

školení frontend
Organizace CSS

organizace CSS

Nastavení editoru

- **Editor Config**

- <http://EditorConfig.org>
- zajistí stejné formátování kódu

- **Prettier**

- <https://prettier.io/>
- hlídá řádky mezi selektory, čárky v hodnotách, 0 na začátku desetinných hodnot

- **Stylelint**

- <https://stylelint.io/>
- hlídá specificitu, zanoření, BEM pravidla, zápis barev, jednotky, ...
- plugin do většiny editorů – VS Code, Sublime, Atom, ...

<https://css-tricks.com/prettier-stylelint-writing-clean-css-keeping-clean-code-two-tool-game/>

organizace CSS

Editor Config

```
[*]  
end_of_line = lf  
indent_size = 2  
indent_style = space  
charset = utf-8  
trim_trailing_whitespace = true  
insert_final_newline = true
```

```
[package.json]  
indent_style = space  
indent_size = 2
```

...

organizace CSS

Prettier

Upozornění a oprava obecných chyb v kódu

- automatická oprava kódu
- možnost spouštět při každém uložení souboru
- integrovaný v editorech nebo jako Gulp/Grunt plugin
- opravy odsazení, destinných míst, velikosti písem, uvozovky, délka řádku, ...

organizace CSS

Stylelint

Upozornění na chyby v CSS

- zapomenuté prázdné deklarace nebo komentáře
- překlepy v názvech vlastností
- zakázání pojmenovaných barev
- omezení počtu desetinných míst
- velikost znaků v zápisu barev
- ...

organizace CSS

Stylelint

```
"extends": "stylelint-config-standard",  
"rules": {  
  "at-rule-empty-line-before": null,  
  "at-rule-name-space-after": null,  
  "at-rule-no-unknown": null,  
  "color-hex-case": "lower",  
  "color-named": "never",  
  ...
```

Konfigurace:

<https://maximgatilin.github.io/stylelint-config/>

organizace CSS

Demo

- ukázka stylelint & prettier v akci

CSS

Metodiky zápisu CSS

- **OOCSS**

- <http://EditorConfig.org>
- jak psát komponenty v CSS

- **BEM**



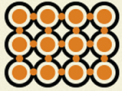


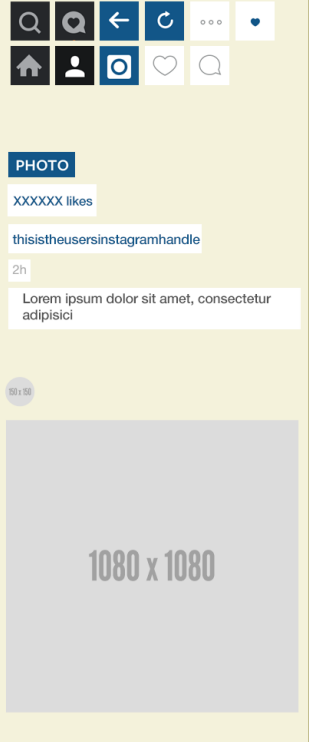

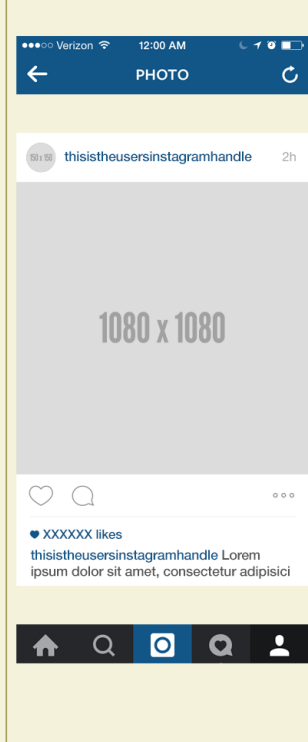
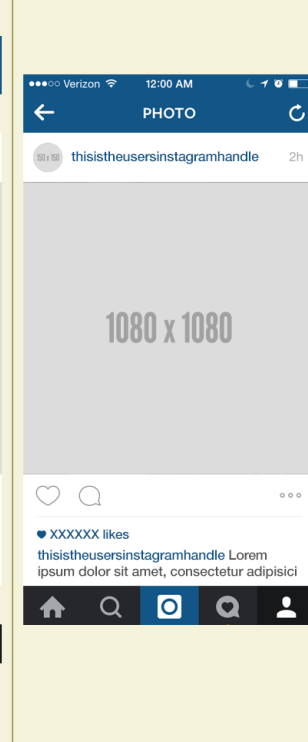
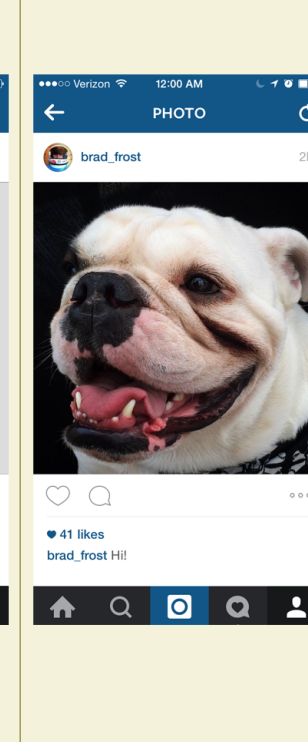

- <http://getbem.com/>
- jak pojmenovávat komponenty v CSS

- **SMACSS, ITCSS**

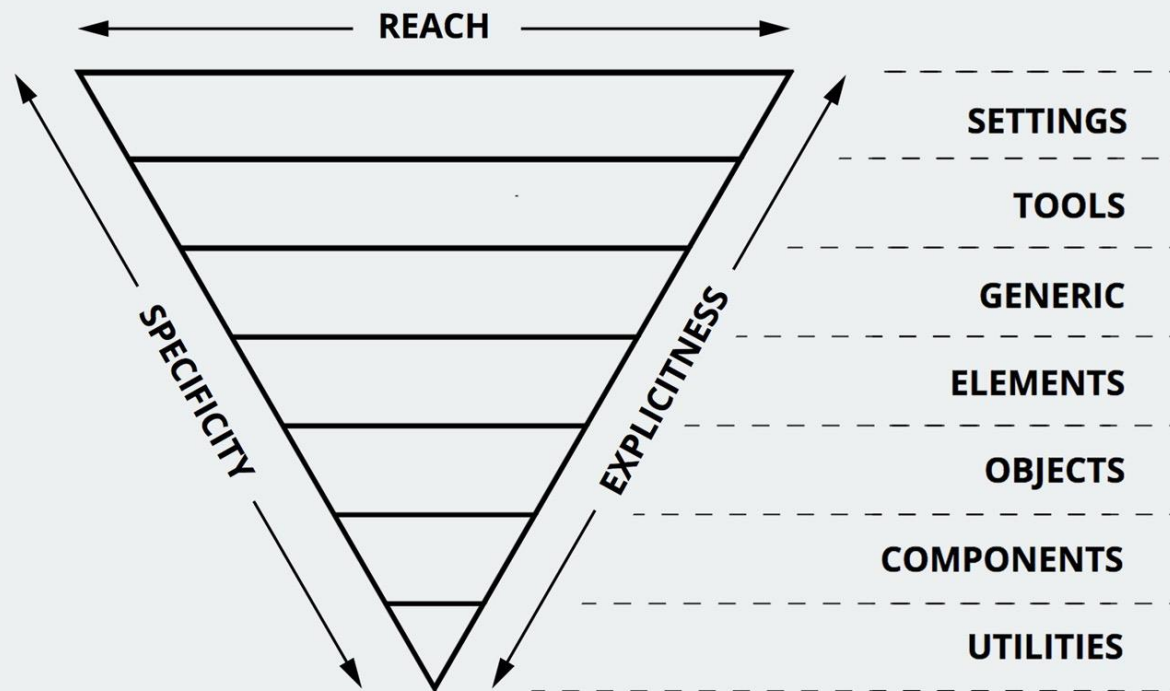
- <https://smacss.com/>
- metodiky organizace celého CSS v rámci projektu

