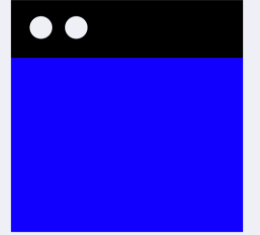




**Code
Academy**

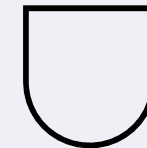


Gytis Juozenas

HTML struktūrizavimas, CSS pagrindai, tekstai, šriftai, spalvos

2022

Front-end programavimas



Šiandien išmoksime

01

HTML struktūrizavimas

02

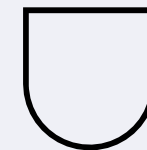
CSS pagrindai

03

CSS selektoriai

04

CSS spalvos



HTML žymos (angl. *tags*)

Kiti HTML elementai.

HTML elementai istoriškai buvo skirstomi į:

- „blokinio lygio“ (angl. *block level*) arba
- „įterptinio lygio“ (angl. *inline level*) elementus.

Palyginimas (Inline vs. block-level elements):

```
<div>The following span is an <span class="highlight">inline element</span>;  
its background has been colored to display both the beginning and end of  
the inline element's influence.</div>
```

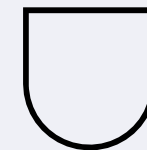
The following span is an **inline element**; its background has been colored to display both the beginning and end of the inline element's influence.

Papildomai:

HTML turinio kategorijos: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories

Block : https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements

Inline: https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements



HTML naudojimas turinio struktūrizavimui

Semantinės reikšmės neturintys HTML “įvyniotojai” (angl. *Non-semantic wrappers*).

Kartais galite susidurti su situacija, kai nerandate idealaus semantinio elemento, kad galėtumėte sugrupuoti kai kuriuos elementus arba apvynioti turinį. Kartais galite tiesiog sugrupuoti elementų rinkinį, kad paveiktumėte juos visus kaip vieną esybę su tam tikra CSS ar „JavaScript“ komanda.

Tokiais atvejais HTML pateikia elementus `<div>` ir ``. Geriausia juos naudoti su tinkamu klasės atributu, kad pateiktumėte tam tikrą etiketę, kad juos būtų galima lengvai pritaikyti.

Pvz.:

```
1 <div class="shopping-cart">
2   <h2>Shopping cart</h2>
3   <ul>
4     <li>
5       <p><a href=""><strong>Silver earrings</strong></a>: $99.95.</p>
6       
7     </li>
8     <li>
9       ...
10    </li>
11  </ul>
12  <p>Total cost: $237.89</p>
13 </div>
```



CSS pagrindai



CSS (Cascading Style Sheets)

CSS („Cascading Style Sheets“) naudojama kuriant ir išdėstant tinklalapius, pavyzdžiui, norint pakeisti turinio šriftą, spalvą, dydį ir tarpus, padalinti jį į kelis stulpelius arba pridėti animacijas ir kitas dekoratyvines funkcijas. Šis modulis suteikia švelnų kelią į CSS meistriškumą pradedant jo veikimo pagrindais, kaip atrodo sintaksė ir kaip galite pradėti jį naudoti, norėdami pridėti stilių prie HTML.

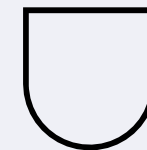


Kas yra CSS

CSS („Cascading Style Sheets“) yra kalba, nurodanti, kaip vartotojams pateikiami HTML dokumentai – kaip jie formuojami, išdėstomi ir pan.

CSS gali būti naudojamas labai paprastam dokumento teksto stiliui kurti, pavyzdžiui, pakeisti antraščių ir nuorodų spalvą ir dydį. Jis gali būti naudojamas maketui kurti – pavyzdžiui, paversti vieną teksto stulpelį maketu su pagrindine turinio sritimi ir šonine juosta, skirta susijusiai informacijai. Jis netgi gali būti naudojamas tokiems efektams kaip animacija.





CSS įtraukimas į mūsų dokumentą

Yra trys skirtingi būdai, kaip pritaikyti CSS HTML dokumente:

- Inline CSS;
- Internal CSS;
- External CSS.

