



## Grile MM

Multimedia Multimedia (Academia de Studii Economice din București)

## Grile Multimedia

1.Care dintre urmatoarele metode nu exista implicit in JavaScript?

**document.getElementsByTagName()**

2.Secventa jQuery \$("p.class1,#p1").css("border","solid green"; modifica culoarea chenarului pentru:

**Toate elementele din pagina de tip paragraf(p) avand clasa CSS class1 si elemental din pagina cu id-ul p1.**

3.Echivalentul jQuery pentru: document.getElementById('elementid') este:

**\$("#elementid");**

4.jQuery este:

**O biblioteca JavaScript**

5.Care este varianta corecta pentru introducerea in HTML a unui CSS extern?

**<link rel="stylesheet" type="text/css" href="stilulmeu.css">**

6.Care dintre urmatoarele secvente respecta sintaxa CSS?

**body{color:black}**

7.Ce atribut HTML este utilizat pentru specificarea stilurilor CSS inline in cadrul unui document?

**Style**

8.Ce elemente HTML vor fi modificate prin intermediul selectorului img.all?

**toate imaginile care au atributul class="all"**

9.Desenarea unui dreptunghi in canvas se realizeaza cu metoda

**fillRect(0,0,50,70)**

10.Care este tagul pentru introducerea in HTML a unui container pentru crearea grafica?

**<canvas>**

11.Care este sintaxa corecta pentru introducerea in pagina a unui script extern?

**<script type="text/javascript" src="scriptulmeu.js">**

12.Care este metoda corecta pentru scriere in canvas?

**fillText(text,x,y)**

13.Elementul HTML folosit pentru reprezentarea unei linii in cardul unui tabel este?

**Tr**

14.Elementul HTML utilizat pentru introducerea unui element de grafica raster in cadrul unei pagini este:

**Svg**

15.Stabilirea culorii rosu pentru desenarea urmatoarei linii pe contextual graphic c asociat unui element canvas se realizeaza prin:

**c.strokeStyle="red";**

16.Scrieti functia JavaScript care primeste ca parametru un vector de obiecte cu proprietatea nume,prenume.Functia trebuie sa adauge in corpul paginii datele sub forma de lista HTML cu formatul nume,prenume

```
<body>

<button type="button" id="buton1">Generare Lista</button>
<p id="p1"></p>
<script>

function creare_lista()
{
    var Persoana={
        nume:"alexandru",
        prenume:"popescu",
        nume_prenume:function(){return "Nume:"+this.nume+" "+"Prenume:
"+this.prenume+" ";}
    }
    var lista=Persoana.nume_prenume()+"<ul>";
    lista+="</ul>";
    document.getElementById("p1").innerHTML=lista;
}

    document.getElementById("buton1").onclick=creare_lista;

</script>
</body>
```

17.Scrieti codul in JS care deseneaza un text in canvasul cu id canvasul\_meu in pozitia in care s-a apasat butonul dreapta mouse.

18.Stabilirea culorii rosu pentru umplerea urmatoarei figure desenate pe contextual graphic c asocait unui element canvas se realizeaza prin:

**c.fillStyle="red";**

19.Formatul de stocare svg este specific stocarii:

**Imaginilor vectoriale**

20.Compresia video are la baza eliminarea redundantei:

**Intra si inter-cadru**

21.Sunetul este definit ca:

**O vibratie care se propaga printr-un material**

22.Compresia jpeg nu utilizeaza:

**Compresia LZW**

23.Care dintre urmatoarele caracteristici NU este specifica imaginii vectoriale

**Imaginea este dependenta de scara de vizualizare**

`<svg>`

`<rect id="r1" x="10" y="10" width="10" height="10"></rect>`

`<rect id="r2" x="10" y="10" width="10" height="10"></rect>`

`</svg>`

1.Pozitionarea dreptunghiului r2 imediat sub dreptunghiul r1 se poate realiza prin urmatoarea secventa:

**`Document.querySelector('#r2').setAttribute('y','20')`**

2.Fie **a** o referinta la un element HTML de tip audio.Pozitionare la jumatatea secventei audio se realizeaza prin:

**a.currentTime=a.duration?a.duration/2:0**

3. Care dintre urmatoarele valori reprezinta culoarea rosu in modelul de culoare HSL  
**(0,100%,50%)**

4.Formatul de stocare a imaginii JPEG este specific

**Imagini de tip raster**

5. In cadrul compresiei JPEG pentru imagini este utilizata

**Compreisa Cosinus Discreta**

6. Care dintre urmatoarele formate este utilizat pentru stocarea sunetului necomprimat?

**WAV**

7.Formatul de stocare SVG este specific stocarii:

**Imagini vectoriale**

8.Care este apelul corect pentru preluarea intr-un obiect a canvas-ului?