CSS Metodiky

Atomic webdesign

 <p>ATOMS</p>	 <p>MOLECULES</p>	 <p>ORGANISMS</p>	 <p>TEMPLATES</p>	 <p>PAGES</p>
 <p>PHOTO</p> <p>XXXXXX likes</p> <p>thisistheusersinstagramhandle</p> <p>2h</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing</p> <p>1080 x 1080</p>	 <p>PHOTO</p> <p>thisistheusersinstagramhandle 2h</p> <p>1080 x 1080</p> <p>XXXXXX likes</p> <p>thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipiscing</p> <p>1080 x 1080</p>	 <p>PHOTO</p> <p>thisistheusersinstagramhandle 2h</p> <p>1080 x 1080</p> <p>XXXXXX likes</p> <p>thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipiscing</p> <p>1080 x 1080</p>	 <p>PHOTO</p> <p>thisistheusersinstagramhandle 2h</p> <p>1080 x 1080</p> <p>XXXXXX likes</p> <p>thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipiscing</p> <p>1080 x 1080</p>	 <p>PHOTO</p> <p>brad_frost 2h</p>  <p>41 likes</p> <p>brad_frost Hi!</p> <p>1080 x 1080</p>

Define an Architecture (ITCSS)



CSS Metodiky

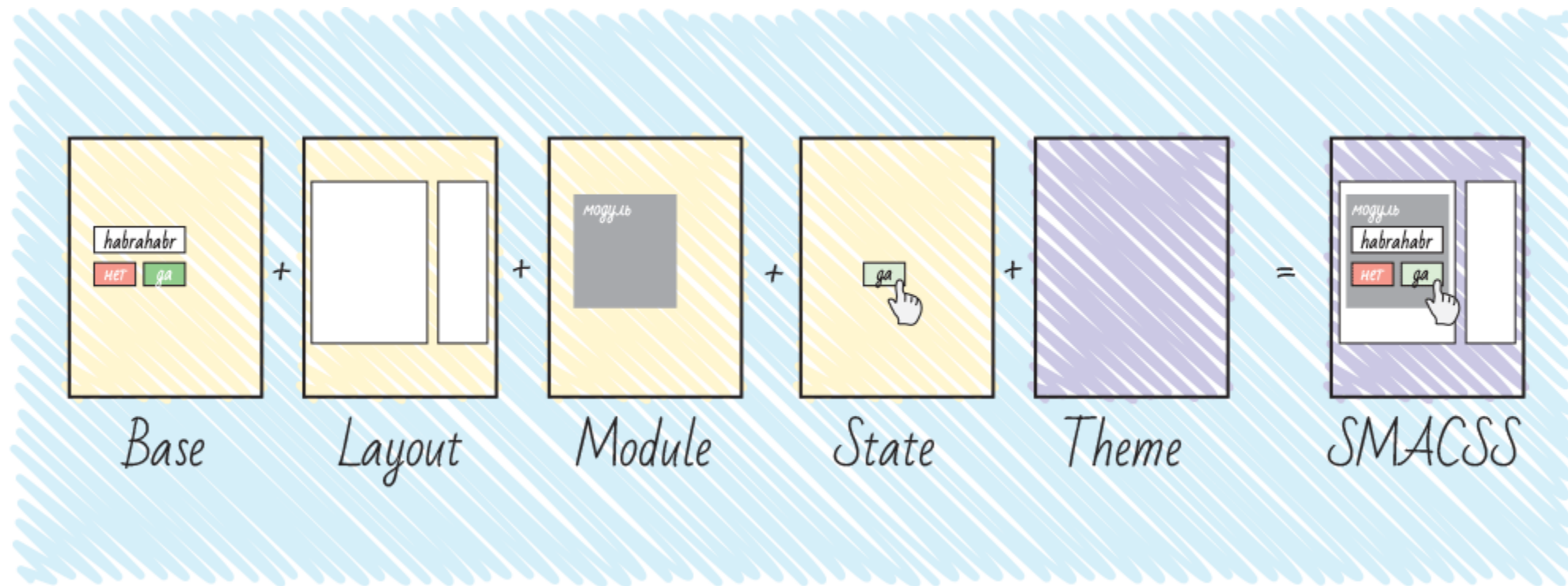
ITCSS

ITCSS	co tam patří	příklady css tříd
Settings	<i>proměnné</i>	\$bg-color, \$font-base, ...
Tools	<i>funkce, mixiny</i>	@function calc-spacing ...
Generic	<i>reset (Reboot, Normalize)</i>	body, input, ...
Elements	<i>základní nastavení layoutu</i>	body, input, h1, p, ...
Objects	<i>základní komponenty</i>	.media, .button,
Components	<i>ostatní komponenty</i>	.article-item, .list, .footer-nav, ...
Utilities	<i>pomocné třídy</i>	.img-fluid, .text-center, ...

CSS Metodiky

SMACSS

Scalable and Modular Architecture for CSS



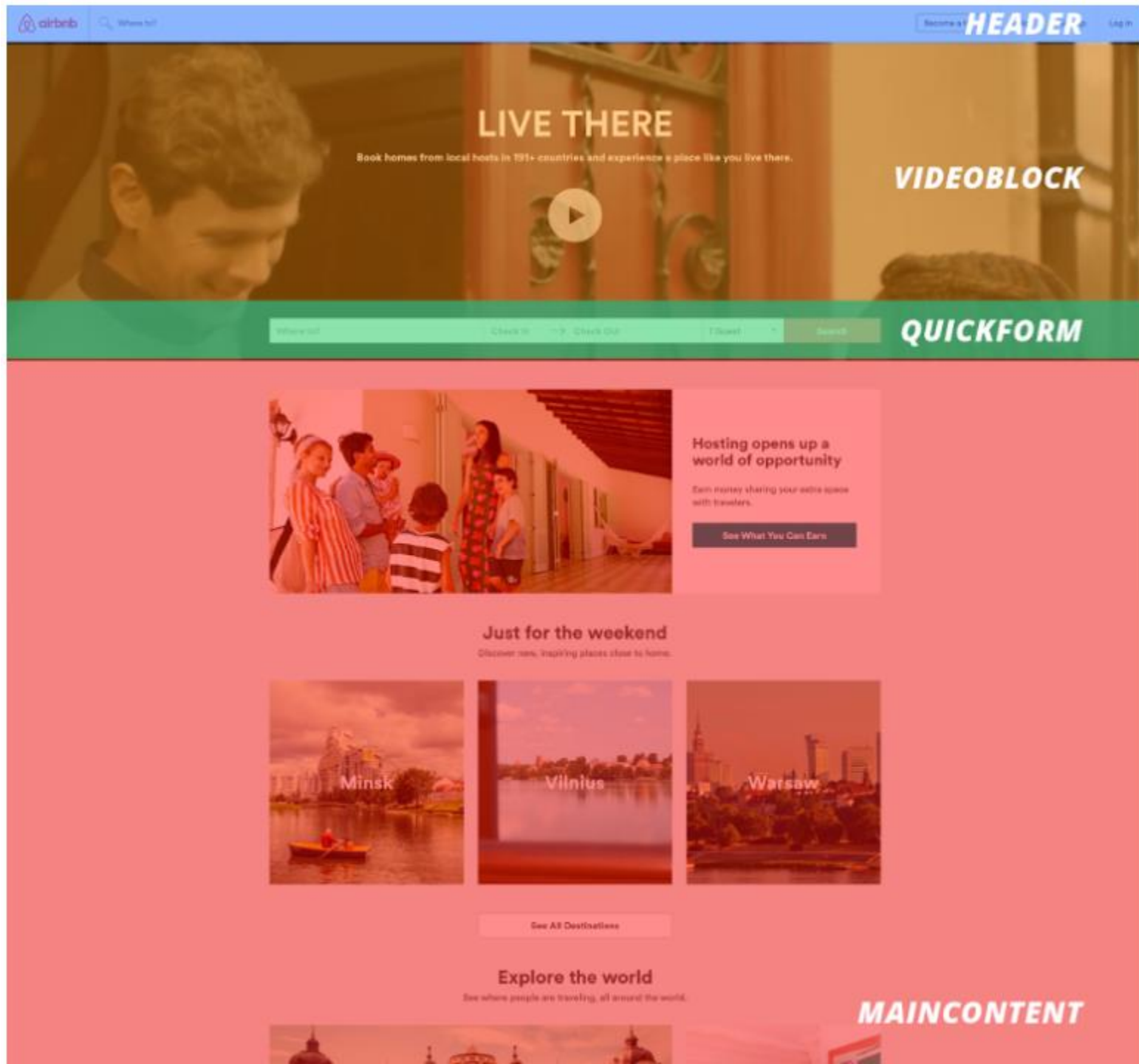
CSS Metodiky

SMACSS

SMACSS	co tam patří	příklady css tříd
Base	<i>normalizace, základní styly</i>	body, input, h1, p, ...
Layout	<i>grid, responsivní fw, wrappery</i>	.row, .container, .header, ...
Modules	<i>komponenty</i>	.article-list, .header-nav, ...
State	<i>stavy</i>	.active, .focus, ...

