Պրոֆայթի ուսումնական կենտրոն, ղեկավար՝ Արմեն Մարջինյան

Profit Training Center

Խմբագիրը
Ա. Արամյան

JavaScript & jQuery

Դաս 1: Գծային ալգորիթմներ

Առաջադրանք 1: Ունենք input դաշտ և button՝

- ★ Input-ում մուտքագրվում է անուն (**օրինակ՝** Hayk) և button-ի click-ի ժամանակ h1-ի մեջ տպել “Hello Hayk”

Առաջադրանք 2: Ունենք input դաշտ և button՝

- ★ Input-ում մուտքագրվում է րոպեների քանակը, button-ի click-ի ժամանակ հաշվել թե քանի ժամ է կազմում մուտքագրված թվանշանը և h1-ի մեջ տպել այն

Առաջադրանք 3: Ունենք input դաշտ և button՝

- ★ Input-ում մուտքագրվում է թվանշան(դրամ), button-ի click-ի ժամանակ հաշվել թե քանի դոլլար է կազմում մուտքագրված թվանշանը և h1-ի մեջ տպել այն

Առաջադրանք 4: Ունենք 2 input դաշտ և button՝

- ★ Առաջին Input դաշտում մուտքագրվում թիվը, իսկ երկրորդում թվի աստիճանը, button-ի click-ի ժամանակ հաշվել մուտքագրված թվի համապատասխան աստիճանը
  - ★ Օրինակ՝ 4,2 թվերի դեպքում արդյունքը կլինի՝ 4^2=16
    - Հուշում՝ **Խնդրի լուծման համար ուսումնասիրել JavaScript Math գրադարանը**

Առաջադրանք 5: Ունենք 2 input դաշտ և button՝

- ❖ Input դաշտերում մուտքագրվում են թվեր: Առաջին դաշտում մուտքագրվում է թիվը, իսկ երկրորդում տոկոսը : Button -ի քլիկի ժամանակ ցուցադրել թվի տոկոսը
  - ★ Օրինակ՝ 150 000 և 50 - ի դեպքում կցուցադրվի 150 000-ի 50 % - ը, այսինքն՝ 150000*50/100 = 7500:

Առաջադրանք 6: Ունենք input դաշտ և button՝

★ Input դաշտում մուտքագրվում է վայրկյանների քանակը, button-ի click-ի ժամանակ մուտքագրված վայրկյանները ձևափոխել օր/ժամ/րոպե/վայրկյան տեսքի և տպել h1-ի մեջ.

❖ Օրինակ՝ 187801 վայրկյանի դեպքում արդյունքը կլինի՝ 2 օր, 4 ժամ, 10 րոպե, 1 վայրկյան

Առաջադրանք 7: Ունենք input դաշտ և button՝

★ Input դաշտում մուտքագրվում է քառանիշ թիվ, button-ի click-ի ժամանակ հաշվել մուտքագրված թվի թվանշանների գումարը

★ Օրինակ՝ 4568 թվի դեպքում արդյունքը կլինի՝ 4+5+6+8=23

Առաջադրանք 8: Ունենք input դաշտ և button՝

★ Input դաշտում մուտքագրվում է թիվ, button-ի click-ի ժամանակ հաշվել մուտքագրված թվի քառակուսի արմատը

★ Օրինակ՝ 9 թվի դեպքում արդյունքը կլինի՝ 4+5+6+8=23

Դաս 2: Ոչ գծային ալգորիթմներ

Առաջադրանք 1: Ունենք 3 ինփութ դաշտեր, որտեղ մուտքագրվում են թվեր

★ A1 կոճակի քլիկի ժամանակ ստուգել, եթե բոլոր այդ թվերը հավասար են 4 -ի, ապա h1 -ի մեջ տպել "այո", հակառակ դեպքում՝ "ոչ"

★ A2 կոճակի քլիկի ժամանակ տպել "այո" եթե մուտքագրված թվերից գոնե մեկը հավասար է 1 -ի և "ոչ" հակառակ դեպքում

★ A3 - կոճակի քլիկի ժամանակ տպել "այո", եթե այդ թվերի մեջ կան իրար հավասար թվեր, և "ոչ"՝ հակառակ դեպքում

Առաջադրանք 2: Ունենք 3 input դաշտ և button՝

- Input դաշտերում մուտքագրվում են թվեր, button-ի click-ի ժամանակ ստուգել այդ թվերից ուղղանկյուն եռանկյուն կարող է կառուցվել թե ոչ
  - ★ **Օրինակ՝** 3,4,5 - ի դեպքում ցույց կտրվի դրական պատասխան,քանի որ թվերը դրական են և 3^2 +4^2=5^2, մնացյալ դեպքերում ցույց կտրվի error:

Առաջադրանք 3: Input դաշտում մուտքագրվում է a թիվը : Button -ին սեղմելիս ցուցադրել, այդ թիվը բնական է թե իրական :

- ★ Օրինակ՝ 7 -ի դեպքում կստանանք դրական պատասխան (true), սակայն 5,2 -ը արդեն պայմանին չի բավարարում :

Առաջադրանք 4: Ունենք select և button

- ★ Select -ից ընտրում ենք երթուղիներից մեկը (օրինակ՝ Երևան - Գյումրի) : Button -ի քլիկի ժամանակ ցույց տալ, տվյալ երթուղուն համապատասխան երթուղայինները և ճանապարհածախսը:

Առաջադրանք 5: Ենթադրենք որևէ մարդ ցանկանում է գրանցվել դասընթացների եւ տեսնել, թե որքան կարժենա իր ընտրած դասընթացները

- ★ Առաջին select -ի միջից նա ընտրում է դիզայներական դասընթացի տարբերակներից որևէ մեկը
  - ○ Ենթադրենք՝ HTML /CSS, Photoshop/Illustrator կամ UI/UX
- ★ Երկրորդ սելեկտից ընտրում է ծրագրավորման դասընթացը
  - ○ Օրինակ՝ JavaScript, PHP, Java կամ C#
- ★ Երրորդ սելեկտից ընտրում է տեսական դասընթացը
  - ○ Օրինակ՝ Ալգորիթմների տեսություն, տվյալների բազաների տեսություն
- ★ Input դաշտում մուտքագրում է դասերի քանակը, որքան, որ ցանկանում է մասնակցել, ենթադրենք՝ X
- ★ Երբ սեղմում է "մասնակցել դասընթացին" կոճակի վրա անհրաժեշտ է հաշվել եւ ասել, թե որքան գումար պետք է վճարի ընտրված դասընթացին X անգամ մասնակցելու դեպքում:

- Օրինակ
  - Ենթադրենք ընտրել է HTML/CSS, Java, եւ ալգորիթմների տեսություն եռյակը
    - Դիցուք դրանցից յուրաքանչյուրի մեկ դասի արժեքը համապատասխանաբար կազմում է 4200, 5700, 12500 դրամ
    - Ենթադրենք մարդը ընտրել է դասընթացի 12 մասնակցություն:
      - Այս դեպքում նա պետք է վճարի 12*4200 + 12*5700 + 12*12500
        - Այսինքն՝ 12 * (4200 + 5700 +12500) = 268800 դրամ
    - Քլիկից հետո էկրանին ցույց տալ՝ "Դուք պետք է վճարեք X դրամ", որտեղ X -ը պետք է լինի գումարի հաշվարկման արդյունքը
      - Եթե դասընթացի մասնակցության գումարը գերազանցում է 300,000 դրամը, ապա անհրաժեշտ է ցույց տալ հետեւյալը
        - Դուք պետք է վճարեք X դրամ, սակայն ստանում եք 40% զեղչ եւ այժմ կարող եք վճարել (x-40%) դրամ

Առաջադրանք 6: Անհրաժեշտ է մշակել ավտոմատ ստուգվող թեստ կազմված 10 հարցերից:

★ Յուրաքանչյուր հարց իրենից ներկայացնում է label -ի մեջ գրված տեքստ:
  - Իսկ պատասխանները առկա են <select> -ում :

