
}
```

```
h1 {  
  font-family: 'OpenSansRegular';  
}
```

transition



Iš vienos spalvos į kitą su animacija

```
.element {  
  background-color: black;  
  transition: background-color 0.2s ease-in-out;  
}
```

```
.element:hover {  
  background-color: blue;  
}
```

```
transition: <property> <duration> <timing-function> <delay>
```

<button>



NETEISINGA!

```
<a href="#"><button>Click Me</button></a>
```

Geriau:

```
<form action="http://diywpblog.com/" method="get">
<button>Visit Website</button>
</form>
```

```
.my_linkable_button {
  background: none repeat scroll 0 0 #075482;
  border: medium none;
  color: #FFFFFFF;
  cursor: pointer;
  margin-bottom: 26px;
  padding: 8px;
}
```

<button>



```
<button type="reset">Trinti</button>  
  
<button type="submit">Siusti</button>  
<input type="submit" value="Siusti">  
  
<button type="button">Tęsti</button>
```

Šiek tiek Javascript:



```
<button onclick="window.location='http://apple.com';">Apple Website</button>  
<button onclick="window.open('http://apple.com');">Apple Website</button>
```

CSS:

```
.button {  
    cursor: pointer;  
}
```



**Google Classroom rasite cssvariables.zip failą
- išskeiskite ir tuomet atsidarykite Brackets viduje**

**style.css faile matome, kad kartojasi spalvų deklaravimas
kintamieji leidžia optimizuoti CSS kodą:**

```
:root {  
  --red: #ff6f69;  
  --beige: #ffeedad;  
  --yellow: #ffcc5c;  
}
```

```
var(--red);  
var(--beige);  
var(--yellow);
```




Jei norime pakeisti tik konkretaus elemento spalvą, galime priskirti naują kintamojo reikšmę to elemento viduje:

```
.item {  
  --red: #ff8e69;  
  background: var(--yellow);  
}
```



Panašiai galime sukurti kintamuosius, kurie galiotų tik konkrečiame elemente ir niekur kitur. Vietoj to, kad deklaruotume juos root elemente, deklaruojame ten kur norime:

--nav-red galios tik **#navbar** viduje ir niekur kitur

Sukurkite --item-yellow ir pritaikykite vietoj geltonos spalvos

```
#navbar {  
  --nav-red: #ff6f19;  
}  
  
#navbar a {  
  color: var(--nav-red);  
}
```



Jei norime sukurti šabloną - pvz., pritaikyti keletą spalvų kodo blokui, galime sukurti naują klasę ir joje aprašyti kintamuosius.

```
<div class="item featured">  
  <h2>prekė 4</h2>  
  <button>daugiau</button>  
</div>
```

```
.featured {  
  --item-yellow: #ffe55b;  
  --red: #ff5564;  
}
```



Galime panaudoti kintamuosius ir responsive dizaine:

```
.grid {  
  --columns: 200px 200px;  
}
```

```
.grid {  
  display: grid;  
  grid-template-columns: var(--columns);  
  grid-auto-rows: 140px;  
  grid-gap: 20px;  
  justify-content: center;  
}
```

```
@media all and (max-width:650px) {  
  .grid {  
    --columns: 300px;  
  }  
}
```

**Padarykite taip, kad
ekranuose, kurie yra mažesni
nei 650px pasikeistų ir fono
spalva: #fffeed**

**CSS Grid turi daug privalumų
lyginant su framework'ais
(pvz. Bootstrap)**



Fixed Layout:

Pikseliai - absoliutūs, nekintantys dydžiai

Fluid Layout:

%, em, rem - kintantys dydžiai pagal sąlygas

Fluid Layouts



default font-size naršyklėse yra 16px

$10 / 16 = 62.5\%$

Pirmiausia, pakeičiame šį font-size, kad gautume **1em = 10px**

Formulė:

TIKSLAS / KONTEKSTAS

(Norimas dydis dalinamas iš tėvinio elemento dydžio)

```
html {  
    font-size: 16px;  
}  
  
body {  
    color: #352a25;  
    font-family: Georgia, serif;  
    font-size: 62.5%; /* 1em = 10px */  
}
```

Fluid Layouts - šrifto dydžio pavyzdys



Norime padaryti h1 dydį 30px - **tikslas**

$30 / 10 = 3em$

Naršyklės šrifto default dydį jau esame pakeitę į 10px - **kontekstas**

```
h1 {  
    font-size: 3em; /* 30px/10px */  
    font-weight: bold;  
}
```

Fluid Layouts - šrifto dydžio pavyzdys



<a> žymė esanti h1 viduje priima h1 šrifto kontekstą

```
h1 a {  
  font-size: 14px; /* konvertuojame 14/30=0.4666667em - nes h1 yra 30px */  
  text-transform: uppercase;  
  text-decoration: none;  
  color: #6C564B;  
}
```




Kai keičiame **margin, kontekstas yra tėvinio elemento plotis**

Kai keičiame elemento **padding, kontekstas yra to paties elemento plotis**



Adaptive design - žinome įrenginio ekrano dydį ir atitinkamai pritaikome savo kodą.

```
@media screen and (max-width:320px){  
  body {  
    font-size: 100%;  
  }  
}
```

Visą @media kodą rašome mūsų css failo apačioje.



“Mobile first” - verčia mus skirti dėmesį svarbiausiems dalykams, supaprastinti turinį, daugiau galvoti apie dizainą ir vartotojo patirties optimizavimą.



Fluid Layouts - iškyla problemų su mažiausiais ekranais

Adaptive Design - tinka konkrečioms tikslams, bet netinka universaliam naudojimui

Responsive Design - lankstus ir universalus



Adaptive dizaine viskas priklauso nuo ekrano pločio, kai tuo tarpu Responsive dizaine viskas priklauso nuo turinio.

Naudojame @media stilizavimui

Responsive images:

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