

# FRONT-END FEJLESZTÉS

5. forduló

cl'c k

A kategória támogatója: Click Clock by BCS -  
Business Consulting Services Group

## Ismertető a feladatlaphoz

Kezdj neki minél hamarabb, mert a feladatot a forduló záró időpontjáig lehet beküldeni, nem addig lehet elkezdni!

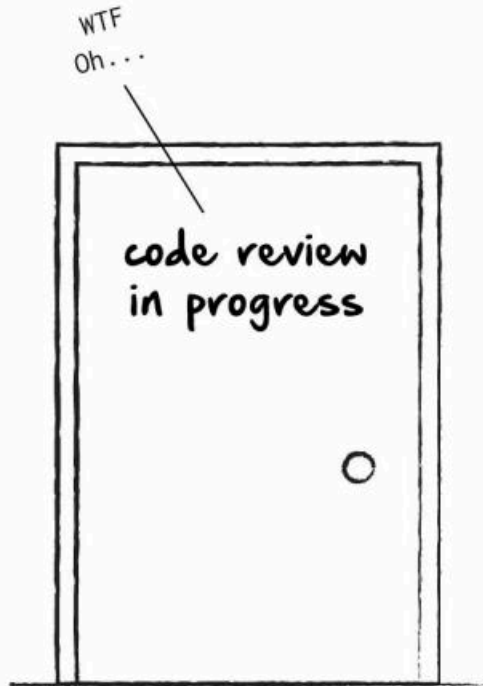
Sok sikert!

---

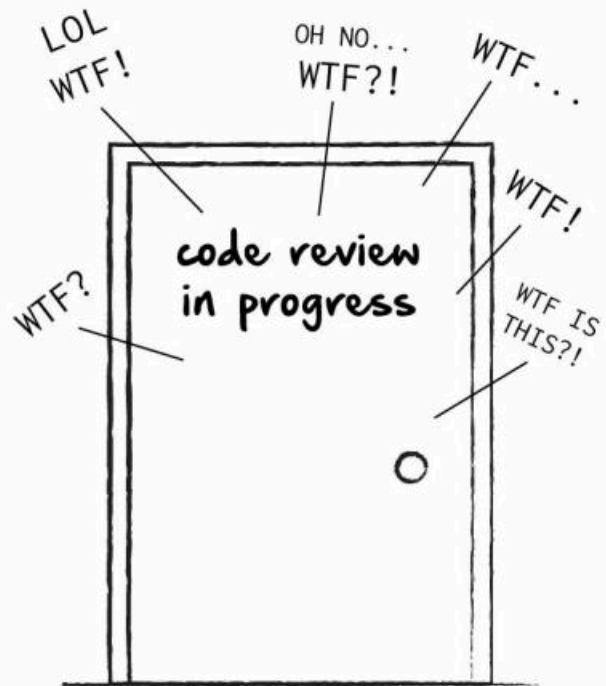
Ennek a fordulónak a témája a kódszervezés és a clean code lesz.

# Code quality is measured in WTFs/min

## Good Code











## Bad Code



## 1. feladat 1 pont

Az alábbi ábrán ugyanannak a projectnek kétféle szervezési módját látod a bal és a jobb oldalon.

- ▲  actions
  - JS cartActions.js
  - JS productActions.js
  - JS sessionActions.js
- ▲  components
  - JS cartItem.js
  - JS footer.js
  - JS header.js
  - JS loginForm.js
  - JS productBox.js
  - JS productButtons.js
  - JS registerForm.js
- ▲  containers
  - JS home.js
  - JS productDetails.js
  - JS productList.js
  - JS shoppingCart.js
- ▲  reducers
  - JS cartReducers.js
  - JS productReducers.js
  - JS sessionReducers.js

- ▲  cart
  - JS cartActions.js
  - JS cartItem.js
  - JS cartReducers.js
  - JS shoppingCart.js
- ▲  common
  - JS footer.js
  - JS header.js
  - JS home.js
- ▲  product
  - JS productActions.js
  - JS productBox.js
  - JS productButtons.js
  - JS productDetails.js
  - JS productList.js
  - JS productReducers.js
- ▲  session
  - JS loginForm.js
  - JS registerForm.js
  - JS sessionActions.js
  - JS sessionReducers.js

Mik az előnye, ha layer (bal oldal) helyett feature (jobb oldal) szerint szervezzük elsődlegesen a mappákat, file-okat egy projecten belül?

## Válaszok

- ☐ Könnyebb egy feature fejlesztése közben navigálni a kódban, hiszen egy mappában lesz minden, ami a feature-höz tartozik
- ☐ Jobb lesz a code co-location és cohesion ("the code that changes together stays together")
- ☐ Nagy project esetén kevesebb mappánk lesz
- ☐ A reducerek egy helyen lesznek.

## 2. feladat 2 pont

Mik az előnyei a CSS-in-JS megoldásoknak?

### Válaszok

- ☐ Nem fogunk letölteni annyi nem használt CSS-t a dead code elimination és a code-splitting miatt
- ☐ Jobban cache-elhető a hagyományos CSS megoldásoknál
- ☐ Gyorsabban tölt be a hagyományos CSS megoldásoknál
- ☐ Nincs selector ütközés egy nagyobb project esetében, mert scope-olt lesz így a style-unk
- ☐ Jobb a code co-location
- ☐ A CSS-in-JS megoldások mindig compile-time állítják elő a style-t

## 3. feladat 3 pont

Hogy lehet a gyakorlatban hatékonyan és folyamatosan, minél jobb, minél inkább standardizált kódminőséget elérni?

### Válaszok

- ☐ Használjunk Prettiert
- ☐ Használjunk Gruntot
- ☐ Használjunk ESLintet
- ☐ Használjunk webpacket
- ☐ Használjunk megfelelően beállított TypeScriptet (strict mode, noUnusedLocals, noUnusedParameters, noUncheckedIndexedAccess stb.)
- ☐ Hozzunk létre valahol egy mindenki által elérhető, egyedi, a cégre szabott style guide dokumentumot
- ☐ Futtassuk a kódminőséget ellenőrző toolokat az IDE-ben
- ☐ Futtassuk a kódminőséget ellenőrző toolokat a build pipeline elején