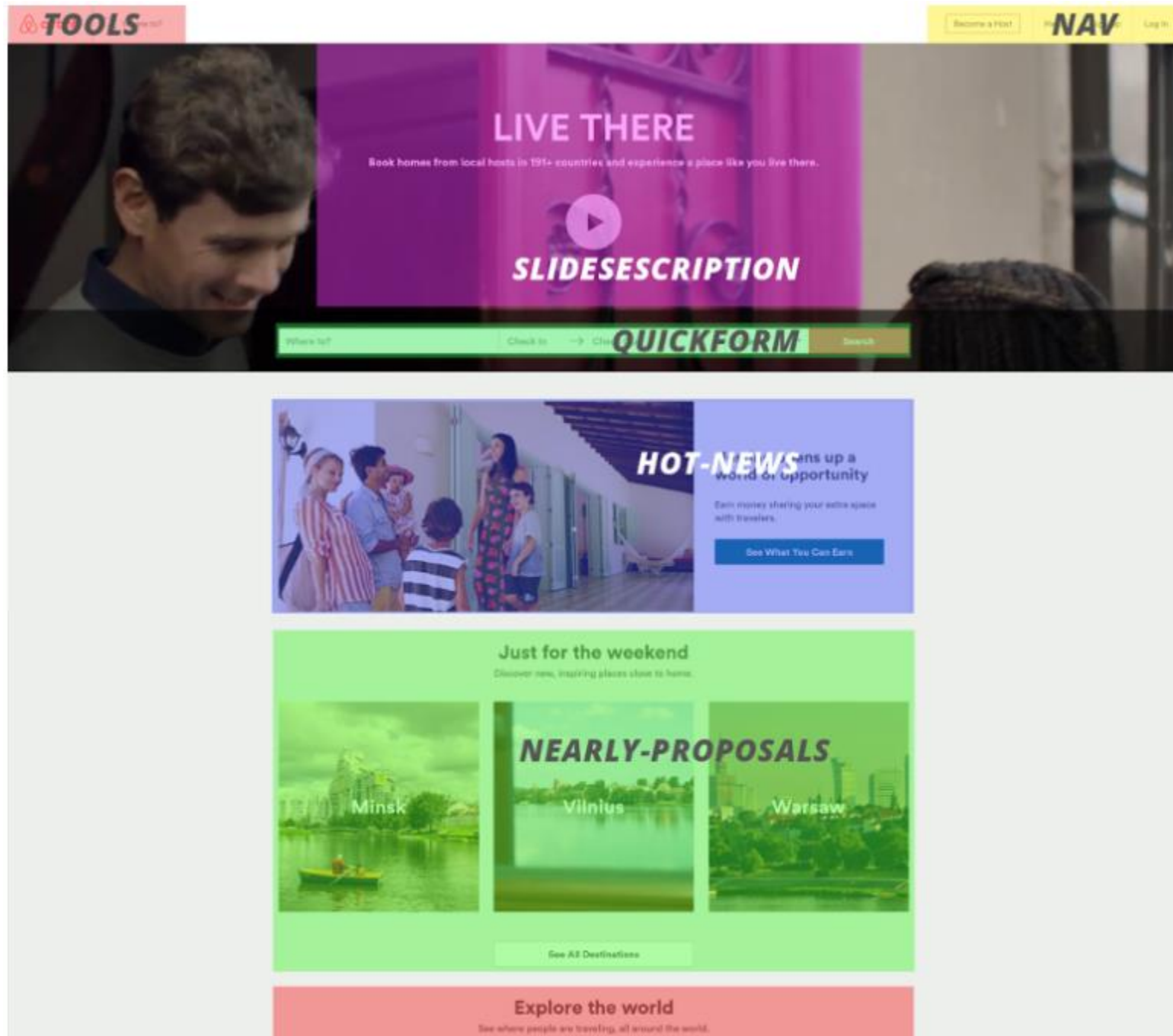
CSS Metodiky

SMACSS



CSS Metodiky

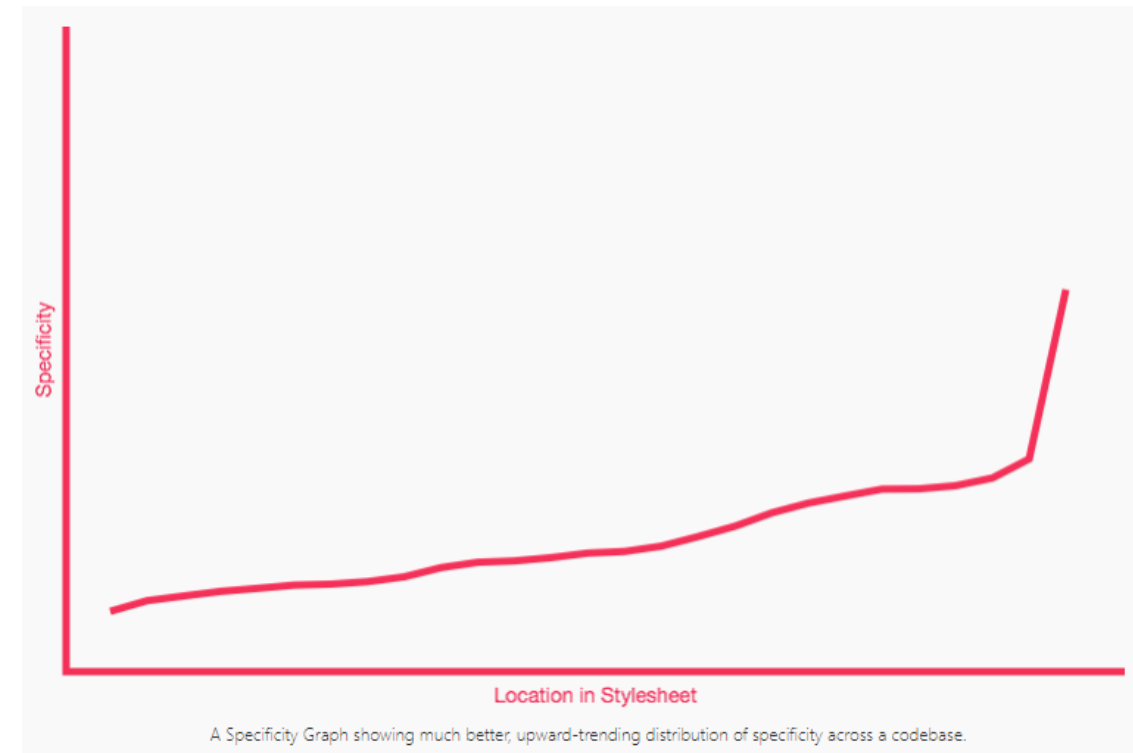
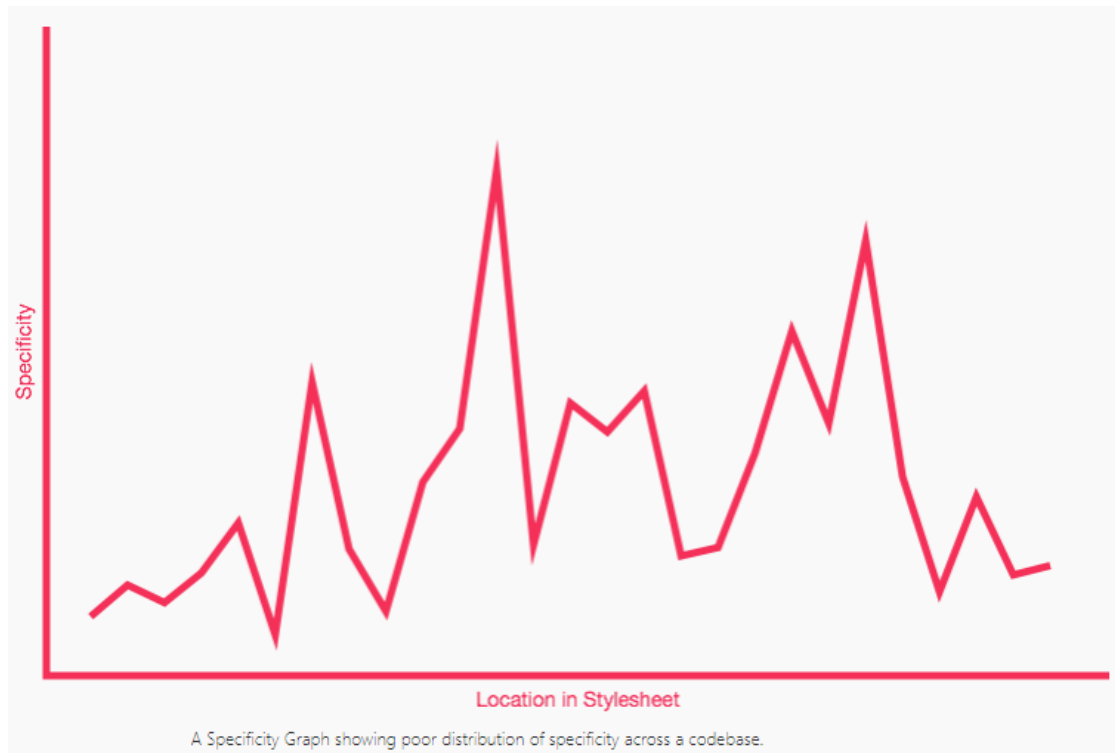
SMACSS



