

CSS design guidelines

Rokas Lengvenis

rokas@producters.com

Talk about

- General selector patterns
- Architectural patterns
- CV.It project

CSS selectors

Rules

- Selector intent
- Think about reusability
- Decrease Nesting

CSS selectors

- Select what you want explicitly

```
.site-nav {  
}
```

```
.header ul {  
}
```

- Very greedy
- Unexpected side effects, tangled CSS
- CSS bloat

CSS selectors

- Write selectors for reusability

```
.btn-green {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  background: green;  
  font-size: 16px;  
  text-align: center;  
  color: #FFF;  
}
```

```
div div ul > li a {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  background: green;  
  font-size: 16px;  
  text-align: center;  
  color: #FFF;  
}
```

- Modular approach
- Reduced repetition

CSS selectors

- Do not nest selectors unnecessarily
- Keep selectors as short as possible

```
.btn-green {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  background: green;  
  font-size: 16px;  
  text-align: center;  
  color: #FFF;  
}
```

```
div div ul > li a {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  background: green;  
  font-size: 16px;  
  text-align: center;  
  color: #FFF;  
}
```

- Increased specificity, lower reusability
- Slower performance (minor impact). Killer selector `ul * {}` :)

CSS architecture patterns

- **SRP** single responsibility principle
- **OCP** extending, not overriding
- **SoC** separation of concerns
- **DRY** don't repeat yourself

CSS architecture patterns

- **SRP** single responsibility principle

```
.btn {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  font-size: 16px;  
  text-align: center;  
}
```

```
.btn.large {  
  float: left;  
  padding: 10px 20px;  
  font-size: 20px;  
}
```


CSS architecture patterns

- **OCP** extending not overriding

```
.btn {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  font-size: 16px;  
  background: #FFF;  
  text-align: center;  
}
```

```
.btn.large {  
  padding: 10px 20px;  
  font-size: 20px;  
}  
  
.btn.uppercase {  
  text-transform: uppercase;  
}
```

```
.btn {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  font-size: 16px;  
  text-align: center;  
}
```

```
.header .btn {  
  padding: 10px 20px;  
  font-size: 20px;  
}
```

CSS architecture patterns

- **SoC** separation of concerns

```
.control-label {  
  padding-top: 7px;  
  padding-right: 0;  
  text-align: right;  
  font-size: 14px;  
  line-height: 19px;  
  color: #6c6c6c;  
}
```

```
.form .control-label {  
  margin-top: 5px;  
}
```

CSS architecture patterns

- **DRY** don't repeat yourself
 - Meaningful repetition only
 - `@extend .class;`
 - `@extend %placeholder;`
 - mixins: `@include border-radius(3px);`

CSS architecture methods

- **Atomic CSS**
- **BEM**
- **Atomic design**
- **OOCSS**

CSS architecture methods

- **Atomic CSS**

```
div.panel-rounded-grey.container
  div.mt-19.mb-70.box-center.panel-dialog-box.new-advertisement
    a.btn.btn-close(href='#')
    div.ml-37.mb-21.pt-20.heading
      i.mr-15.icon.icon-plus-rounded-green
      h1 Skelbimas savanoriams
    form.ml-19.pb-55.w-750.form.advertisement
```

- No need to write stuff into css file
- Narrow scope, not afraid of breaking something else
- Less bloat
- Little maintenance on CSS side
- Less abstraction

CSS architecture methods

- **Atomic CSS**

Difference from inline styles

- class specificity, can be overridden
- inline styles bloats your html file, atomic classes are small concise, reusable

Disadvantages

- responsive designs
- less abstraction on html side

CSS architecture methods

- **BEM**

```
.btn {} - block  
.btn__icon {} - element  
.btn--cancel {} - modifier
```

Why?

- You can override class by accident
- Overhead while naming
- Less nesting

CSS architecture methods

- **Atomic Design**

- **Atoms** (An HTML element, e.g. an input)
- **Molecules** (a search for example including a label, input and a button)
- **Organisms** (header of a site, navigation etc)
- **Templates** (A wireframe of the whole site + layout)
- **Pages** (Exact page)

Why?

- Provides a modular approach
- Complies to SoC, SRP, DRY
- Less nesting

CSS architecture methods

- **OOCSS**
 - **Keep the structure and skin separate**
 - **Separate container and content**

```
.btn-green {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  background: green;  
  font-size: 16px;  
  text-align: center;  
  color: #FFF;  
}
```

```
div div ul > li a {  
  display: inline-block;  
  border: 1px solid #FFF;  
  padding: 5px 10px;  
  background: green;  
  font-size: 16px;  
  text-align: center;  
  color: #FFF;  
}
```

CV.It

- **Challenges**
 - **Rich UI components**
 - **Pixel perfect**
 - **Multifarious design**

CV.It

- **Technology stack used**
 - Jade
 - Scss
 - grunt-spritesmith for img sprites
 - Grunt/Bower/Yeoman

CV.lt

- Jade

```
div.panel-rounded-grey.container
  div.mt-19.mb-70.box-center.panel-dialog-box
    a.btn.btn-close(href='#')
    div.ml-37.mb-44.pt-20.heading
      i.mr-15.icon.icon-plus-rounded-green
      h1 Naujas skelbimas
    form.pb-55.box-center.w-342.form.large
      div.form-group
        div.col-sm-12
          select(id='category', name='category')
            option(value='1') Darbuotojų paieškos skelbimas
            option(value='2') Kitoks skelbimas
            option(value='3') Toks pat skelbimas
        div.form-group
          div.col-sm-12
            select(id='language', name='language')
              option(value='1') Lietuvių k.
              option(value='2') Anglų k.
              option(value='3') Kinų k.
        div.form-group
          div.col-sm-12
            select(id='type', name='type')
              option(value='1') Įprasta forma
              option(value='2') Neįprasta forma
              option(value='3') Per prasta forma
      div.mt-60.form-group
        div.col-sm-12.text-center
          button.btn.btn-green.large.uppercase(type='submit')
          span Pradėti skelbimą
```