★ Օգտագործողը պատասխանները լրացնելուց հետո սեղմում է "stugel" կոճակի վրա, որպեսզի ստանա իր գնահատականը:
  - Գնահատականն անհրաժեշտ է ցույց տալ h1 -ի մեջ՝
    - Եթե այն գերազանց է ցույց տալ կանաչ գույնով,
    - Լավ արդյունքի դեպքում օգտագործել վարդագույնը,

- Բավարար արդյունքի դեպքում՝ նարնջագույնը,
- Անբավարար արդյունքի դեպքում՝ կարմիրը:

Առաջադրանք 7: 3 Input դաշտերում մուտքագրվում են a,b,c թվերը

★ Sort կոճակի քլիկի ժամանակ դասավորել մուտքագրված թվերը աճման կարգով եւ ցույց տալ h1 էլեմենտի մեջ:

Առաջադրանք 8: Կունենանք input դաշտ, որտեղ user-ը մուտքագրում է որևէ տեքստ : Երբ կսեղմի publish կոճակինն այդ տեքստը կհայտնվի h1 -ի մեջ : Ստորին հատվածում form - է, որտեղ կլինեն input դաշտեր:

★ font-size դաշտում կնշվի թե տառերը ինչ չափսի լինի
★ font-family կլինի select այստեղից կնշվեն տառատեսակը
★ color (input type= "color") դաշտում ընտրվում է տառերի գույնը ցուցում
★ background (input type= "color") դաշտում նշվում է տառերի ֆոնի գույնը
★ երբ կսեղմվի set styles կոճակին , այս բոլոր պարամետրերը կանցնեն մեր h1 ում գրված տեքստին :

Դաս 3: Լոկալ և գլոբալ փոփոխականներ(var, let)

Առաջադրանք 1: էկրանի վրա ունենք h1 և button, որի յուրաքանչյուր սեղման ժամանակ անհրաժեշտ է h1 -ի մեջ ավելացնել թվերը՝ 0 -ից սկսած: Այսպիսով առաջին սեղման ժամանակ ունենում ենք 1, ապա ՝ 2 և այսպես շարունակ: Եթե հերթական թիվը զույգ է, ապա թիվը գրել կարմիր գույնով, հակառակ դեպքում՝ կանաչ:

- Հուշում: **Զույգ թվերն այն թվերն են, որոնք անմնացորդ բաժանվում են 2 -ի այսինքն՝ x%2 == 0**

Առաջադրանք 2: էջին ունենք button(Next), button(Prev) և մեկ img :

★ Next - ի քլիկի ժամանակ img - ի պատկերը պետք է փոփոխել՝ ցուցադրելով հաջորդ նկարը, իսկ Prev -ի քլիկի դեպքում նախորդ նկարը :

★ Եթե նկարներից վերջինն է ցուցադրվում և սեղմվում է Next -ը , ապա պետք է վերադառնալ առաջին նկարին : Համապատասխանաբար առաջին նկարի դեպքում Prev սեղմելիս ցուցադրել վերջին նկարը , այսպիսով կստանանք անվերջ slider :

Առաջադրանք 3: էջի վրա ունենք նկար

- ★ Նկարի նախնական չափսերն են՝ w=200, h=200;
  - ■ Նկարի յուրաքանչյուր քլիկի ժամանակ նրա չափսերը մեծանում են 50-ով
    - Երբ նկարի լայնությունը ավելի մեծ է լինում, քան 500 -ը, ապա
      - Հաջորդ բոլոր քլիկների ժամանակ նկարը սկսում է փոքրանակ 50 -ով
        - ■ Եթե փոքրացումների ժամանակ նկարի լայնությունը փոքր է լինում 200 -ից
          - Նկարը նորից սկսում է մեծանալ եւ այդպես շարունակ

Դաս 4: Ցիկլ(for, while, do while), ցիկլի ընդհատում(break, continue)

Առաջադրանք 1: էջին ունենք 2 ինփութ դաշտ: Որոնցում մուտքագրվում են թվեր: Կոճակին սեղմելիս ցուցադրել, թե այդ թվերի արանքում քանի զույգ թիվ կա և թե որոնք են դրանք

- ★ Օրինակ. ՝ 2,9 թվերի դեպքում կստանանք հետևյալ արդյունքը ՝ "Առկա է 4 զույգ թիվ (2,4,6,8)"

Առաջադրանք 2: FIZBAZ խնդիր

- ★ Անհրաժեշտ է տպել բոլոր երկնիշ թվերը
  - Եթե թիվը բաժանվում է և 5 -ի և 3 -ի ապա նրա դիմաց գրել FIZBAZ:
  - Եթե տվյալ թիվը առանց մնացորդի բաժանվում է 3 -ի ապա նրա դիմաց գրել FIZ

- Եթե տվյալ թիվը առանց մնացորդի բաժանվում է 5 -ի ապա նրա դիմաց գրել BAZ

Առաջադրանք 3:

★ Աստղանիշների օգնությամբ նկարել հավասարակողմ եռանկյան պատկեր
★ Աստղանիշների օգնությամբ նկարել դատարկ քառակուսու պատկեր

Առաջադրանք 4:

★ Տպել առաջին երկնիշ թիվը, որը 17 -ով բազմապատկելիս ստացվում է 300 -ից մեծ թիվ
★ Տպել 8 -րդ երկնիշ թիվը, որը 17 -ով բազմապատկելիս ստացվում է 300-ից մեծ թիվ

Առաջադրանք 5: Ունենք մուտքագրման դաշտ և կոճակ : Կոճակին սեղմելիս ցուցադրել մուտքագրված թվի ֆակտորիալը

★ Օրինակ. ՝ 6 - ի դեպքում կստանանք 720 => 6! = 1*2*3*4*5 = 720
★ Թվի ֆակտորիալը 1-ից մինչև այդ թիվը բոլոր թվերի արտադրյալն է

Առաջադրանք 6: a,b,c (ոչ իրար հավասար) թվերը կոչվում են Պյութագորասյան եռյակ, եթե $c^2=a^2+b^2$

★ Օրինակ՝ 3,4,5 -ի դեպքում 25 = 16 + 9
    - Ներկառուցված ցիկլերի օգնությամբ տպել մինչև 99 բոլոր Պյութագորասյան եռյակները

Առաջադրանք 7: Ունենք 2 ինփութ դաշտ և մեկ կոճակ: Առաջինում կմուտքագրվի X թիվ երկրորդում Y: Երբ կսեղմվի կոճակին կասվի թե առնվազն քանի անգամ պետք է X -ից հանվի Y -որպեսզի X -ը ստացվի ավելի փոքր քան X/2:

★ Օրինակ. ՝ X = 10 Y = 2: Այս դեպքում կստացվի 3, քանի որ 10-ից եթե հանենք 3 հատ 3 , ապա արդյունքում կստացվի ավելի փոքր թիվ քան 5 -ը (10-ի կեսը):

Առաջադրանք 8: Ունենք ինփութ դաշտ և կոճակ : Դաշտում մուտքագրվում է թիվ : Կոճակին սեղմելիս ցուցադրել այդ թիվը երկուական համակարգում

★ Օրինակ. ՝ 45 թվի դեպքում կստանանք 101101 :

Առաջադրանք 9: Ունենք ինփութ դաշտ և կոճակ : Երբ կմուտքագրվի թիվ և կսեղմվի կոճակին ցուցադրել՝ արդյոք այդ թիվը 3 - ի աստիճան է թե ոչ :

★ Օրինակ. ՝ 9 - ի դեպքում կստանանք այո քանզի 3^3 = 9 սակայն 6 -ի դեպքում ոչ

Առաջադրանք 10:Ֆիբոնաչչիի հաջորդականություն է կոչվում թվերի խումբը, որտեղ առաջին երկու անդամներն իրար հավասար են, իսկ յուրաքանչյուր հաջորդ անդամ հավասար է նախորդ երկուսի գումարին

★ Օրինակ՝ $F_1 = 1$, $F_2 = 1$, $F_3 = F_1+F_2 = 2$, $F_4 = F_3+F_2 = 3$ եւ այլն

- Ֆիբոնաչչիի առաջին թվերն են՝ 1,1,2,3,5,8,13,21 եւ այլն
  - Նախապայմանով ցիկլի օգնությամբ տպել Ֆիբոնաչչիի հաջորդականության առաջին 20 անդամները