organizace CSS

Rozložení specificity

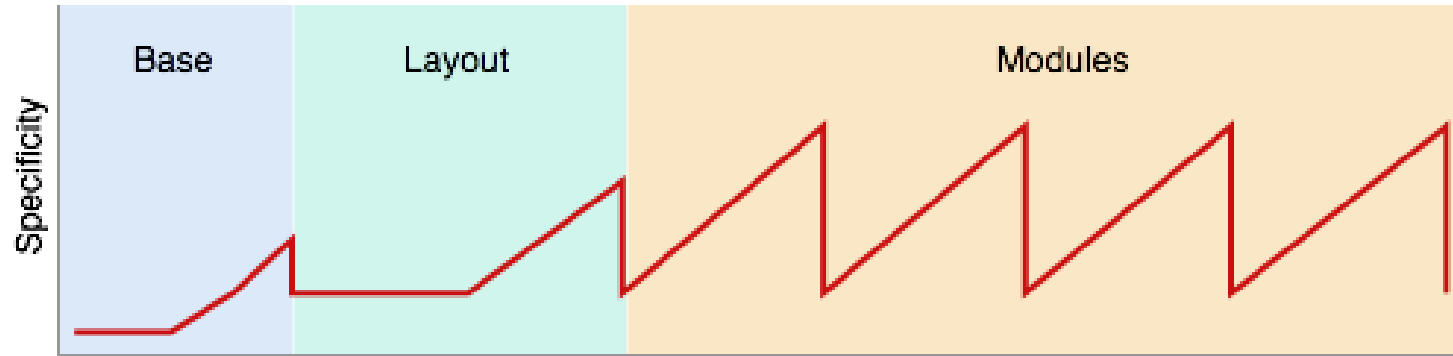
<https://cssstats.com/>



<https://csswizardry.com/2014/10/the-specificity-graph/>

organizace CSS

Rozložení specificity



H1, p, a, ... | .header, .grid, ... | .media-article, .pagination, ...

organizace CSS

Refaktoring - Struktura

zkuste opravit CSS strukturu projektu:

<https://codepen.io/svobodalukas/pen/ePybxa>

CSS Metodiky

Object oriented CSS

Cíle OOCSS

1. Zrychlení psaní kódu i načtení CSS souboru
2. Škálovatelnost
3. Efektivita
4. Snazší údržba
5. Lepší čitelnost
6. Příbuznost s ostatními koncepty zápisu CSS

Nevýhody

1. Roste počet tříd u elementu
2. Může být zbytečné u menších projektů
3. Chvilku trvá, než se dostane do krve

CSS Metodiky

OOCSS

1. Oddělení CSS od struktury

- nikdy nepoužíváme HTML tagy jako selektory (kromě základní typografie)

2. Nezávislost na kontextu

`.footer .nav { ... }` → `.nav-footer { ... }`

3. Snaha o minimální specifičnost

- nepoužíváme id selektory, !important nebo kombinace selektorů
- `.nav.nav-footer { ... }` → `.nav-footer { ... }`

CSS Metodiky

OOCSS

```
<a href="#" class="button-blue-small">Click me!</a>
<style>
  .button-blue-small {
    display: inline-block;
    text-align: center;
    margin: 10px 5px;
    text-decoration: none;
    color: #FFF;
    border: 1px solid #0082BE;
    background: #00A4EF;
    font-size: 13px;
    padding: 5px 20px;
  }
</style>
```

CSS Metodiky

OOCSS

1. **Struktura**
2. **Velikost**
3. **Styly**

CSS Metodiky

OOCSS – základní struktura

```
.button {  
  display: inline-block;  
  text-align: center;  
  margin: 10px 5px;  
  text-decoration: none;  
}
```

CSS Metodiky

OOCSS – velikost

```
.button-small {  
  font-size: 13px;  
  padding: 6px 20px;  
}
```


CSS Metodiky

OOCSS – styly

```
.button-blue {  
  color: #FFF;  
  border: 1px solid #0082BE;  
  background: #00A4EF;  
}
```

CSS Metodiky

OOCSS

```
<a href="#" class="button button-small button-blue">Click me!</a>
```

```
<style>  
  .button { ... }  
  .button-small { ... }  
  .button-blue { ... }  
</style>
```

CSS Metodiky

OOCSS - problémy

```
<div class="article article-list">  
  <div class="article-header">  
    ...  
  </div>  
</div>
```

CSS Metodiky

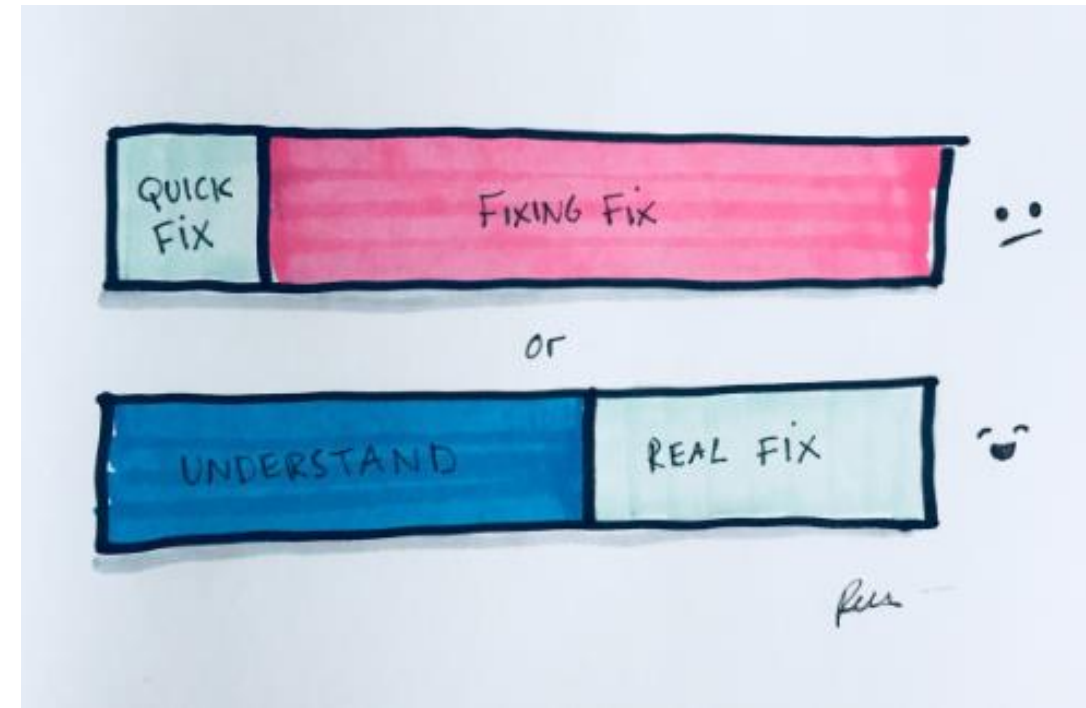
komponenta versus kontext

<https://codepen.io/machal/pen/JmdRaa>

organizace CSS

Refactoring

- Začít malými často používanými částmi
- Atomizace komponent
`.article .article-main h3` → `.article-heading`
- Izolovat nový kód od ostatního obsahu
- Kusy "šeredného" kódu sdružovat do jednoho místa (*shame.css*)



organizace CSS

Refaktoring - Komponenta

refaktorujte CSS komponentu:

<https://codepen.io/svobodalukas/pen/oapmWw>

CSS Metodiky

Block – Element – Modifier

Výhody

1. "Nadstavba" OOCSS (platí všechny podmínky OOCSS)
2. Jedná se především o metodiku jak pojmenovávat komponenty
3. Jednoduše lze poznat význam třídy
4. Snadno pochopitelný

Nevýhody

1. Hledání vhodného jména
2. Nezvyklý a často zdlouhavý zápis tříd

Block – Element – Modifier

<http://getbem.com/>

Blok `.nav-main {}`

- Nezávislá část stránky, znovupoužitelná
- Bloky lze do sebe zanořit

Element `.nav-main__list {}`

- Prvek uvnitř bloku, nefunguje bez bloku

Modifikátor `.nav-main--dark {}`

- Varianta bloku nebo elementu
- Mění vzhled, chování nebo stav

CSS Metodiky

BEM - modifikátory

```
<!-- Takhle ne -->  
<figure class="photo">  
    
  <figcaption>Look at me!</figcaption>  
</figure>
```

```
<style>  
  .photo { }  
  .photo img { }  
  .photo figcaption { }  
</style>
```

CSS Metodiky

BEM - modifikatory

```
<!-- Takhle ano -->
<figure class="photo">
  
  <figcaption class="photo__caption">Look at me!</figcaption>
</figure>

<style>
  .photo { }
  .photo__img { }
  .photo__caption { }
</style>
```

CSS Metodiky

BEM - modifikátory

```
<a href="#" class="button button--small button--blue">  
  Click me!  
</a>
```

```
<style>  
  .button { ... }  
  .button--small { ... }  
  .button--blue { ... }  
</style>
```

CSS Metodiky

BEM - modifikatory

```
<!-- Takhle ne -->
```

```
<figure class="photo">  
    
  <figcaption class="photo__caption">  
    <blockquote  
      class="photo__caption__quote">  
      Look at me!  
    </blockquote>  
  </figcaption>  
</figure>
```

```
<style>  
.photo { }  
.photo__img { }  
.photo__caption { }  
.photo__caption__quote { }  
</style>  
</body>
```

CSS Metodiky

BEM - modifikatory

```
<!-- Takhle ano -->
```

```
<figure class="photo">  
    
  <figcaption class="photo__caption">  
    <blockquote  
      class="photo__quote">  
      Look at me!  
    </blockquote>  
  </figcaption>  
</figure>
```

```
<style>  
.photo { }  
.photo__img { }  
.photo__caption { }  
.photo__quote { }  
</style>  
</body>
```

CSS Metodiky

BEM - modifikatory

```
<!-- Takhle ne -->
```

```
<figure class="photo">
```

```
  
```

```
  <figcaption class="photo__caption photo__caption--highlighted">
```

```
    Look at me!
```

```
  </figcaption>
```

```
</figure>
```

```
<style>
```

```
  .photo__img--highlighted { }
```

```
  .photo__caption--highlighted { }
```

```
</style>
```

CSS Metodiky

BEM - modifikatory

```
<!-- Takhle ano -->
```

```
<figure class="photo photo--highlighted">
```

```
  
```

```
  <figcaption class="photo__caption">
```

```
    Look at me!
```

```
  </figcaption>
```

```
</figure>
```

```
<style>
```

```
  .photo--highlighted .photo__img { }
```

```
  .photo--highlighted .photo__caption { }
```

```
</style>
```

CSS Metodiky

BEM - modifikatory

```
<!-- Takhle ne -->
```

```
<div class="somesthesis somesthesis--fastread">  
  <div class="somesthesis__someelement"></div>  
</div>
```

```
<style>  
  .somesthesis { }  
  .somesthesis--fastread { }  
  .somesthesis__someelement { }  
</style>
```


CSS Metodiky

BEM - modifikatory

```
<!-- Takhle ano -->
```

```
<div class="some-thesis some-thesis--fast-read">  
  <div class="some-thesis__some-element"></div>  
</div>
```

```
<style>  
  .some-thesis { }  
  .some-thesis--fast-read { }  
  .some-thesis__some-element { }  
</style>
```

organizace CSS

Refaktoring - BEM

- Upravte v naší refaktorované komponentě zápis, aby odpovídal metodice BEM.
- Můžete využít codepen:
<https://codepen.io/svobodalukas/pen/XxZmwY>

organizace CSS

Základní pravidla

Co nejnižší specifická

```
.heading {  
  font-size: 24px;  
}
```

```
#header .nav h3 {  
  font-size: 16px;  
}
```



```
.heading {  
  font-size: 24px;  
}
```

```
.heading--sm {  
  font-size: 16px;  
}
```

organizace CSS

Základní pravidla

CSS nezávislé na HTML

```
h1.heading {  
  font-size: 24px;  
}
```

```
nav > ul > li {  
  font-size: 16px;  
}
```



```
.heading {  
  font-size: 24px;  
}
```


```
.list-item {  
  font-size: 16px;  
}
```

organizace CSS

Základní pravidla

Snadno pochopitelné CSS

```
.heading {  
  @include my-font-b(24px, 30px, #d5d5d5);  
}
```



```
.heading {  
  font-size: 24px;  
  margin-bottom: 30px;  
  border-bottom: 1px solid #d5d5d5;  
}
```

organizace CSS

Základní pravidla

Využívání globálních a lokálních proměnných

```
.heading {  
  font-size: 24px;  
  color: #d5d5d5;
```

```
@media (min-width: 768px) {  
  ...  
}
```



```
.heading {  
  font-size: $text-xl;  
  color: $color-secondary;
```

```
@media (min-width: $screen-md) {  
  ...  
}
```

organizace CSS

Základní pravidla

Snažíme se nezanořovat (pokud to není nezbytně nutné)

```
.heading .nav .nav-footer ul > li a {  
  ...  
}
```



```
.heading-link {  
  ...  
}
```

organizace CSS

Základní pravidla

Zbytečně nevyužíváme zanořování pomocí &

```
.header {  
  background: #f5f5f5;  
  &__nav {  
    font-size: 16px;  
    &--large {  
      font-size: 18px;  
    }  
  }  
}
```



```
.header {  
  background: #f5f5f5;  
}  
  
.header__nav {  
  font-size: 16px;  
}  
  
.header__nav--large {  
  font-size: 18px;  
}
```


organizace CSS

Základní pravidla

Matematické výrazy píšeme pochopitelně

```
.content {  
  border-top: 76px;  
}
```



```
$header-height: 60px;  
$spacing-base: 16px;
```

```
.content {  
  border-top: calc($header-height + $spacing-base);  
}
```

organizace CSS

Základní pravidla

Komentujeme kód, ideálně v angličtině

```
.content {  
  position: relative;  
}
```



```
.content {  
  // used by .ad-banner (has absolute position)  
  position: relative;  
}
```

organizace CSS

Základní pravidla

Vyhňeme se !important

```
.text-red {
```

```
  color: red !important;
```

```
}
```

```
.text-red.text-more-red {
```

