

1. Feladat - GIT verziókezelés

Leírás

A 2.-5. feladatok megoldása közben GIT verziókezelést kell alkalmaznia az alábbi követelmények szerint:

Feladatok

1. a kapott junior-frontend-vizsga mappát csomagolja ki. A kicsomagolt mappa legyen verziókezelte.
2. A mappában található **personal-info.md** fileban töltsse ki a személyes adatokat (név, email cím) és ezután commitoljon a fő ágban (initial commit)
3. Hozzon létre a feladatok megoldására 4 ágot **feladat-1, ...,feladat-4** neveken.
4. Minden feladat a saját ágában legyen megvalósítva és a feladatok megoldása közben használjon strukturáltan commitokat és commit üzeneteket.
5. A feladatok megoldása végeztével az ágakat olvassza be a fő ágba.
6. A vizsga befejezésével az elkészült állományt töltsse fel bitbucket vagy github privát repositoryba és hívja meg olvasási joggal az alábbi felhasználót a szolgáltatótól függően:
 - a. Bitbucket username: i-workshop
 - b. Github username: G-Horvath

Amennyiben a feladatok megvalósításakor keletkezik node_modules/ mappa, úgy azt sem verzió kezelni (.gitignore) sem beküldeni a megoldással nem kell!

2. Feladat - HTML+CSS

Leírás

Készítsen el az alábbi leírásnak megfelelően, HTML és CSS segítségével egy egyszerű weboldalt, ügyeljen arra, hogy a feladatokhoz tartozó tesztesetek lefussanak, valamint a tiszta kód elveinek használatára is törekedjen a kód készítése során. A formázásokat, csak és kizárólag INLINE CSS segítségével hajtsa végre, amennyiben a feladat másképp nem kéri!

A weboldal alap HTML-jét kell folytatni a megadott feladat szerint.

Feladatok

7. Az oldalon hozzon létre egy H1-es node-ot, aminek a tartalma legyen az, hogy „Főoldal”.
 - a. Állítsa be inline CSS-ben, hogy a betűköz ezen a node-on 3 pixeles legyen.
 - b. Állítsa be inline CSS-ben, hogy a szöveg legyen aláhúzott.
8. Hozzon létre egy szekció konténer elemet a H1-es node alá (az azonosítója legyen „container”)
 - a. A szélessége ennek a konténer elemnek legyen 1200 pixel
 - b. A minimum magassága legyen 200 pixel
 - c. A háttérszíne legyen EFEFEF színkódú
9. A konténer elemen belül hozzon létre két paragrafust, melynek tartalma legyen 1-1 bekezdésnyi Lorem ipsum szöveg.
 - a. Az első bekezdés kezdődjön „Lorem ipsum” szöveggel!
 - b. A második bekezdésben az egyik szó legyen egy link a Google-re (<https://www.google.com>)
10. Helyezzen el az első bekezdésben egy képet („html.png”)
 - a. A kép címe és helyettesítő szövege is legyen „HTML”
 - b. A kép legyen 300 pixel széles

3. Feladat - Bootstrap

Leírás

Készítsen el egy rendelési űrlapot Bootstrap segítségével, ügyeljen arra, hogy a feladatokhoz tartozó tesztesetek lefussanak, valamint a tiszta kód elveinek használatára is törekedjen a kód készítése során. A formázásokat, csak és kizárólag bootstrap osztályok segítségével hajtsa végre!

A weboldal alap HTML-jét kell folytatni a megadott feladat szerint.

Feladatok

1. Hozzon létre egy konténer elemet section node-dal.
 - a. Ez legyen úgy pozícionálva, hogy 6 oszlop szélességet foglaljon el az oldal közepén nagy (large) kijelzőre igazítva.
 - i. Ehhez biztosítsa a szerkezetben a szükséges egyéb elemeket is! (Működjön a grid system és középen legyen a 6 oszlop széles konténer közrefogva a megfelelő méretű side elemekkel)
2. Hozzon létre a konténerben egy form elemet aminek a átküldési metodikája legyen POST, a form végpontja pedig legyen egy „*feldolgoz.php*” állomány relatív eléréssel. *(A feldolgoz.php nem fog létezni, de ez a feladat szempontjából nem érdekes!)*
3. A form elemen belül hozzon létre 3 input szekciót, amik a következő beviteli mezőknek fogják biztosítani a helyet (ezek még nem a beviteli mezők!):
 - a. email
 - b. név
 - c. rendelés szövege
4. Az előző elemekre egységesen használja a bootstrap form csoportosító osztályát
 - a. Ezekben a szekciókban hozza létre az input mezőket a következő azonosítókkal:
 - i. email → email
 - ii. név → name
 - iii. rendelés szövege → text_message
 - b. Minden inputhoz tartozzon egy címke is, ami megadja, hogy mit kell ott kitölteni.
 - c. Biztosítsa, hogy minden input ugyanezekkel a név attribútumokkal is rendelkezzen.
5. Legyen a form alján önállóan egy submit gomb, aminek a név attribútuma legyen „Ok”.