Դաս 5: Միաչափ զանգվածներ

Առաջադրանք 1: Հայտարարել զանգված և հաշվել

★ Միջին քառակուսային = $(x[0]^2+x[1]^2 + .. + x[n]^2)/n$, որտեղ n = x.length

Առաջադրանք 2: Երկու Input դաշտերում օգտագործողը մուտքագրում է X եւ Y թվեր, Button -ի քլիկի ժամանակ

★ Ստանալ պատահական թվերից X տարր պարունակող զանգված, որտեղ տարրերը 0 -ից Y հատվածի ամբողջ թվեր են

★ Օրինակ՝ եթե առաջին input դաշտում մուտքագրել ենք 5, իսկ երկրորդ input-ում 10 ,պետք է ստանալ 5 տարր պարունակող զանգված, որոնք կլինեն 0-10 հատվածի թվեր(օրինակ՝ [2,5,6,1,10])

- Հուշում՝ Պատահական թվերը ստանում ենք **JavaScript Math.random()** ֆունկցիայի միջոցով

Առաջադրանք 3: Տպել զանգվածում բոլոր զույգ թվերի գումարը

★ Օրինակ. ՝ [1,2,3,4,5] -ի դեպքում կստանանք 6, քանզի առկա է միայն 2 զույգ թիվ 2 և 4 , իսկ 2+4 =6

Առաջադրանք 4: Զանգվածի բոլոր բացասական թվերը դարձնել 0

- ★ Օրինակ. ՝ [1,2,-5,-6,8] -ի դեպքում կստանանք [1,2,0,0,8]

Առաջադրանք 5: Տպել զանգվածի այն տարրերը որոնք ավելի մեծ են քան զանգվածի տարրերի միջին թվաբանականը

- ★ Օրինակ. ՝ [1,2,3,4,5] -ի դեպքում միջին թվաբանականը կլինի (1+2+3+4+5)/5 = 3 ուստի պետք է տպել 4 և 5

Առաջադրանք 6: Հայտարարել X թվային զանգված

- ★ Գտնել այդ զանգվածում զույգ տարրերի արտադրյալը
- ★ Հաշվել այդ զանգվածի երկրորդ մեծագույն տարրը
  - ○ Օրինակ՝ եթե var x = [12,6,8,3] ապա առաջին մեծագույնը 12 -ն է, երկրորդը՝ 8
- ★ Ստանալ նոր Y զանգված, որի մեջ ավելացնել X զանգվածի այն տարրերը, որոնք գտնվում են [10;20] հատվածում

Առաջադրանք 7: Ունենք անուններով լի զանգված

- ★ բոլոր անունները ցուցադրել h1 -ի մեջ
- ★ ունենք input , երբ կգրվի այնտեղ անուն և կսեղմվի append կոճակին, այդ բառը կավելացվի զանգվածին և h1 -ի պարունակությունը կթարմացվի
- ★ երբ կսեղմվի remove կոճակին, ապա զանգվածից կհեռացվի հենց այդ բառը , հակառակ դեպքում, եթե մուտքագրված բառը չկա մեր զանգվածում, էկրանին ցուցադրել Error

Առաջադրանք 8: Զանգվածից հեռացնել կրկնվող տարրերը

- ★ Օրինակ. ՝ [55,44,55,30,30] -ի դեպքում կստանանք [55,44,30]

Առաջադրանք 9: Ունենք տարբեր թվերից կազմված զանգված

- ★ Ստանալ այդ զանգվածում զույգ տարրերի քանակը

★ Հաշվել այդ զանգվածի երկրորդ մեծագույն տարրը

- Օրինակ՝ եթե var x = [12,6,8,3] ապա առաջին մեծագույնը 12 -ն է, երկրորդը՝ 8

Առաջադրանք 10: Հայտարարել զանգված որը կցուցադրվի որևէ թեգում : Երբ կսեղմվի կոճակին խառնել այս տարրերը պատահական հերթականությամբ

★ Օրինակ. ՝ [1,2,3,4] -ի դեպքում կարող ենք ստանալ [2,3,4,1]

Առաջադրանք 11: Հայտարարել զանգված՝ կազմված թվերից : Փոխել զանգվածի մեծագույն և փոքրագույն տարրերի տեղերը

★ Օրինակ. ՝ [1,2,3,4] -ի դեպքում կարող ենք ստանալ [4,2,3,1]

Դաս 6: Զանգվածների կիրառումը DOM-ում (querySelectorAll)

Առաջադրանք 1: Ունենք 3 տողից և 3 սյունից կազմված դիվերի խումբ

★ Յուրաքանչյուր դիվի քլիկի ժամանակ այդ դիվի մեջ գրել 0-9 հատվածի պատահական թիվ

★ Եթե մեկ այլ դիվի վրա քլիկ անելիս այնտեղ հայտնվի կրկնվող թիվ, ապա այդ կրկնվող դաշտերին տալ որևէ պատահական գույն

Առաջադրանք 2: Ունենք div-ից ստեղծված լամպ , երբ կսեղմվի միացնել կոճակի վրա ապա վառել լամպը , կրկին սեղմելիս անհրաժեշտ է այն անջատել :

Առաջադրանք 3: էկրանի վրա ունենք 10 նկար

★ Randomize - կոճակի քլիկի ժամանակ նկարներին հաղորդել պատահական width և height

Առաջադրանք 4:

★ Քլիկ է արվում առաջին վանդակում, այնտեղ հայտնվում է x թիվ

- ★ Քլիկ է արվում երկրորդ վանդակում այնտեղ էլ է հայտնվում y թիվ
- ★ Երրորդ վանդակում քլիկ անելիս հայտնվում է z թիվը
    - Եթե x != y
        - Ապա տեղերով փոխել x -ի y -ի արժեքները, նույնը պետք է անել ցանկացած երրորդ քլիկի ժամանակ

Առաջադրանք 5: Ունենք 8 x 8 չափի դիվերի աղյուսակ (այսինքն՝ 8 տող, 8 սյուն)

- ★ Դիվերից յուրաքանչյուրի քլիկի ժամանակ, այդ դիվին տալ պատահական background գույն
    - Հաջորդ անգամ դիվի վրա սեղմելիս նախորդ գույնը պետք ա կրկին դառնա սպիտակ
        - Այսպիսով յուրաքանչյուր քլիկի ժամանակ գույն ունենում է միայն մեկ դիվ

Առաջադրանք 6: էկրանի վրա ունենք մի դիվ : Այս դիվին քլիք անելիս այն կհայտնվի էջի պատահական դիրքում և կստանա կամայական գույն :

Առաջադրանք 7: էկրանի վրա ունենք 10 նկար : Կամայական նկարի սեղմելիս այն կստանա active կլասը : Ինփութ դաշտում կգրվի որևէ նկարի հասցե և կսեղմվի apply

Բոլոր active կլաս ունեցող նկարները կստանան այդ նկարի հասցեն որպես src ատրիբուտ

Առաջադրանք 8: էկրանի վրա կունենանք 4x4 դիվերի շարք : Ստորին հատվածում կլինի 3 կոճակ :

- ★ main - կոճակին սեղմելիս անհրաժեշտ է ներկել այն դիվերը, որոնք գտնվում են գլխավոր անկյունագծում
- ★ Secondary - կոճակին սեղմելիս անհրաժեշտ է ներկել այն դիվերը, որոնք գտնվում են օժանդակ անկյունագծում
- ★ Chess - կոճակին սեղմելիս անհրաժեշտ է ներկել դիվերը շախմատային հերթականությամբ
- ★ Top - կոճակին սեղմելիս ներկել գլխավոր անկյունագծից վերև տեղակայված դիվերը
- ★ bottom - կոճակին սեղմելիս ներկել գլխավոր անկյունագծից ներքև տեղակայված դիվերը

Առաջադրանք 9: էկրանին ունենք 10 դիվ:

★ Յուրաքանչյուրին սեղմելիս այն ստանում է պատահական գույն , եթե մյուս դիվերի մեջ այս գույնը եղել է ապա այդ դիվերի մեջ կգրվի match և նրանց քլիք նրանց գույնը այլևս չէ փոխվի :