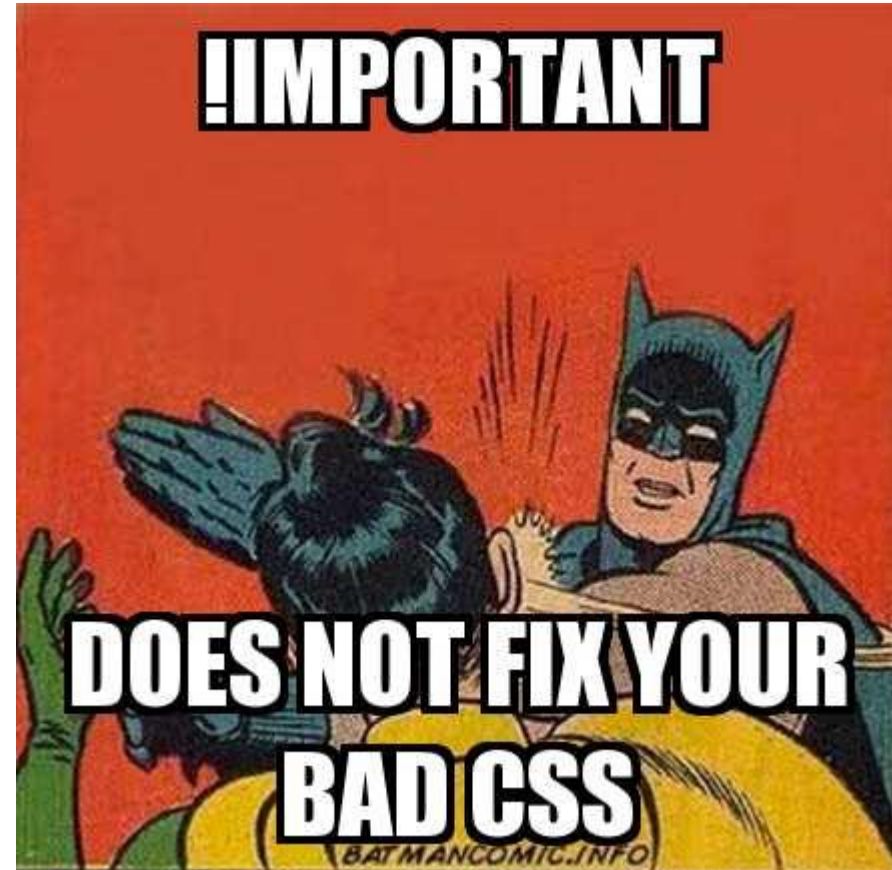
```
  color: #e30604 !important;
```

```
}
```

```
#main.text-blue {
```

```
  color: blue !important;
```

```
}
```



organizace CSS

Základní pravidla

Nepoužíváme inline styly

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

```
<li class="text-red" style="color: blue;" >  
  ...  
</li>
```

organizace CSS

Základní pravidla

nemícháme různé jednotky

```
h1 {  
  font-size: 24px;  
}
```

```
.heading {  
  font-size: 3em;  
}
```

```
.heading-2 {  
  font-size: 2rem;  
}
```



```
h1 {  
  font-size: 1.5rem;  
}
```

```
.heading {  
  font-size: 3rem;  
}
```

```
.heading-2 {  
  font-size: 2rem;  
}
```

organizace CSS

Základní pravidla

fallbacky necháme strojům

```
.block {  
  -ms-transform: rotate(90deg);  
  -webkit-transform: rotate(90deg);  
  transform: rotate(90deg);  
}
```



```
.block {  
  transform: rotate(90deg);  
}  
  
/* autoprefixer for build */
```

organizace CSS

Kdy to smrdí

podezřelé CSS konstrukce

```
h2 {  
  font-size: 2em;  
  margin-bottom: 0.5em;  
  padding-bottom: 0.5em;  
  border-bottom: 1px solid #ccc;  
}  
.no-border {  
  padding-bottom: 0;  
  border-bottom: none;  
}
```



```
h2 {  
  font-size: 2em;  
  margin-bottom: 0.5em;  
}  
.headline {  
  padding-bottom: 0.5em;  
  border-bottom: 1px solid #ccc;  
}
```

organizace CSS

Kdy to smrdí

no Single Source of Truth

```
.btn {}
```

```
.header .btn,
```

```
.header .btn:hover {}
```

```
.sidebar .btn {}
```

```
.modal .btn {}
```

```
.page aside .btn {}
```

```
nav .btn {}
```



```
.btn {}
```

```
.btn--large {}
```

```
.btn--primary {}
```

```
.btn--nav {}
```


organizace CSS

Kdy to smrdí

Brute Force

```
.foo {  
  margin-top: -12px;  
  position: relative;  
  z-index: 1000;  
  height: 48px;  
  float: left;  
}
```

organizace CSS
Demo

ukázka špatného CSS

organizace CSS
Základní pravidla

Zásady psaní respektujícího CSS

<https://www.vzhurudolu.cz/prirucka/rcss-zasady>

CSS Metodiky

Utility CSS

```
<ul class="list-reset text-center pt-6">
  <li class="pl-2 pr-1 inline-block">
    <a href="..." target="_blank" class="relative block w-10 h-10 text-black opacity-50
      border-2 border-black rounded-full">
      <svg width="20" height="20" class="fill-current absolute left-50 top-50 translate-xy">
        <use xlink:href="#ic_facebook" />
      </svg>
    </a>
  </li>
  <li class="pl-2 pr-1 inline-block">
    <a href="..." target="_blank" class="relative block w-10 h-10 text-black opacity-50
      border-2 border-black rounded-full">
      <svg width="20" height="20" class="fill-current absolute left-50 top-50 translate-xy">
        <use xlink:href="#ic_twitter" />
      </svg>
    </a>
  </li>
</ul>
```

CSS Metodiky

Utility CSS

Výhody

1. kód se píše velmi rychle
2. není potřeba zasahovat do CSS
3. malý výsledný CSS
4. styly jsou nezávislé na kontextu
5. nemusíte přemýšlet nad názvem třídy

Nevýhody

1. vypadá to jako inline styly
2. méně přehledný kód - je to ukecané
3. je potřeba si pamatovat jak se co styluje

