

Metodika štýlu písania kódu pre Angular



Metodika štýlu písania kódu pre Angular

Tím č. 03 FireAnt

Vedúci tímu: Branislav Pecher, Ivan Srba

Autor metodiky: [Vajk Pomichal](#)

Dátum vytvorenia: 12.10.2019

Poslednú zmenu vykonal: [Vajk Pomichal](#)

Dátum poslednej zmeny: 27.10.2019

- [Úvod](#)
- [Konvencie pomenovaní](#)
 - [Súbory](#)
 - [Kód](#)
- [Písanie kódu](#)
 - [Komentáre](#)
 - [Vnáranie blokov](#)
 - [null vs undefined](#)
 - [Úvodzovky pri reťazcoch](#)
 - [Medzery medzi členmi](#)
- [Poznámka na koniec](#)

Úvod

Budeme nasledovať konvencie určené na oficiálnej stránke [Angular.io](#). Keďže samotný dokument je veľmi rozsiahli a venuje sa nie len štýlu písania kódu, tak v nasledujúcich sekciách opíšeme najdôležitejšie pravidlá. Na písanie kódu použijeme **najmä** konvencie definované pre TypeScript opísané na stránke [odkaz](#). **Pri prípadných odchýlkach je táto metodika smerodajná.**

Odporúčané vývojové prostredie (IDE) je [Webstorm](#), ktorý poskytuje aj profesionálnu verziu v študentskej licencií.

Konvencie pomenovaní

Súbory

súbory pomenujeme podľa vzoru *feature.type.ts*

Na **oddelenie slov** v pomenovaní použijeme **pomlčky** (*hero-list.component.ts*)

Na **oddelenie názvu od typu** použijeme **bodku**, používané typy sú napr.: *.service*, *.component*, *.pipe*, *.module*, *.directive*

Kód

Na pomenovanie **tried** použijeme **PascalCase**

Na pomenovanie **funkcií a lokálnych premenných** použijeme **camelCase**

NEpoužívame prefix "I" pri názvoch interface-ov

Používame **celé slová** pri pomenovaní, ak to je možné (žiadne skratky)

Názov symbolu musí byť v súlade s názvom súboru, v ktorom sa nachádza

Symbol	Súbor
<code>@Component({ ... })</code>	app.component.ts
<code>export class AppComponent { }</code>	

Pomenovanie funkcií a parametrov musia byť **zrozumiteľné**

```
//AVOID
function div(x, y) {
  const val = x / y;
  return val;
}

//DO
function divide(divident, divisor) {
  const quotient = divident / divisor;
  return quotient;
}
```

Písanie kódu

Komentáre

Používame čo najmenej komentárov, snažíme sa o samo-opisujúci sa kód

```
//AVOID
// check if meal is healthy or not
if (meal.calories < 1000 && meal.hasVegetables) {
    ...
}

//DO
if (meal.isHealthy()) {
    ...
}
```

Vnárание blokov

Vnorené bloky **odsadzujeme pomocou 4 medzier** a otváracia zátvorka ("{" sa nachádza v riadku, ktorá začína odsadzanie.

else ide na nový riadok za uzatvárajúcou ("}") zátvorkou

```
//AVOID
if(condition)
{
  if(anotherCondition)
  {
    ...
  }
  else
  {
    ...
  }
}

//DO
if (condition) {
  if (anotherCondition) {
    ...
  }
  else {
    ...
  }
}
```

null vs undefined

používame **undefined**. Nepoužívame null.

Úvodzovky pri reťazcoch

Reťazce sa uvádzajú do **apostrofov (')**. Reťazec, ktorý obsahuje apostrof budeme uvádzať do **úvodzoviek (")**.

Medzery medzi členmi

Píšeme medzeru medzi operátora a operanda.

```
//AVOID
for(var i=0,n=str.length;i<10;i++){ }
...
if(x<10){ }
...
function f(x:number,y:string):void{ }

//DO
for (var i = 0, n = str.length; i < 10; i++) { }
...
if (x < 10) { }
...
function f(x: number, y: string): void { }
```

Poznámka na koniec

Na tejto stránke sa nachádza len výber najdôležitejších pravidiel. Väčší zoznam konvencií sa nachádza aj na stránke [odkaz](#), v prípade odchýlok je smerodajná naša metodika!!!