Դաս 7: Ֆունկցիոնալ ծրագրավորման ներածություն

Առաջադրանք 1:Գրել հետեյալ նկարագրությանը համապատասխանող ֆունկցիաները

★ gumar(x) - հաշվում եւ վերադարձնում է x զանգվածի տարրերի գումարը
★ Ֆունկցիան ստանում է զանգված և գտնում այդ զանգվածի մեծագույն և փոքրագույն տարրերի գումարը
★ Ֆունկցիան ստանում է զանգված տեղերով փոխում է մեծագույն և փոքրագույն տարրերը և վերադարձնում ստացված զանգվածը
★ Ֆունկցիան ստանում է տեքստ և հաշվում, թե քանի բառ կա այդ տեքստում
★ Ֆունկցիան ստանում է տեքստ և հաշվում, թե նրա բառերից քանիսն են սկսվում a տառով
★ Ֆունկցիան ստանում է զանգված, խառնում զանգվածի տարրերը (shuffle) և վերադարձնում ստացվածը
★ sortA(x) - եթե x զանգվածի տարրերը աճման կարգով են դասավորված, վերադարձնում է true, հակառակ դեպքում false
★ toUpperCaseV3(x) - x տողի յուրաքանչյուր բառի առաջին տառը դարձնում է մեծատառ
★ removeArray(x, n) - x զանգվածից հեռացնում է նրա n-րդ տարրը եւ վերադարձնում ստացվածը
    ○ Օրինակ՝ var y = [2,3,4]; y = removeArray(y, 1) -ի դեպքում կստանանք y = [2,4]
★ insertArray(x,k,n) - x զանգվածում ավելացնում է k տարրը n -րդ դիրքում
    ○ Օրինակ՝ var x= [4,1,2] զանգվածի դեպքում insertArray(x, 4,1) -ի դեպքում կստանանք հետեյալ զանգվածը՝ [4,4,1,2]
★ isPrime(x) - ֆունկցիան վերադարձնում է true, եթե x թիվը պարզ է, եւ false՝ հակառակ դեպքում
    ○ Թիվը կոչվում է պարզ, եթե այն անմնացորդ բաժանվում է միայն իր եւ մեկի վրա

        - Օրինակ՝ 5,7,11 թվերը. 8 թիվը օրինակ պարզ չէ, քանի որ բաժանվում է նաեւ 2-ի եւ 4 -ի
- ★ Օգտագործելով isPrime ֆունկցիան գրել մեկ այլ ֆունկցիա countPrime(x) -ը որը վերադարձնում է x թվերի զանգվածում առկա պարզ թվերի քանակը
- ★ Գրել match(x) - ֆունկցիան, որը վերադարձնում է x զանգվածի կրկնվող տարրերը
- ★ Գրել IsPalindrom ֆունկցիան, որը ստանում է x տեքստը եւ ստուգում է արդյո՞ք այն պալինդրոմ է
    a. Այսինքն՝ աջից եւ ձախից կարդացվում է նույն կերպ
- ★ Գրել ConvertToBinary(x) ֆունկցիան, որը 10-ական համակարգի x թիվը ձևափոխում է 2-ական համակարգի
    - Օրինակ ConverToBinary(45) կստացվի 101101

Առաջադրանք 2: JoinV2(x,y) - ֆունկցիան x զանգվածի տարրերը վերածում է տողի՝ կապելով իրար ըստ y սիմվոլի

- ★ Օրինակ JoinV2([“armen”,”hayk”, “david”], “/”])
    - Կստանանք “armen/hayk/david” տեքստը

Առաջադրանք 3: SplitV2(x,y) - ֆունկցիան x տողը տրոհում է զանգվածի՝ ըստ y սիմվոլի

- ★ Օրինակ՝ SplitV2(“gmail.@gmail.com”, “@”) կանչի դեպքում կստանանք հետեւյալ զանգվածը՝ [“gmail.”, “gmail.com”]

Առաջադրանք 4: replaceV2(s,x,y) ֆունկցիան s տողի մեջ փոխում է x ենթատեքստը y -ով եւ վերադարձնում ստացվածը

- ★ Օրինակ՝ replaceV2(“i hate javascript”, “hate”, “love”) արտահայտության դեպքում կստանանք՝ I love javascript տեքստը

Առաջադրանք 5: indexAll(x,k) - ֆունկցիան վերադարձնում է x տողի մեջ k տառի հանդիպման բոլոր ինդեքսները

- ★ Օրինակ՝ indexAll(“ararat”, “a”) -ի դեպքում կստանանք այսպիսի զանգված՝ [0,2,4], քանի որ a տառը “ararat” բառի մեջ հանդիպում է 0, 2, եւ 4 ինդեքսներում

Առաջադրանք 6: getRandomNumber(a,b) ֆունկցիան վերադարձնում է պատահական թիվ a,b հատվածից

- ★ Օրինակ՝ getRandomNumber(0,10) -ի դեպքում կարող է ստացվել օրինակ 7
- ★ Եթե ֆունկցիան ստանում է նաեւ երրորդ պարամետր

- Օրինակ getRandomNumber(0,10,5), ապա անհրաժեշտ է ստանալ զանգված, որն ունի 5 տարր եւ այդ տարրերը պատահական թվեր են 0-ից 10 հատվածից

★ Ֆունկցիան կարող է ստանալ նաեւ 4-րդ պարամետր {true,false}

- getRandomNumber(a,b,n,T)
  - Եթե t == true
    - Ապա ստացվում է n տարր պարունակող, a,b հատվածից պատահական թվեր պարունակող զանգված, որտեղ տարրերը չեն կրկնվում
      - Վերադարձնել false, եթե այդպիսի զանգված անհնար է ստանալ
  - Եթե t == false, կամ բացակայում է այդ պարամետրը, ապա տարրերը կարող են կրկնվել

Առաջադրանք 7: FilterF(x,y) - ֆունկցիան ստանում է x զանգված և y արտահայտություն և վերադարձնում է x զանգվածից այն տարրերը, որոնք պարունակում են y արտահայտություն

★ Օրինակ՝ FilterF([karine,nune,arman,sevak,mari],"ar") -ի դեպքում կստանանք այսպիսի զանգված՝ [karine,arman,mari], քանի որ "ar" արտահայտությունը հանդիպում է միայն այս բառերում

Առաջադրանք 8: makeUnique(x) - որպես արգումենտ ստանում է x զանգված և վերադարձնում է նոր զանգված որում առկա են x զանգվածի կրկնվող տարրերը

★ Օրինակ՝ makeUnique(['c', 'a', 'b', 'c','d', 'a', 'c']), այս դեպքում կվերադարձնի՝ ['a','c'] զանգվածը

Առաջադրանք 9: remake(x,y,z) այս ֆունկցիան ստանում է մեկ զանգված և երկու ինդեքս այսպիսով այն փոխում է այդ երկու ինդեքսներում գտնվող տարրերը տեղերով

★ Օրինակ՝ remake(["c", 'a', 'b'],0,2) կվերադարձնի ["b", 'a', 'c']

Առաջադրանք 10: validRange(x,y,z) ստանում է զանգված և երկու թիվ, այն վերադարձնում է true, եթե զանգվածի թվերը գտնվում են միայն այս միջակայքում: Եթե զանգվածում առկա է այնպիսի թիվ, որը նշված միջակայքում չի, ֆունկցիան վերադարձնում է false

★ Օրինակ՝ validRange([1,2,3,4,5],3,5) այս դեպքում կվերադարձնի false, քանի որ առկա են 3-ից փոքր թվեր (3 -ից 5 նշանակում է , որ 3ից փոքր չլինի և 5 ից մեծ)

Առաջադրանք 11:maskify(x) - ստանում է որպես պարամետր որևէ x տող, և եթե տողի երկարությունը մեծ է 4 ից, ապա վերադարձնում այն, բոլոր սիմվոլները փոխակերպված # -ով, բացի վերջին 4 սիմվոլները

- ★ օր. ՝ maskify(“123456789”) պետք է վերադարձնի #####6789
  - սակայն եթե նշանների քանակը 4ից փոքր է ապա պետք է վերադարձնի նույն կերպ