CSS Metodiky

Utility CSS

Tailwind

<https://tailwindcss.com>

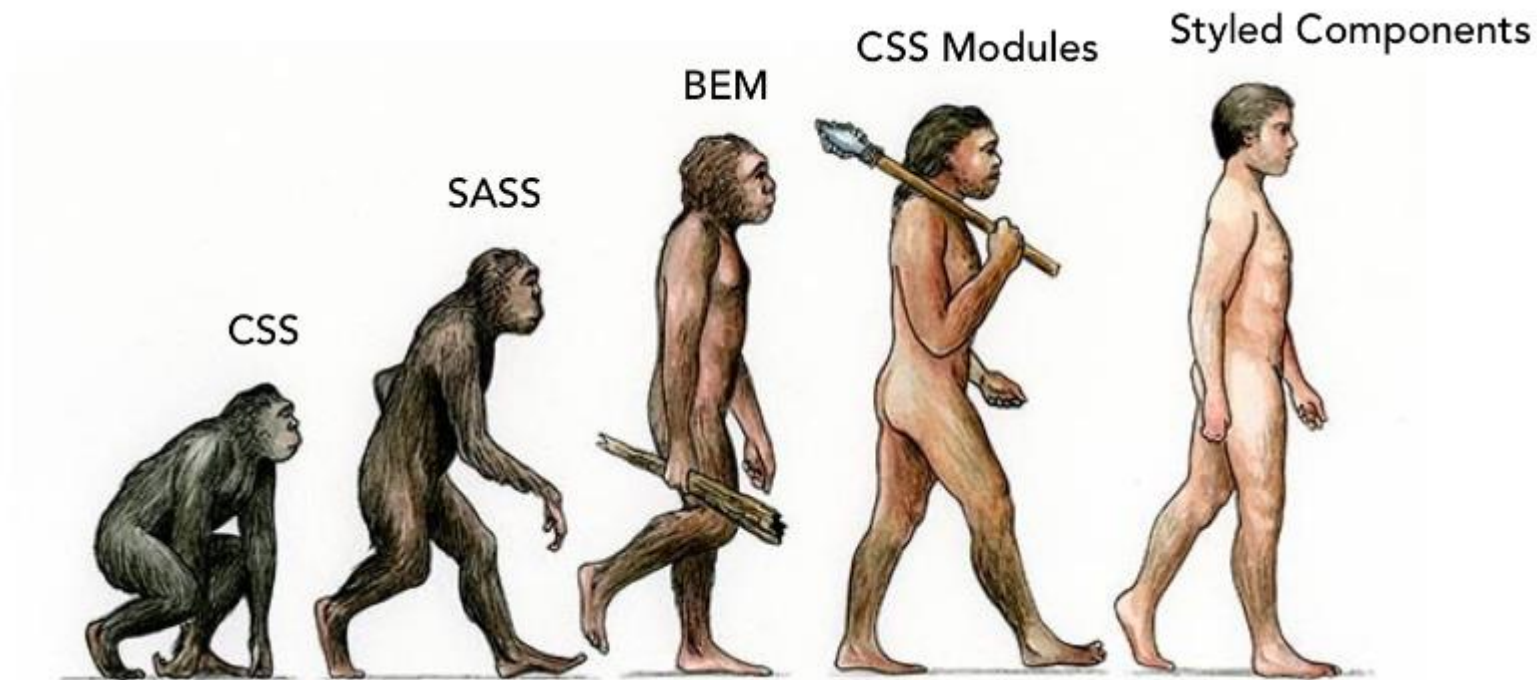
Tachyons

<https://tachyons.io/>

CSS Metodiky

Styled components & CSS Modules

<https://www.styled-components.com/docs/basics>



<https://medium.com/@perezpriego7/css-evolution-from-css-sass-bem-css-modules-to-styled-components-d4c1da3a659b>

CSS Metodiky

Styled components

```
const Button = styled.a`
display: inline-block;
border-radius: 3px;
padding: 0.5rem 0;
margin: 0.5rem 1rem;
width: 11rem;
background: transparent;
color: white;
border: 2px solid white;

${props => props.primary && css`
background: white;
color: palevioletred;
`}
`
```

```
render(
<div>
<Button
href="https://github.com/styled-
components/styled-components"
target="_blank"
rel="noopener"
primary
>
GitHub
</Button>

<Button as={Link} href="/docs" prefetch>
Documentation
</Button>
</div>
)
```


CSS Metodiky

Styled components

Výhody

1. CSS kód není globální, neexistuje dědičnost
2. vždy se vypíše jen potřebný CSS kód
3. při události mohu na jednom místě řešit vzhled i fungování
4. nemusíte přemýšlet nad názvem třídy

Nevýhody

1. závislost na JS

CSS Metodiky

Design systems

<https://adele.uxpin.com/>

Colour Palette

Find closest color


This is the primary and secondary colour palettes as described in the PSD styleguide and in [colour_palette.sass](#)

GRAY PALETTE

#2c3643	#3b444f	#67747c
\$darkgray	\$titlegray	\$bodygray


PRIMARY PALETTE

#142b44	#1d508d	#297cbb
\$darkblue	\$navblue	\$lblue
#16c98d	#feef6d	#ffc83f
\$green	\$yella	\$orange



Classic Paris: a fresh look at the city of light

Immortalised in the posters of Toulouse-Lautrec and later on

 by Phil McCracken

Card with image and author

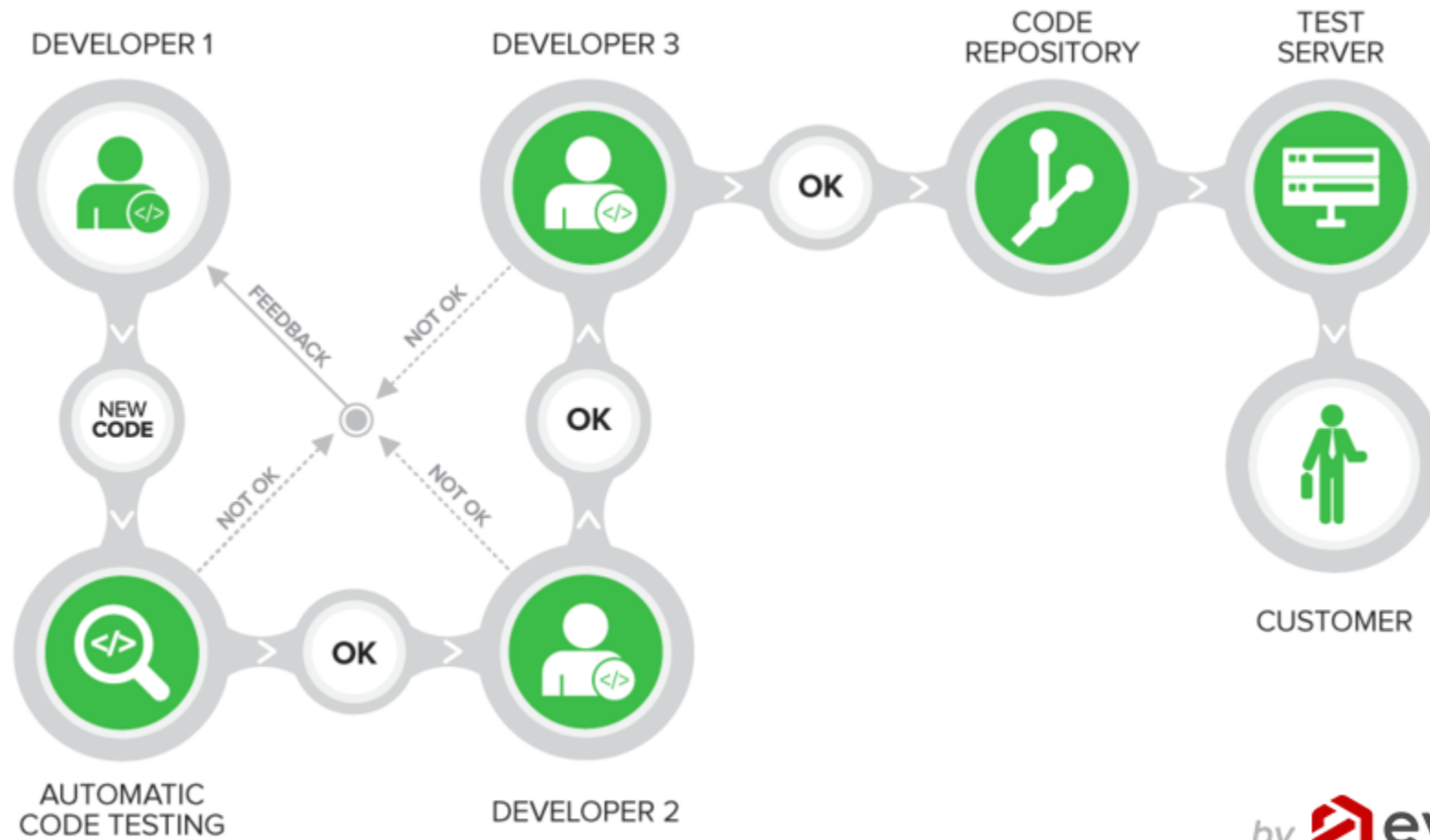
= ui_component('cards/card', properties: {as_below})

```
{
  url: "#",
  title: "Classic Paris: a fresh look at the city of light",
  description: "Immortalised in the posters of Toulouse-Lautrec and late
  image_alt: "",
  image_url: "https://www.lonelyplanet.com/travel-blog/tip-article/wordp
  author_avatar: "https://community-media.lonelyplanet.com/users/avatars
  author_name: "Phil McCracken",
  target: " blank",
}
```