Inline CSS

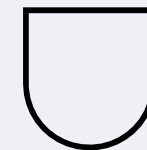
```
<p style="color: blue;">This is a paragraph.</p>
```

Internal CSS

```
<head>  
  <style type = text/css>  
    body {background-color: blue;}  
    p { color: yellow;}  
  </style>  
</head>
```

External CSS

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



CSS inline styles

Mažiausiai rekomenduojamas, beveik nenaudojamas (yra išimčių) stilių aprašymo būdas.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
  </head>
  <body>
    <h1 style="color: blue;background-color: yellow;border: 1px solid black;">Hello World!</h1>
    <p style="color:red;">This is my first CSS example</p>
  </body>
</html>
```




CSS sintaksė

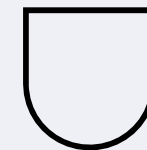
CSS yra taisyklėmis paremta kalba – jūs apibrėžiate taisykles, nurodančias stilių grupes, kurios turėtų būti taikomos tam tikriems jūsų tinklalapio elementams ar elementų grupėms. Pavyzdžiui, „noriu, kad pagrindinė mano puslapio antraštė būtų rodoma kaip didelis raudonas tekstas“.

Šis kodas rodo labai paprastą CSS taisyklę, kuri leistų pasiekti aukščiau aprašytą stilių:

```
h1 {  
  color: red;  
  font-size: 5em;  
}
```

CSS sintaksės struktūra:





CSS internal styles

Techniškai niekada nenaudojamas nebent atveju kai dirbama su kokia CMS (Content Management System) kai jums yra blokuojamas CSS failų editinimas.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
    <style>
      h1 {
        color: blue;
        background-color: yellow;
        border: 1px solid black;
      }

      p {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```



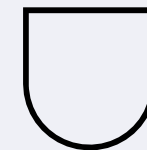
CSS external stylesheet

Pagrindinis CSS style aprašymo metodas kuriam yra kuriamas atskiras CSS failas.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```

```
h1 {
  color: blue;
  background-color: yellow;
  border: 1px solid black;
}

p {
  color: red;
}
```



CSS spalvos (CSS Colors)

Spalvos nurodomos naudojant iš anksto nustatytus spalvų pavadinimus arba RGB, HEX, HSL, RGBA, HSLA reikšmes.

- [CSS spalvų pavadinimai](#) – CSS spalvą galima nurodyti naudojant iš anksto nustatytą spalvų pavadinimą (pvz.: red, green ir pan.);
- [CSS RGB spalvos](#) – CSS spalva gali būti nurodyta kaip RGB reikšmė, naudojant šią formulę: *rgb (red, green, blue)* arba *rgba (red, green, blue, alpha)*;
- CSS HEX vertė – CSS spalvą galima nurodyti naudojant šešioliktinę reikšmę formoje: #rrggbb (kur rr (raudona), gg (žalia) ir bb (mėlyna) yra šešioliktinės vertės tarp 00 ir ff);
- [CSS HSL vertė](#) CSS spalvą galima nurodyti naudojant atspalvį, sodrumą ir šviesumą (HSL), pateikiant tokią formą: *hsl (hue, saturation, lightness)*.





Užduotis nr. 1

Susikurkite 4 paragrafus ir nuspalvinkite tekstą per inline CSS skirtingomis spalvomis naudojantis:

- CSS spalvų pavadinimais (pvz.: red, orange, green ir pan.);
- CSS RGB spalvomis *rgb* (red, green, blue) arba *rgba* (red, green, blue, alpha);
- CSS HEX vertėmis #rrggbb (kur rr (raudona), gg (žalia) ir bb (mėlyna) yra šešioliktinės vertės tarp 00 ir ff);
- CSS HSL vertėmis *hsl* (hue, saturation, lightness).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
  </head>
  <body>
    <h1 style="color: blue;background-color: yellow;border: 1px solid black;">Hello World!</h1>
    <p style="color:red;">This is my first CSS example</p>
  </body>
</html>
```

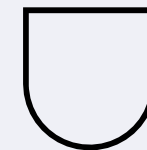


CSS selektoriai

CSS selektoriai naudojami norint „surasti“ (arba pasirinkti) HTML elementus, kuriuos norite stilizuoti.

CSS selektorius galime suskirstyti į penkias kategorijas:

- Paprasti selektoriai (pasirinkite elementus pagal pavadinimą, id, class);
- Kombinatoriaus selektoriai (elementus pasirinkite pagal konkretų tarpusavio ryšį);
- Pseudo klasės selektoriai (pasirinkite elementus pagal tam tikrą būseną);
- Pseudo elementų selektoriai (pasirinkite ir stiliaus elemento dalis);
- Atributų parinkikliai (elementus pasirinkite pagal atributą arba atributo vertę).