Դաս 8: Ռեկուրսիա, ռեկուրսիվ ֆունկցիաներ

Առաջադրանք 1: Qanak(x,k) - հաշվում է x զանգվածում k տարրերի քանակը

Առաջադրանք 2: Գրել ֆունկցիա, որը կստուգի փոխանցվող զանգվածի տարրերը աճման կարգով են դասավորված թե ոչ:Եթե աճման կարգով դասավորված չեն վերադարձնում է false

Առաջադրանք 3: Գրել ֆունկցիա, որը որպես պարամետր ստանում է տող,և եթե տողում կան մեծատառով սկսվող բառեր, ապա առանձնացնում է բացատով

- ★ օրինակ՝ myFunction(“barevErevanJan”), պետք է վերադարձնի barev Erevan Jan տեքստը

Առաջադրանք 4: indexF(x,k) - վերադարձնում է x զանգվածում k թվի ինդեքսը, իսկ եթե տվյալ թիվը չկա զանգվածում վերադարձնում է -1

- ★ Օրինակ indexF([4,3,2,8,9], 3) -ի դեպքում կստանանք 1, քանի որ x[1] = 3

Առաջադրանք 5: MaxF(x) - ֆունկցիա վերադարձնում է զանգվածի մաքսիմումը

Առաջադրանք 6: evenSum(x) ֆունկցիան հաշվում է x զանգվածի զույգ տարրերի գումարը

Առաջադրանք 7: PowF(x,y) - ֆունկցիան հաշվում է x թվի y աստիճանը

- ★ Օրինակ՝ PowF(2,3) = 2^3 = 8, եթե y-ը նշված չի,ֆունկցիան վերադարձնում է x թվի քառակուսին

Առաջադրանք 8: fibonacci(x) - ֆունկցիան վերադարձնում է զանգված որի տարրերը Ֆիբոնաչիի հաջորդականության առաջին x թվերն են :

★ Օրինակ՝ fibonacci(5) կվերադարձնի [1, 1, 2, 3, 5];

Առաջադրանք 9: generalize(arr) - ֆունկցիան ստանում է որպես արգումենտ որևէ զանգված՝ բաղկացած բացառապես թվերից և վերադարձնում է այս զանգվածի թվերի ընդհանուր ամենամեծ բաժանարարը:

★ Օրինակ՝ generalize([20, 155, 30]) կվերադարձնի 5, քանի որ ամենամեծ բնական թիվը, որի վրա 20-ը 155-ը և 30 անմնացորդ բաժանվում են հենց 5-ն է:

Առաջադրանք 10: Գրել ֆունկցիա, որը որպես պարամետր ստանում է զանգված, և վերադարձնում է զանգվածի զույգ տարրերի միջին թվաբանականը

★ Օրինակ՝ avgF([2,4,5,3,7,8,4]), պետք է վերադարձնի 5, քանի որ (2+4+8+6)/4=5

Դաս 9: Անանուն ֆունկցիաներ, լյամբդա արտահայտություն(sort, map, filter, reduce)

Առաջադրանք 1: Ամբողջ թվեր պարունակող x զանգվածը դասավորել այնպես, որ սկզբից լինեն զույգ թվերը, վերջում՝ կենտերը

Առաջադրանք 2: Ամբողջ թվեր պարունակող x զանգվածը դասավորել այնպես, որ սկզբում լինեն պարզ թվերը, վերջում՝ մնացյալը

Առաջադրանք 3: Անուններ պարունակող զանգվածը դասավորել այնպես, որ Պալինդրոմ անունները լինեն սկզբում

Առաջադրանք 4: Ունենք անունների զանգված,զանգվածի յուրաքանչյուր տարրի առաջին տառը դարձնել մեծատառ

Առաջադրանք 5: Ունենք անունների զանգված,զանգվածից հեռացնել այն տարրերը, որոնք պարունակում են a տառ

Առաջադրանք 6: Ունենք անունների զանգված,զանգվածի տարրերը դասավորել այբբենական կարգով

Առաջադրանք 7: filterV2(x, F) - ֆունկցիան x զանգվածում թողնում է միայն այն տարրերը, որոնք բավարարում են F ֆունկցիային

★ Օրինակ՝ x = filterV2([2,4,6,5,6,13,2], function(y){
return y % 2 != 0;
})
Դեպքում x զանգվածը կդառնա [5,13]
  - Նկատենք, որ եթե ֆունկցիայի պարամետրը մեկ այլ ֆունկցիա է, ապա այնտեղ կարող է կիրառվել նաեւ լյամդա արտահայտություն
    - Օրինակ՝ x = filterV2([4,2,3,2,3], y => y==2)

Առաջադրանք 8: mapV2(x,F) - ֆունկցիան x զանգվածի տարրերը անցկացնում է F ֆունկցիայի միջով

★ Օրինակ x = mapV2([4,2,3,2,3], y => y+1) կանչի դեպքում կստանանք x = [5,3,4,3,4] զանգվածը

Առաջադրանք 9: Ունենք անունների զանգված. գրել ֆունկցիա, որը կվերադարձնի զանգվածի այն անունները որոնց երկարությունը մեծ է 7- ից:

Առաջադրանք 10: Գրել ֆունկցիա, որը տրված զանգվածից կհեռացնի բացասական թվերը:

Առաջադրանք 11: Գրել ֆունկցիա, որը տեքստի յուրաքանչյուր տառից առաջ ավելացնում է բացատ:

Առաջադրանք 10: Գրել ֆունկցիա, որը կվերադարձնի զանգվածի այն տարրերը, որոնք 10-ի բազմապատիկ են:

Առաջադրանք 12: intersection() ֆունկցիան , որը ստանում է որպես պարամետր 2 կամ 3 զանգված և վերադարձնում է մեկ այլ զանգված : Այն բաղկացած կլինի փոխանցված զանգվածների ընդհանուր տարրերից :

★ Օրինակ՝ intersection([1, 2, 3], [101, 2, 1, 10], [2, 1]) կվերադարձնի [1, 2]

Դաս 10: jQuery ներածություն

Առաջադրանք 1:Ստեղծել Calculator, որը հնարավորություն է տալիս պարզագույն գործողություններ կատարել թվերի հետ

- ★ Պարզագույն գործողություններ՝ +, -, *, /, sin, cos, ֆակտորիալ, armat, x^2, x^3
- ★ Գործողություններ իրական թվերի հետ 4.5 + 2.7
    - Թույլ չտալ, որպեսզի թվերում կետը կրկնվի
        - Օրինակ՝ 2...6 կամ 5.55.3 + 4
        - Թույլ տալ գրել 0.5 (0 -ով սկսվող)
- ★ C -ի քլիկի ժամանակ դարձնել 0 արժեքը
- ★ <- քլիկի ժամանակ ջնջել վերջին նիշը
    - Օրինակ՝ 576 -ի դեպքում կստացվի 57, հաջորդ քլիկում՝ 5, հաջորդում՝ 0
- ★ Թույլ տալ կատարել մի քանի գործողություն՝
    - օրինակ՝ 2+3-4*2
- ★ Հնարավորություն տալ գործողություններ անել նաեւ բացասական թվերի հետ.
    - Օրինակ՝ -5+5 , որը կստացվի 0
    - սա նշանակում է,որ կունենանք մի կոճակ, նրան սեղմելիս դիմացից կավելացվի մինուս նշան,եթե այն չկար, հակառակ դեպքում կհեռացվի:
- ★ Bin կոճակի քլիկի ժամանակ տրված թիվը ներկայացնել երկուական համակարգում
    - Օրինակ՝ 45 -ի դեպքում 101101
        - http://www.wikihow.com/Count-in-Binary
        - http://www.electronics-tutorials.ws/binary/bin_2.html

Առաջադրանք 2:Ունենք ռեգիստրացիոն ֆորմա, որտեղ օգտագործողը լրացնում է իր տվյալները