Expand snippet

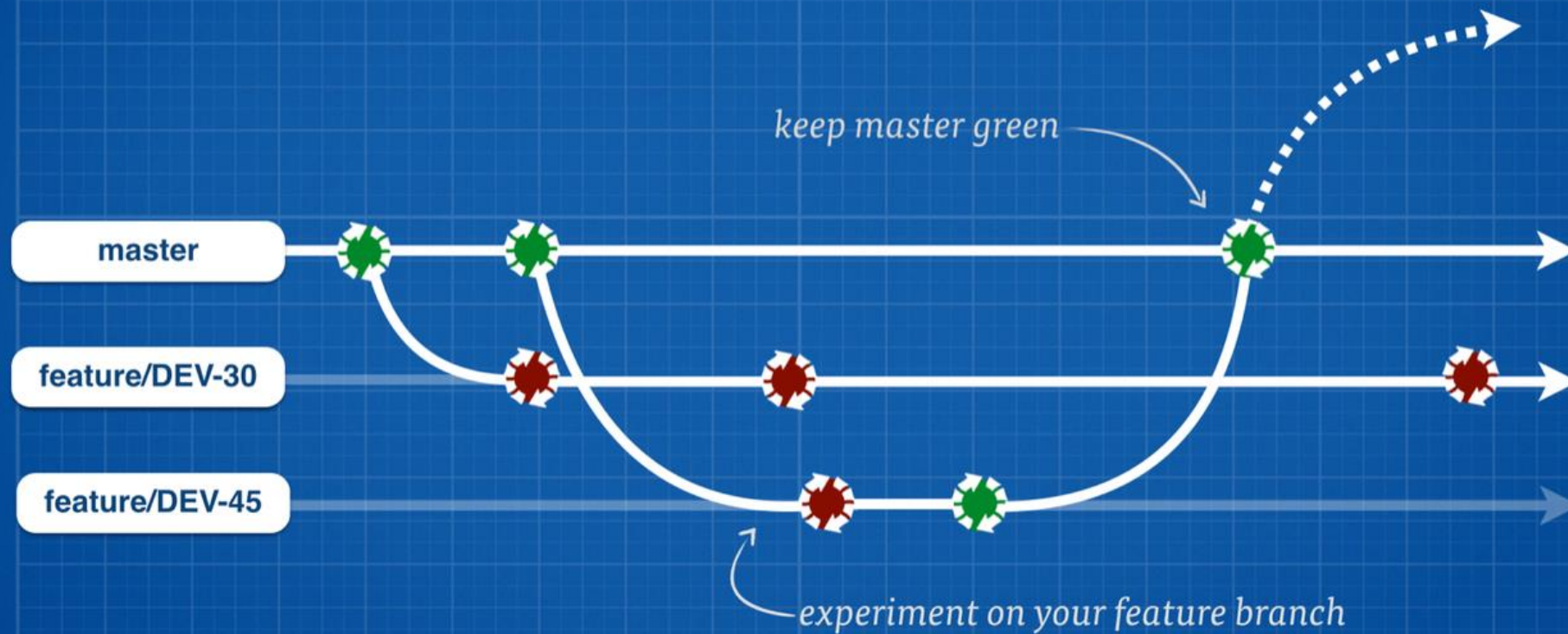
organizace CSS

Code review



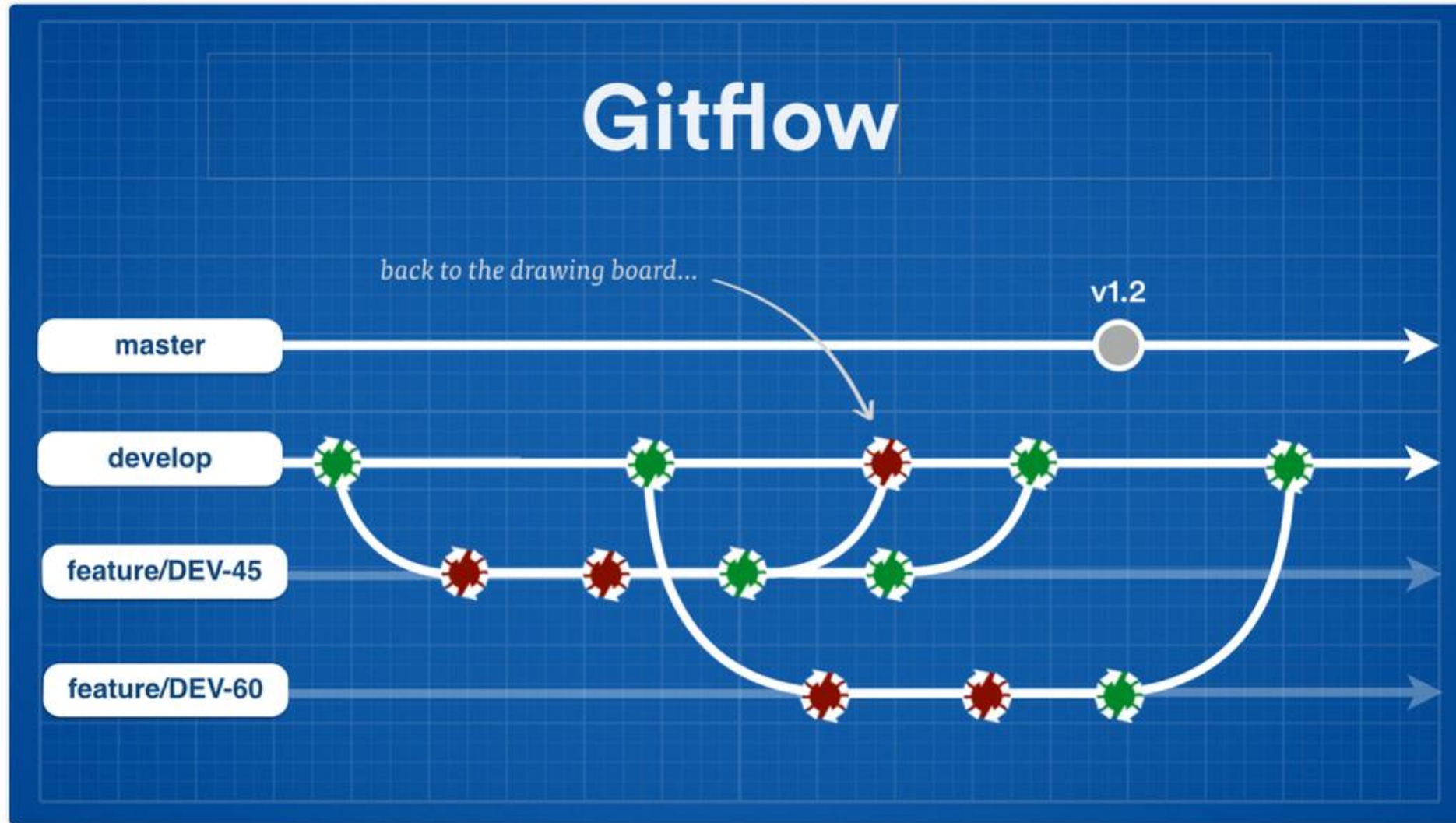
organizace CSS
Git workflow

Basic workflow



organizace CSS

Git workflow



Hledání chyb

Chyby v kódu

- Editory kódu
- W3C validátor
- Chrome Developer Tools
- Automatizační nástroje (Gulp, WebPack)

Hledání chyb

Chrome DevTools

- CTRL+SHIFT+I / CMD+OPT+I pro zobrazení
- *Elements*
 - CTRL+F / CMD+F - vyhledá selektor v kódu
 - filtrování CSS vlastností
 - možnost přidávat vlastní selektor
 - přepínání :hover stavů
 - možnost testování responsivity
- *Sources*
 - možnost modifikovat CSS a JS soubory
 - CTRL+P / CMD+P pro přepínání mezi soubory

Hledání chyb

Vizuální chyby, použitelnost

- Fyzická zařízení
- Browserstack (<https://www.browserstack.com>)
alternativy:
 - <https://crossbrowsertesting.com/>
 - <https://www.androidauthority.com/best-android-emulators-for-pc-655308/>
 - [https://docs.microsoft.com/en-us/previous-versions/windows/apps/ff402563\(v=vs.105\)](https://docs.microsoft.com/en-us/previous-versions/windows/apps/ff402563(v=vs.105))
- Automatické testování
- Guerilla testing

Hledání chyb Ladění

- Codepen – izolovaný kód, lze vyzkoušet kritický kus
- Chrome DevTools – úpravy CSS, testování hover a focus, emulace pomalého internetu
- Prohlížeče – Firefox, IE, Edge, Safari
- Reálné zařízení – mobil, tablet