CV.It

- **Scss**

```
.form-control {  
  @extend %border-style;  
  display: block;  
  width: 100%;  
  padding: 6px;  
  font-size: 14px;  
  line-height: 19px;  
}  
  
.checkbox, .radio {...}  
  
// Selectize styling  
.selectize-control.single {  
  .selectize-input {  
    @extend .form-control;  
  }  
}  
  
.selectize-dropdown [data-selectable] {  
  font-size: 14px;  
  line-height: 20px;  
}
```

CV.It

- **grunt-spritesmith**

```
.icon {  
  display: inline-block;  
}  
  
.icon-cvlt-logo {  
  @include sprite($logo);  
  &:hover {  
    @include sprite($logo-hover);  
  }  
}  
  
.icon-vz-logo {  
  @include sprite($vz-logo);  
}  
  
.icon-caret-down-white {  
  @include sprite($caret-down-white);  
}
```

CV.It

Architectural solutions

- Defined elements as separate ***modular components***
- Use ***atomic CSS*** for handling pixel perfect issues and diverse layouts
- Use “***extend by class***” to conquer diversity

CV.It

Designing components

- Subcomponent naming convention .name-subname
- Actions goes to separate declaration
- Extension by @extend if see no reason polluting html with extended class names

```
/* Derived tables (layouts + skin combined)
```

```
=====
.table.tax-results {
  @extend .tax-results-layout;
  @extend .tax-results-skin;
}

.table.advertisements {
  @extend .header-md;
  @extend .header-grey;
  @extend .layout-advertisements;
}

.table.users {...}

.table.letters {...}

.table.answers {...}
```

```
/* Widget Richtext
```

```
=====
.widget-ad-upload {
  .ad-upload-image {
    position: relative;
    width: 287px;
    line-height: 0;
    border: 1px solid #d6d6d6;
    img {
      width: 100%;
    }
    .ad-upload-controls {
      position: absolute;
      top: -1px;
      right: -47px;
    }
  }
}
```

```
//Actions
```

```
.widget-ad-upload {
  .ad-upload-thumbnail {
    display: none;
  }
  &.empty {
    .ad-upload-image {
      display: none;
    }
    .ad-upload-thumbnail {
      display: block;
    }
  }
}
```


CV.It

Atomic CSS for layout

- If layout will be reusable and is complex, put it to css file
- If layout will not be reused, a small change needed use helper
- If layout is for component (ex table), put it under class

```
h1 Naujas vartotojas
form.ml-19.pb-55.w-90p
div.form-group
  label.col-sm-4.control-label(for='name') Vardas
  div.col-sm-5
    input.form-control(id='name', type='text', placeholder='Vardas')
div.form-group
  label.col-sm-4.control-label(for='surname') Pavardė
  div.col-sm-5
    input.form-control(id='surname', type='text', placeholder='Pavardė')
div.form-group
  label.col-sm-4.control-label(for='position') Pareigos
  div.col-sm-5
    input.form-control(id='position', type='text', placeholder='Pareigos')
div.mt-40.form-group
  label.col-sm-4.control-label(for='phone') Telefonas
  div.col-sm-5
    input.form-control(id='phone', type='text')
```

Vardas	<input type="text" value="Vardas"/>
Pavardė	<input type="text" value="Pavardė"/>
Pareigos	<input type="text" value="Pareigos"/>
Telefonas	<input type="text"/>
El. paštas	<input type="text"/>

CV.It

Extend by class

- If layout will be reusable and is complex, put it to css file
- If layout will not be reused, a small change needed use helper
- If layout is for component (ex table), put it under class

```
table.mb-26.table.users
  <thead>
    <tr>
      <th> Vartotojas
      <th> Pareigos
      <th> Telefonas
      <th> El. paštas
    </tr>
  </thead>
  <tbody>
  </tbody>
</table>
```

```
<button>btn.btn-green.large.uppercase(type='submit')
  <i>fa.fa-check
  <span>Išsaugoti
</button>
<button>ml-5.btn.btn-grey.large.uppercase(type='reset')
  <span>Atšaukti
</button>
```

CV.It

Full example

```
div#user-nav
  div.container
    ul.nav.nav-site
      li.nav-item.link-grey-darker.dropdown.active
        a(href='#') Mano skelbimai
        i.fa.fa-chevron-down
        ul.dropdown-menu.nav-menu
          li.menu-item: a.dark-blue(href='#') Naujas skelbimas
          li.menu-item: a.dark-blue(href='#') Skelbimai
          li.menu-item: a.dark-blue(href='#') Šablonai
      li.nav-item.link-grey-darker
        a(href='#') Mano kandidatai
      li.nav-item.link-grey-darker.dropdown
        a(href='#') Įrankiai
        i.fa.fa-chevron-down
        ul.dropdown-menu.nav-menu
          li.menu-item: a.dark-blue(href='#') Naujas skelbimas
          li.menu-item: a.dark-blue(href='#') Skelbimai
          li.menu-item: a.dark-blue(href='#') Šablonai
      li.nav-item.link-grey-darker
        a(href='#') Profilis
      li.nav-item.link-grey-darker
        a(href='#') CV paieška
```

CV.It

Final remarks

- Reusability vs efforts
- Extend @ vs %
- Create ui library

Q&A ?