- ★ Անուն, ազգանուն, տարիք եւ այլն (օրինակ՝ bootstrap ֆորմա): Button -ի վրա քլիկի ժամանակ ստուգել՝ արդյոք բոլոր դաշտերը լրացված են, թե ոչ
    - ★ Եթե ոչ, ապա ցույց տալ error օգտագործելով bootstrap -ի alert alert-danger կլասը
    - ★ Հակառակ դեպքում՝ ցույց տալ հաջողության մասին հաղորդագրություն օգտագործելով bootstrap -ի alert alert-success ֆունկցիան
    - ★ Հաղորդագրությունները ցույց տալ անիմացիայով (show, fadeIn եւ այլն)

Առաջադրանք 3: Ունենք h1 թեգեր, որոնք ունեն պատահական background : Էջում կլինի նաև input դաշտ:

- ★ input -ի մեջ կմուտքագրվի որևէ տեքստ :
- ★ Կամայական h1-ի վրա կրկնակի քլիք անելիս՝ պետք է նրա մեջ գրվի input -ի արժեքը :
- ★ Եթե h1-ի վրա կկատարվի աջ քլիք, ապա վերջինիս մեջ գրված տեքստը կդատարկվի :

Առաջադրանք 4: էջում կլինի Input դաշտ search -ի համար, ol և filtrate կոճակ :
Նախապես կունենանք նաև զանգված, որում առկա են մարդկանց անուններ :

- ★ Input-ում կգրվի որևէ տեքստ և կսեղմվի կոճակին :
- ★ Ol - ի մեջ կգրվեն այն մարդկանց անունները որոնց անվան մեջ գտնվել է input -ում մուտքագրված տեքստը :
- ★ Օր.՝ [‘Aram’, ‘Suren’, ‘Karen’] մեր զանգվածն է : Input -ում գրվում է ‘re՛ և սեղմվում է կոճակին. այս դեպքում կցուցադրվեն Suren և Karen անունները, քանի որ նրանց անվան մեջ եղել է ‘re’ տառակապակցությունը:
- ★ Դիզայնի օրինակ http://prntscr.com/h0fzqy:

Առաջադրանք 5: էջում նախապես կունենանք հետևյալ div-ը՝ http://prntscr.com/h0fyx4

- ★ Add Data քլիկի ժամանակ կբացվի ֆորմա, որտեղ կլրացվեն օգտատիրոջ տվյալները՝ համապատասխանաբար անունը, ազգանունը, տարիքը, նկարի հասցեն, սոցիալական ցանցի հասցեն, ինչպես նաև հավելյալ տեղեկություններ:
- ★ Save կոճակին սեղմելիս՝ անհրաժեշտ է որպեսզի տվյալները փոխանցվեն սկզբնական div-ի համապատասխան դաշտերում:

Դաս 11: jQuery - անիմացիա և callback ֆունկցիաներ

Առաջադրանք 1:Նախկինում, ուսանողի կողմից մշակված որևէ կայքում ավելացնել հետևյալը

- ★ Header -ում menu -ին սեղմելիս՝ պատուհանը կիջնի մինչև էջի համապատասխան հատվածը (animate):
- ★ Երբ էջի scroll -ը իջնում է 100 -ից, ապա header -ը կփոքրանա, հակառակ դեպքում նորից կմեծանա:

★ Ստորին աջ անկյունում կլինի ֆիքսված կոճակ, որի թափանցելիությունը կփոխվի scroll -ի հետ համատեղ . Որքան scroll -ը վերև է գտնվում այնքան կոճակը ավելի մուգ է . Կոճակին սեղմելիս անհրաժեշտ է էջը բարձրացնել դեպի վերին դիրք (2 վայրկյանում):

Առաջադրանք 2:Ուղարկված կայքում ավելացնել հետևյալ փոփոխությունները

★ Պատրաստել հետևյալ հատվածը՝
  - http://prntscr.com/gud2x8
★ Menu-ի li-երին սեղմելիս animate-ով պետք է տեղափոխվենք համապատասխան բաժին
★ էջի ներքևում տեղադրված ֆորման անհրաժեշտ է ենթարկել վալիդացիայի
  - Error-ները ցույց պետք է տրվեն fadeIn/fadeOut էֆեկտներով, ճիշտ այնպես, ինչպես օրիգինալ կայքում է՝http://prntscr.com/dbo5n5
    - Կայքը՝https://drive.google.com/file/d/0BzOEcXEXwKmGZ191ZVpGVThmbVdzVi0ycXpyMGtGeWZyeF9J/view

Դաս 12/13/14: jQuery դինամիկ ծրագրավորման մեթոդը

Առաջադրանք 1: էջում կլինի Input դաշտ և span :

★ Span -ում ցուցադրվում է թե քանի սիմվոլ է գրված input -ի մեջ և քանի սիմվոլ կարող է ամենաշատը մուտքագրվել օր. ՝ 5/12 սա նշանակում է, որ ներկա պահին գրված է 5 տառ, իսկ ընդհանուր կարելի է գրել 12 տառ, այսինքն մնացել է մուտքագրելու ևս 7 տառ :
★ Եթե input ում մուտքագրված տառերի քանակը հասնում է թույլատրելի սահմանին, ապա input-ը կարմիր եզրագիծ է ստանում և չի թույլատրում այլևս տեքստ գրել:

Առաջադրանք 2: էջին ունենք 1 input և 1 select դաշտ :

★ Input -ում կմուտքագրվի x թիվ (օր. ՝ 3): select դաշտից, պետք է ընտրվի որևէ կենդանու տեսակ (օր. ՝ փիղ, արջ ...)

- Երբ կսեղմվի create կոճակին համապատասխան քանակությամբ (x) կստեղծվեն նկարներ, որոնցում պատկերված կլինի select -ից ընտրված կենդանին:
- Նկարներին սեղմելիս անհրաժեշտ է տալ նրանց active կլասը (արտահայտիչ եզրագիծ), ևս մեկ անգամ սեղմելիս այդ կլասը կհեռացվի
- Remove կոճակին սեղմելիս՝ բոլոր active կլասը ունեցող նկարները կհեռացվեն էջից:

Առաջադրանք 3: Տրված է table: Երբ քլիք կարվի td-ներից որևիցե մեկի վրա, այն կստանա կարմիր background, երկրորդի վրա քլիք անելու դեպքում տեղերով փոխել նրանց մեջ գրված տեքստերը: Եթե արդեն նշված որևէ td ունենք, ապա կամայական վայրում աջ քլիքը կհանի նշումը:

Առաջադրանք 4: Ունենք երկու ինփութ դաշտ

★ Առաջինում մուտքագրվում է x թիվը երկրորդում y
- Button -ի քլիկի ժամանակ դինամիկ կերպով ստեղծել HTML աղյուսակ, որն ունի x տող և y սյուն
- Աղյուսակի td -ներից յուրաքանչյուրի քլիկի ժամանակ այդ դաշտին տալ պատահական գույն, իսկ մյուս td -ների գույնը նորից դարձնել սպիտակ

Առաջադրանք 5: Ունենք անունների զանգված

★ Ընտրել այդ զանգվածից պատահական անուն
★ Shuffle անել անունը
★ Համապատասխան տառերով ստեղծել դիվեր և տեղադրել դրանք էջի պատահական դիրքերում
★ Օգտագործողը պետք է քլիկ անելու տարբերակով վերականգնի անունը
- Եթե հերթական քլիկը ճիշտ տառի վրա է, այն դարձնել կանաչ, հակառակ դեպքում՝ կարմիր

Առաջադրանք 6: Դինամիկ կերպով էջին ավելացնել նկարներ

★ Քլիկ անելով յուրաքանչյուր նկարի վրա վերջինս ընտրվում է, այսինքն՝ ստանում է active կլասը
- Եթե արդեն ուներ ակտիվ կլասը, ապա հեռացնել այդ կլասը

★ Ունենք button, որը կոչվում է change style

- Քլիկ անելով այդ դաշտի վրա հայտնվում է դիվ, որի մեջ կան տարբեր Input դաշտեր և save կոճակը
  - Input type = range - սրանով կսահմանվի ընտրված նկարների border-radius -ը
  - Input type = text - սրանով կսահմանվի լայնությունը
  - Input type = text - սրանով բարձրությունը
  - Input type = range - սրանով թափանցելիությունը (opacity)
- save կոճակին քլիկ անելիս անհրաժեշտ է բոլոր ընտրված նկարները (active կլաս ունեցող) ստանան բոլոր այդ սթայլեր, որոնք տրվել են դիվի մեջ
  - save կոճակի քլիկից հետո դիվը պետք է անհետանա