4. Feladat - Javascript

Leírás

Készítsen el az alábbi feladatok megvalósításaként 1 javascript file-t és 3 eljárást. A tiszta kód elveinek használatára törekedjen a kód készítése során.

A feladatot a mellékelt javascript.html felhasználásával oldja meg

Feladatok

1. Hozzon létre egy js mappát. A mappában hozzon létre és kössön be egy app.js file-t a javascript.html file-ba
2. Hozzon létre egy void eljárást az alábbi követelmények szerint:
 - a) Az eljárás neve: generateNumbers
Visszatérése: -
 - b) lefutásakor a javascript.html file-ban található #result dobozba megjelenít 10 sorban soronként egy számot a számok 5 és 45 közé kell eszenek
 - c) ha nem található #result doboz akkor közölje konzol figyelmeztetéssel
 - d) A javascript.html file-ban található generate feliratú gomb futtassa le az eljárást
3. Hozzon létre egy void eljárást az alábbi követelmények szerint:
 - a) Az eljárás neve: detectNumbers
Argumentum: id - string
Visszatérése: false vagy tömb (a detektált számokkal)
 - b) lefutásakor a javascript.html file-ban található #result dobozban található számokat olvassa be egy tömbbe, amivel visszatér. Ha nem található az adott elem akkor false visszatérést ad.
4. Hozzon létre egy eljárást az alábbi követelmények szerint:
 - a) Az eljárás neve: szabadon választott név
Argumentum: tömb típusú
Visszatérése: tömb (a páros számokkal)
 - b) lefutásakor a detectNumbers('result') eredményéből gyűjti ki a páros számokat és visszatér a páros számokat tartalmazó tömbbel.
 - c) figyeljen arra, hogy a paraméterként megadott eljárás visszatérése lehet false típusú is
 - d) Az eljárás futtatásának eredményét logolja konzolra.

5. Feladat - Angular

Leírás

Készítsen el az alábbi feladatok megvalósításaként 1 javascript file-t és 3 eljárást. A tiszta kód elveinek használatára törekedjen a kód készítése során.

A feladatot a mellékelt javascript.html file-ban oldja meg

Feladatok

- 1) Készítsen egy angular projektet a neve „vizsga” legyen
- 2) A projekten belül hozzon létre egy komponenst a neve legyen „exam”
- 3) Állítsa be az elemet routingját, /exam útvonalra.
- 4) Importálja a Bootstrap-et a projektbe, elég a CSS része
- 5) Hozzon létre egy beviteli mezőt a komponens oldalán, rendelje hozzájuk a megfelelő bootstrap osztályokat
- 6) Készítsen egy gombot, oldja meg hogy event bindingolva legyen egy folyamat hozzá a neve legyen saveResult().
- 7) Készítsen legalább egy, új bekezdést az input mező alá, melyben a mező értékétől függően fog eredményt megjeleníteni, aktuálisan, ha a mező értékét változtatjuk és rákattintunk a gombra! (Két irányú adatkötéssel)
- 8) A feladata az, hogy értékelje ki az input mező értékét és írja ki a bekezdésben, az páros szám-e vagy sem. Erre készítsen el egy kiértékelő függvényt. Ha egyáltalán nem értelmezhető a beírt kifejezés, jelenítsen meg hibaüzenetet.
- 9) Oldja meg, hogy az saveResult() event végrehajtása megtörténjen a gombra való kattintáskor, a művelet pedig amit végrehajt, készítsen egy log-ot egy string típusú tömbbe, amit a dokumentum végén, megjelenít, a megfelelő angularos vezérlés segítségével. A lista is dinamikusan frissüljön a mentés után!