**Document.getElementById("test")**

9.Ce element este utilizat pentru introducerea unei secvente CSS in interiorul unui document

**Link**

10.Care dintre urmatorii selectori CSS selecteaza elemental cu id-ul "maria: din document?

**#maria**

11. RGB este un model:

**Aditiv**

12. Stergerea unei suprafete dreptunghiulare de pe un element de tip <canvas> se poate realiza cu urmatoarele metode aferente Canvas RenderingContext2d.

`clearRect(x,y,width,height)`

13.Care dintre urmatoarele raspunsuri relecta continutul vectorului data, obtinut dupa cum urmeaza, in cazul imaginii de mai jos

`Const context=canvas.getContext(„2d”)`

`Const imageData=context.getImageData(0,0,2,0)`

`Const data=imageData.data;`

R: [255,0,0,255; 0,0,0,255; 255,255,255,255; 197,52,144,255]

14. Coordonatele in cadrul unui element de tip canvas sunt calculate raportat la:

Coltul stanga-sus

15. Care dintre urmatoarele afirmatii nu este adevarata in cazul graficii vectoriale:

Calitatea lor este afectata atunci cand sunt scalate

16. Care dintre urmatoarele elemente SVG este utilizat pentru a grupa alte elemente (linii,texte etc)?

`G /?group??`

17. Care dintre urmatoarele formate se utilizeaza in cazul imaginilor raster?

Gif

18. Care dintre urmatoarele attribute nu este utilizat in mod obisnuit pentru a caracteriza suntetul:

intensitate

Care dintre următoarele atribute nu este utilizat în mod obișnuit pentru a caracteriza sunetul:

Alegeți o opțiune:

- ☐ a. amplitudine
- ☐ b. durată
- ☐ c. frecvență
- ☒ d. intensitate

19. Desenarea unei imagini pe un element de tip <canvas> se poate realiza cu următoarele metode aferente CanvasRenderingContext2D:

`Void ctx.drawImage(image,dx,dy);`

20. C este o referinta la un canvas de latime 20 pixeli si inaltime 10 pixeli

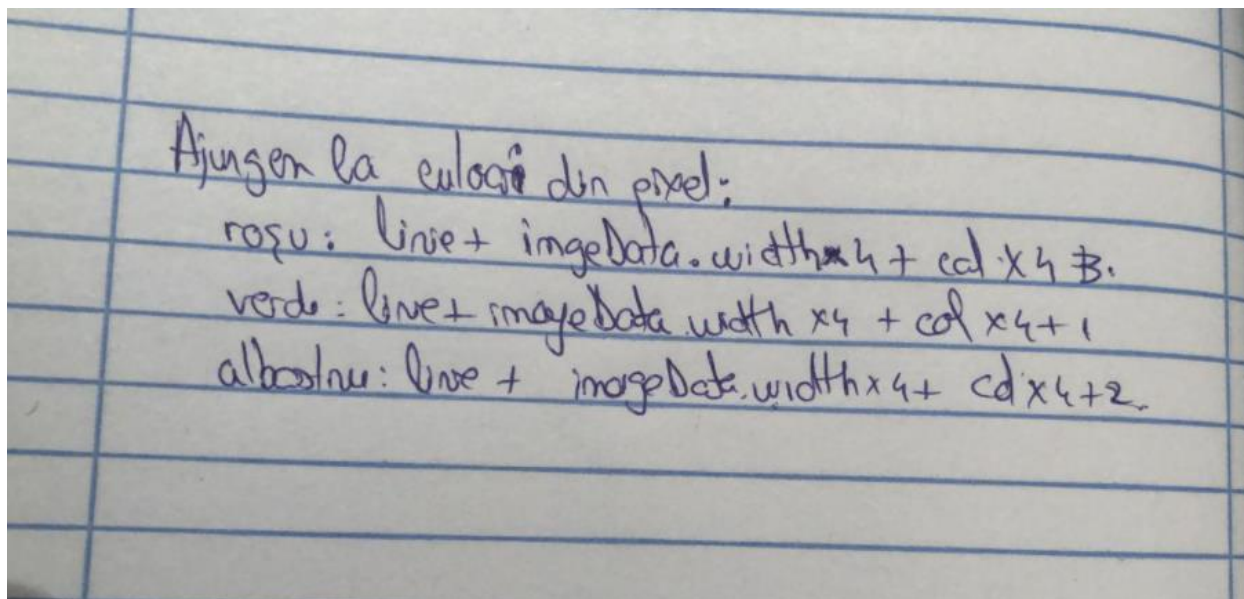
Fie codul:

`Let context=C.getContext(„2D”);`

`Let imageData=context.getImageData(0,0,W,H);`

Componenta de culoare verde pentru pixelul de la linia 3, coloana 4 este accesata prin:





Pt linia 1, coloana 0

`imageData.data[81]`

21. Care dintre urmatoarele valori reprezinta culoarea rosu in modelul de culoare HSL?

(0,100%,50%)

22. In cadrul compresiei JPEG pentru imagini este utilizata:

Transformata cosinus discreta

23. Formatul de stocare a imaginii JPEG este specific:

Imaginilor de tip raster

24. Pierderea de informatie in cazul compresiei JPEG este influentata de:

Modalitatea de aplicare a transformarii cosinus discreta

25. Formatul de stocare SVG este specific stocarii:

Imaginilor vectoriale

26. Care este apelul corect pentru preluarea intr-un obiect a canvas-ului?

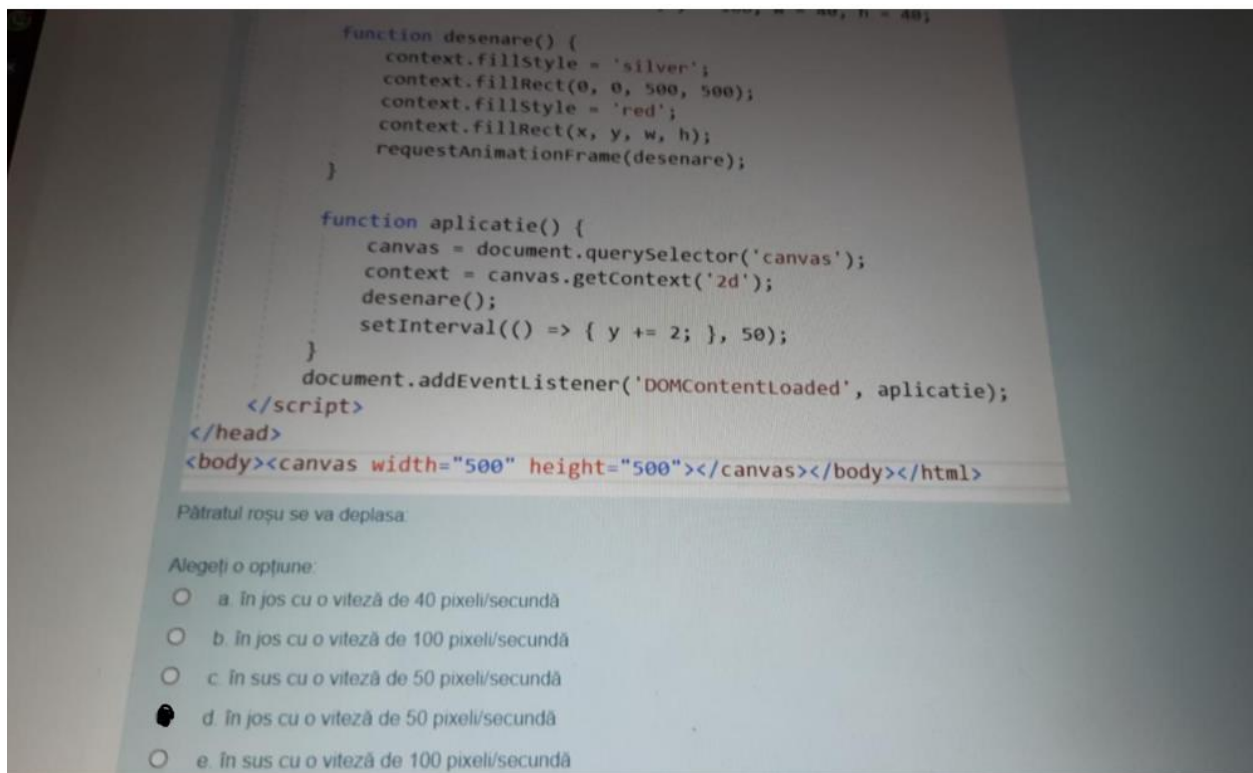
Var b=document.getElementById(„test”)

27.Ce element HTML este utilizat pentru introducerea unei secvente CSS in interiorul unui document?