Առաջադրանք 7: Create կոճակի քլիկի ժամանակ դինամիկ մեթոդով ստեղծել 10 դիվեր և տեղադրել էջի տարբեր կետերում

★ Փոխել դիվերի կոորդինատները յուրաքանչյուր 500 միլիվայրկյանի ընթացքում և նրանց տալ պատահական արժեքներ

- Transition տալու դեպքում կստացվի, որ դիվերը շարժվում են պատահական ուղղություններով
- Յուրաքանչյուր դիվի քլիկի ժամանակ դադարեցնել քլիկ արվածի շարժը

Առաջադրանք 8: Տրված է դիվ: Յուրաքանչյուր անգամ էջը թարմացնելիս անհրաժեշտ է այդ դիվին ավելացնել պատահական քանակությամբ checkbox-ներ, որոնք նշված կլինեն պատահական սկզբունքով:

Դաս 15/16: Ինտերակտիվ ծրագրավորում Event-ների օգնությամբ

Առաջադրանք 1: Տրված են input դաշտեր (ենթ. 15 հատ): Առաջին վայրկյանին դինամիկ կերպով առաջին input -ի արժեքը դարձնել 1: Մեկ վայրկյան հետո երկրորդ input -ի արժեքը կդառնա 2 և այդպես մինչև վերջին input-ը, երբ վերջին ինպուտում կգրվի վերջին թիվը, այս գործընթացը չի դադարը, այլ կվերսկսվի նորից սակայն այս անգամ առաջին input-ում հաշվարկը կսկսվի 15 ից:

Առաջադրանք 2:Մշակել հետևյալ խաղը՝

- ★ Start կոճակին սեղմելիս վերևից թափվում են բառեր 500 միլիվայրկյան ինտերվալով(բառերը կարող եք պահել զանգվածում, և ընտրել պատահական բառ), օգտվողը պետք է հասցնի հավաքել բառը մինչև բառի էկրանի ներքևի պատը հասնելը(օգտվում եք keydown event -ից)
- ★ յուրաքանչյուր ճիշտ հավաքված բառը խաղացողին բերում է մեկ միավոր
- ★ եթե բառը հասնում է էկրանի ներքևի պատին ,ապա խաղն ավարտվում է:

Հուշում՝ ինտերվալի յուրաքանչյուր քայլում ընտրել զանգվածից պատահական բառ և ստեղծելով դիվ,ավելացնել body-ին պատահական դիրքում: Երբ դիվը անցնում է ($(div).position().top>$('body').height()), ապա դադաարեցնել խաղը և ցուցադրել պարտության մասին հաղորդագրություն:

Առաջադրանք 3:Ունենք 3 դիվեր, որոնցից մեկը ներկայացնում է կարմիր, մյուսը՝ կանաչ, մյուսը՝ կապույտ դիվեր: Կա նաև 3 այլ տարրաներ, որոնք դատարկ են: Մկնիկի օգնությամբ հնարավոր է գույները տեղափոխել համապատասխան դատարկ տարրաների մեջ(կապույտը՝ կապույտ եզրագիծ ունեցող դիվի, կարմիրը՝ կարմիր, իսկ կանաչը՝ կանաչ եզրագծով դիվի): Որի արդյունքում տարրան կներկվի համապատասխան գույնով:

Առաջադրանք 4:Ունենք HTML table, որի մեջ լրացված են տվյալներ

- ★ Օրինակ՝ անուն, ազգանուն, տարիք և այլն
  - Յուրաքանչյուր td -ի վրա double քլիկի ժամանակ
    - Այդ td -ի մեջ դինամիկ ավելացնել ինփութ, որի մեջ գրված կլինի այն տեքստը, ինչ գրված էր td -ում
    - Տեքստը փոխելուց և որևէ այլ տեղ քլիկ անելուց (օգտագործել change event-ը) փոխել աղյուսակի տեքստը

Առաջադրանք 5: էջի պատահական կետին քլիք անելիս՝ անհրաժեշտ է տվյալ հատվածում դնել օղակ (կլոր դիվ) , երբ քլիք կարվի նույն օղակի վրա ,ապա այն կանհետանա:

Առաջադրանք 6:

Առաջադրանք 7:

Առաջադրանք 8:

Առաջադրանք 9:

Առաջադրանք 10:

Դաս 17:OOP

Առաջադրանք 1:Ունենք Erankyun և Uxxankyun կլասները

★ Եռանկյունը բնութագրվում է a,b,c կողմերով (ուղղանկյան դեպքում այն կունենա միայն a և b) եւ երեք մեթոդներով

- Paragic - հաշվում է տվյալ պատկերի պարագիծը
- Makeres -հաշվում է տվյալ պատկերի մակերեսը
- Յուրաքանչյուրի մեջ ունենալ draw() մեթոդը
  - Uxxankyun - կլասում, Draw() մեթոդը նկարում է ուղղանկյան պատկեր աստղանիշներով,
    - Եթե մեթոդը ստանում է որևէ պարամետր
      - Օրինակ՝ Draw(true) ապա ուղղանկյունը պետք է լինի դատարկ
  - Erankyun - կլասում
    - Draw մեթոդը նկարում է ուղղանկյուն եռանկյուն
      - Draw(true) կանչի դեպքում անհրաժեշտ է նկարել հավասարասրուն եռանկյուն

Առաջադրանք 2:Ստեղծել Timer կլասը

a. var t = new Timer(x); t.start()
   i. Դինամիկ կերպով էջին կավելացնի հետհաշվարկ կատարող timer

b. var t2 = new Timer(5); t.start(t2) կաշխատի t -ն, անմիջապես ավարտին կաշխատի t2 -ը

Առաջադրանք 2: Ունենք Project կլասը, որը բնութագրում է պրոյեկտին:

Կլասի դաշտերն են.

- name - պրոյեկտի անվանումը
- languages - օգտագործված ծրագրավորման լեզուների զանգվածը:

Օրինակ՝ let p1 = new Project("Marketing Site", ["JS", "Angular","CSS"]);

Ունենք նաև Developer կլասը, որը ինֆորմացիա է պարունակում ծրագրավորողի մասին:
Կլասի դաշտրեն են.

- name - Անուն
- surname - ազգանուն
- projects - զանգված, որի մեջ կպահպանվեն նրա կատարած պրոյեկտները (օբյեկտնե Project կլասից)
- photo - նկարը

Developer կլասի մեթոդներն են

1. addProject(x) - projects զանգվածին ավելացնում է x project -ի տվյալը
2. toString - էկրանին տպում է տվյալ ծրագրավորողի տվյալը և նրա կատարած պրոյեկտների տվյալները

   Օրինակ՝

   ```
   let p = new Developer("Hayk", "Davtyan", "1.jpg");

   p.addProject(new Project("Basic Store", ["AngularJS", "PHP"]));

   p.addProject(new Project("Social Network", ["Node.js", "React.js"]));
   ```

p.addProject(new Project("3D Snake Game", ["JS", "Canvas","CSS3"]));

p.toString() ֆունկցիան կանչելու դեպքում էկրանին կհայտնվի ծրագրավորողի տվյալը և նրա կատարած պրոյեկտների տվյալները գեղեցիկ դիզայնով

Առաջադրանք 3: Մշակել Calendar համակարգը

- ★ Այն իրենից ներկայացնում է մեկ ընդհանուր կլաս, որը ունի ամիս , տարի դաշտերը և get_days, create մեթոդները :
- ★ Get_days - ֆունկցիան կանչելիս, այն վերադարձնում է տվյալ ամսվա բոլոր օրերը զանգվածի տեսքով : Յուրաքանչյուր օր իրենից ներկայացնում է օբյեկտ, որը ունի ամսաթիվ [1-31] և օր [0-6 (ուրբաթ, շաբաթ. . .)] դաշտերը:
- ★ Create մեթոդը կանչելիս՝ պետք է ցուցադրվի աղյուսակ, որը պատկերում է տվյալ ամսվա բոլոր օրերը օրացույցի տեսքով :
- ★ էջում կունենանք նաև 2 հավելյալ կոճակներ (next, prev) :
  - Next -ին սեղմելիս կպատկերվի հաջորդ ամսվա օրացույցը
  - Prev -ին սեղմելիս կպատկերվի նախորդ ամսվա օրացույցը
- ★ Օրացույցը պատկերելիս, եթե տվյալ ամսվա մեջ առկա է ներկա պահի ամսաթիվը, ապա վերջինս ներկել կանաչ գույնով :