CSS selektoriai

CSS elemento parinkiklis – Elementų parinkiklis pasirenka HTML elementus pagal elemento pavadinimą.

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

CSS id parinkiklis – id parinkiklis naudoja HTML elemento atributą tam tikram elementui pasirinkti.

```
#paragrap1 {  
  text-align: center;  
  color: red;  
}
```

CSS class parinkiklis – class parinkiklis parenka HTML elementus su konkrečiu klasės atributu.

```
.center {  
  text-align: center;  
  color: red;  
}
```

```
selector {  
  property: value;  
}
```



ID: #344-34-4344

Class: Male

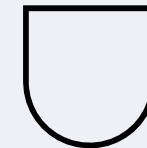
Class: Employee



ID: #123-54-9877

Class: Female

Class: Employee



CSS selektoriai

CSS nesting parinkiklis parenka elemente/klasėje/id esantį kitą elementą/klasę/id.

```
p .center {  
  text-align: center;  
  color: blue;  
}
```

CSS grupavimo parinkiklis – grupavimo parinkiklis parenka visus HTML elementus su tais pačiais stiliaus apibrėžimais.

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

Taip pat galime nurodyti, kad klasė turėtų paveikti tik tam tikrus HTML elementus.

```
p.center {  
  text-align: center;  
  color: red;  
}
```

PASTABA: HTML elementas gali turėti daugiau negu vieną klasę.



CSS selektoriai

Koks stilius bus naudojamas, kai HTML elementui yra nurodytas daugiau nei vienas stilius?

Visi puslapio stiliai bus „pakopiniai“ į naują „virtualų“ stiliaus lapą pagal šias taisykles, kur numeris vienas turi aukščiausią prioritetą:

1. Įterptasis stilius (HTML elemento viduje) (Inline style);
2. Išoriniai ir vidiniai stiliaus lapai (head dalyje) (External and internal style sheets);
3. Naršyklės numatytasis nustatymas.



CSS class bei ID pavadinimų taisyklės



snake_case

Pros: Concise when it consists of a few words.
Cons: Redundant as hell when it gets longer.
`push_something_to_first_queue, pop_what, get_whatever...`



PascalCase

Pros: Seems neat.
`GetItem, SetItem, Convert, ...`
Cons: Barely used. (why?)



camelCase

Pros: Widely used in the programmer community.
Cons: Looks ugly when a few methods are n-worded.
`push, reserve, beginBuilding, ...`



skewer-case

Pros: Easy to type.
`easier-than-capitals, easier-than-underscore, ...`
Cons: Any sane language freaks out when you try it.



SCREAMING_SNAKE_CASE

Pros: Can demonstrate your anger with text.
Cons: Makes your eyes deaf.
`LOOK_AT_THIS, LOOK_AT_THAT ...`



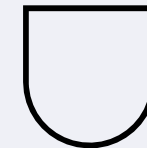
nocase

Pros: Looks professional.
Cons: Misleading af.
`supersexyhippohalamus, bool ...`



fUcKtHeCaSe

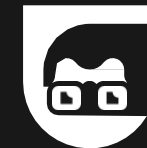
Pros: Can live outside of the law.
Cons: Can be out of a job.



CSS plotis, aukštis bei fono spalva

Plotis ir aukštis per CSS yra nurodomas atskiruose property ir kol kas jį matuosime pixeliais. O foninė spalva tiesiog naudojant viena iš kelių spalvų aprašymų būdų:

- [width](#) – CSS plotį galima nurodyti naudojantis pixeliais (ir kitais matmenimis, tačiau juos mokysimės vėliau);
- [height](#) – CSS aukštį galima nurodyti naudojantis pixeliais (ir kitais matmenimis, tačiau juos mokysimės vėliau);
- [background-color](#) – CSS fono spalvą galima nurodyti naudojant bet kurią spalvai aprašyti išraišką.



Užduotis nr. 2

Sukurkite tris bet kokios šalies vėliavas su pavadinimais virš jų, naudokites klasėmis, ID, nestinkit selektorius, kaip tik norit, bet svarbiausia kad būtų external stylesheet CSS'ui aprašyti.

Lietuva



Vokietija



Latvija