Առաջադրանք 4: Անհրաժեշտ է մշակել քոմենթների ավելացման համակարգ

a. Index.html էջում ունենք ֆորմ, որտեղ օգտագործողը կարող է ավելացնել քոմենթներ
   i. http://prntscr.com/de2bog
   ii. Օգտագործողը մուտք է գործում՝ URL -ում ունենալով name/surname պարամետրեր
      1. Օրինակ՝ index.html?name=Hayk&surname=Harutyunyan
   iii. էջում նախ ցույց է տրվում բոլոր նախկինում ավելացված քոմենթները

iv. Եթե օգտագործողը ավելացնում է նոր քոմենթ, ապա անհրաժեշտ է պահել այդ ավելացվածը LocalStorage -ում
    1. Ցանկացած մեկնաբանություն իրենից ներկայացնում է Comment օբյեկտ, որին կարող են բնութագրել տարբեր պարամետրեր
        a. Այդ թվում՝ հեղինակի անուն, ազգանուն, մեկնաբանության պարունակություն
            i. Քոմենթը կարող է պարունակել նաեւ այն ժամանակահատվածը, որում այն արվել է
v. Քոմենթների խումբը կառավարելու համար կօգտագործենք մեկ այլ օբյեկտ, որին կանվանենք App
    1. App - օբյեկտը կարող է ունենալ տարբեր մեթոդներ
        a. getAllComments() - ցույց կտա բոլոր ավելացված քոմենթները
        b. save/read/insert եւ այլն

Առաջադրանք 4: Array-ի prototype -ին ավելացնել sum() ֆունկցիան որը կվերադարձնի զանգվածի գումարը

1. օրինակ՝ var x=[2,8,10,9,5] ; console.log(x.sum()) կվերադարձնի 10

Առաջադրանք 5: Կասենք x1 օբյեկտը հավասար է x2 օբյեկտին, եթե հավասար են նրանց բոլոր դաշտերը

★ Օրինակ՝
★ p1 = {name: "A", surname: "B"} օբյեկտը հավասար է p2={name: "A", surname: "B"} օբյեկտին
    - Object -ի prototype -ին ավելացնել equals() ֆունկցիան որը կվերադարձնի true, եթե օբյեկտները հավասար են եւ false՝ հակառակ դեպքում
        - Ֆունկցիան պետք է կանչվի հետևյալ կերպ
            - var s = p1.equals(p2); console.log(s);

Դաս 2: Գրաֆիկան JavaScript-ում, Canvas

Առաջադրանք 1:էջում կլինի canvas -ը, 1 select, 3 input դաշտ և գույնը ընտրելու համար նախատեսված input :

- ★ select -ից ընտրում ենք պատկերի տեսակը քառակուսի կամ շրջան
- ★ առաջին 2 ինփութներից կնշենք նրա դիրքի կոորդինատները (x,y) , իսկ վերջին ՝ 4րդ ինփութ դաշտից կընտրենք այդ պատկերի չափը (քառակուսու համար դա կլինի կողմի երկարությունը, շրջանի դեպքում՝ շառավիղը)
- ★ Գույների դաշտից կընտրվի պատկերի fillStyle -ը :
- ★ Draw կոճակին սեղմելիս այդ պատկերը կհայտնվի canvas -ում :
- ★ էջը թարմացնելիս այս պատկերները պետք է չանհետանան իրենց դիրքերից:

Առաջադրանք 2:Ստեղծել Agar.io անիմացիան:

- ★ Canvas-ում Ունենք մի շրջան, որը անընդմեջ շարժվում է առանց կանգ առնելու :
- ★ Canvas-ի պատերին դիպչելիս՝ վերջինս փոխում է իր ուղղությունը՝ շարժվելով հակառակ ուղղությամբ :
- ★ Այս ընթացքում Canvas-ի մեջ՝ պատահական դիրքում հայտնվում է մեկ այլ , ավելի փոքր անշարժ շրջան : Երբ Մեր շարժվող շրջանը հպվում է անշարժ-ին այն մեծանում է , փոքրը մեկ այլ պատահական դիրքում է հայտնվում, իսկ Canvas -ի չափերը մեծանում են :

Առաջադրանք 3:էջում canvas - ով ստեղծել մեկ գնդակ, այն պետք է շարժվի տարբեր ուղղություններով,եթե բախվում է canvas - ի պատերին, ուղղությունը փոխել

Առաջադրանք 4:Canvas-ի դաշտի վրա երբ կկատարվի mousemove event-ը և մկնիկը կտեղաշարժենք մկնիկի հետագծով ներկել:

Առաջադրանք 5:

Առաջադրանք 6:

Առաջադրանք 7:

Առաջադրանք 8:

Առաջադրանք 9:

Առաջադրանք 10:

Դաս 3/4: OOP App-ների մշակում (localStorage, sessionStorage, JSON)

Առաջադրանք 1: Index.html էջում կունենանք 2 input դաշտ և մեկ կոճակ :

- ★ input ում օգտատերը կմուտքագրի իր անունը և ազգանունը : Երբ վերջինս կսեղմի կոճակին պետք է պահպանել նրա տվյալները localStorage -ում և նրան տեղափոխել profile.html էջը որտեղ h1 թեգի մեջ գրված կլինի ողջույնի որևէ խոսք Օրինակ. “Welcome Sandra Johannson”
- ★ Եթե օգտատերը փակի browser -ը և նորից մուտք գործի index.html նրան անմիջապես տեղափոխել profile.html :
- ★ Profile.html -ում կլինի logout կոճակ, որին սեղմելիս նա կհայտնվի inde.html -ում և localStorage -ից կջնջվի նրա տվյալը :
- ★ Եթե օգտատիրոջ տվյալները բացակայում են localStorage -ից և նա փորձում է մուտք գործել profile.html ապա անհապաղ տեղափոխել error.html էջը, որտեղ կարմիր տառերով գրված կլինի Access Forbidden :

Առաջադրանք 2: էջում կունենանք video

- ★ Երբ օգտատերը կփակի պատուհանը և ևս մեկ անգամ այն կբացի պետք է video-ն մեկնարկի հենց այն պահից, երբ video-ն կանգնեցվել էր:

Առաջադրանք 3: Ունենք 3 հատ select դաշտ:

- ★ Ամեն մեկում ընտրվում է պատահական անվանում:
- ★ Ցանկացած ընտրության դեպքում պետք է պահել այս 3 ի արժեքները localStorage -ում: Երբ էջը refresh կկատարվի պետք է select - ներում ցուցադրվեն այն արժեքները, որոնք ընտրվել էին նախօրոք:

Առաջադրանք 4:էջին ունենք դիվ, որին կարելի է սեղմել մկնիկով և տեղաշարժել: Երբ էջը կթարմացվի, այս դիվը պետք է չկորցնի իր նախկին դիրքը:

Առաջադրանք 5:

Առաջադրանք 6:

Առաջադրանք 7:

Առաջադրանք 8:

Առաջադրանք 9:

Առաջադրանք 10:

Դաս 2: Lightbox կիրառում

Առաջադրանք 1: index.html էջում տեղադրել 10 նկար, յուրաքանչյուրին սեղմելիս այն կստանա active կլասը, save կոճակին սեղմելիս կտեղափոխվենք view.html էջ, որտեղ կցուցադրվեն ընտրված նկարները lightbox էֆֆեկտով:

view.html - ում որտեղ ցուցադրվում է ընտրված նկարներով slide - ը, ավելացնել Remove և Shuffle button-ները

- Remove քլիկ անելիս, ջնջվում են բոլոր նկարները (նաև localStorage - ից) և տեղափոխվում ենք index.html
- Shuffle քլիկի ժամանակ փոխել նկարների տեղերը