Link

28. Patrutul rosu se va deplasa ::

In jos cu o viteza de 50 pixeli/secunda



29. Formatul EPS este special proiectat pentru a stoca:

Grafica vectoriala

30. Reducerea dimensiunii fisierelor audio se poate realiza prin:

Ținând cont de mascarea frecvențelor și/sau ținând cont de mascarea temporală

31. Grafica raster(bitmap) folosește o reprezentare sub forma de :

Matrice de puncte

32. Setarea culorii folosite pentru desenarea formelor cu umplere pe un <canvas>, pentru care putem accesa contextul grafic prin variabila c, se poate face după cum urmează:

```
c.fillStyle="red";
```

33. Următoarele proprietăți sunt disponibile pentru CanvasRenderingContext2D (canvas.getContext(„2d”)):

strokeStyle

lineWidth

textAlign

font

34. Care dintre următoarele NU sunt formate de stocare pentru imagini vectoriale?

PNG

35. Desenarea conturului unui dreptunghi pe un element de tip <canvas> se poate realiza cu următoarele metode aferente CanvasRenderingContext2D:

```
strokeRect(x,y,width,height)
```

36. Care dintre urmatoarele attribute NU este utilizat in mod obisnuit pe un element audio>

showControls

37. Latimea reala a fisierului de tip imagine (ex:\*.jpeg) afisat in cadrul unui element <img>, exprimata in pixeli, poate fi obtinuta prin intermediul urmatoarei proprietati aferente HTMLImageElement.

naturalWidth

38. Metoda corecta pentru desenarea unui text pe un element de tip canvas:

fillText(text,x,y);

39. Care dintre urmatoare nus este o metoda de transformare disponibila pentru CanvasRenderingContext2D (canvas.getContext(„2d”));

Deplasare

40. In CSS prin table#unu se selecteaza:

Tabelul cu id-ul unu

41. Desenarea unui cerc folosind un context grafic asociat unui element canvas se realizeaza prin intermediul functiei:

Arc

42. Grafica vectoriala foloseste o reprezentare sub forma de

Functii matematice

43. Care dintre urmatoarele valori nu este o valoare acceptabila pentru proprietatea CSS position?

Float

1 Care din urmatoarele metode nu exista mplicit in JavaScript?

- A. document.getElementsByTagName();
- B. document.getElementById();
- C. document.getElementByTagName();
- D. document.getElementsByName();

2 Care este tagul pentru introducerea in HTML a unui container pentru crearea grafica?

- a. <paint>
- b. <style>
- c. <canvas>
- d. <graphics>

3. Care este metoda corecta pentru scriere in canvas?

- a. font(x,y,text)
- b. fillText(text,x,y)
- c. fillstyle(text,x,y)
- d. paint(text,x,y)

4. Ce element HTML este utilizat pentru specificarea stilurilor CSS externe pentru un document?

- a. CSS
- b. link
- c. style
- d.class

5.Care este varianta corecta de includere a unei imagini ase.jpg in cadrul unui document HTML?

- A. <image src="ase.jpg" alt="Exemplu imagine">
- B. <img alt ="Exemplu imagine ">ase.jpeg</img>
- C.< img src="ase.jpg" alt="Exemplu imagine">
- D.<img href="ase.jpg" alt="Exemplu imagine">

6.Elementul HTML folosit pentru reprezentarea unui rand in cadrul unui table este

A. tr

B. table

C. cell

D. td

E. tc

7. Elementul HTML utilizat pentru introducerea unui element de grafica vectoriala in cadrul unei pagini este

A. graph

B. svg

C. canvas

D. vector

8. Stabilirea culorii rosu pentru umplerea urmatoare figure desenate pe contextual graphic c asociat unui element canvas se realizeaza prin

A. c.strokeStyle("red")

B. c.fillStyle="red"

C. c.color="red"

D. c.stroleStyle="red"

9. Formatul de stocare SVG este specific stocarii:

A. imaginilor necomprimate

B. fisierelor de tip sunet

C. imaginilor vectoriale

D. imaginilor de tip raster

10. Compresia video are la baza eliminarea redundantei:

A. doar audio

B. intra si inter-cadru

C. doar intra-cadru

D. doar inter-cadru

11. Sunetul este definit ca:

- A. Energie magnetica intr-un mediu elastic
- B. O vibratie care se propaga printr un mediu material
- C. Energie electrica propagate prin vid
- D. Energie electrica statica transmisa prin orice mediu

12. Compresia JPEG NU utilizeaza:

- A. compresia RLE
- B. compresia Huffman
- C. transformata cosinus discrete
- D. compresia LZW

13. Care dintre urmatoarele caracteristici NU este specifica imaginilor vectoriale

- A. mentine semnificatia imaginii
- B. obiectele componente sunt descrise matematic
- C. fisierul imagine este mic
- D. imaginea este dependenta de scara de vizualizare

14. Desenarea unui dreptunghi in canvas se realizeaza cu metoda

- A. `paint(x,y,a,b)`
- B. `fillRect(0,0,50,70)`
- C. `fillStyle(0,0,10,30)`
- D. `square(x,y,a,b)`

15. Care este apelul corect pentru preluarea unui obiect al canvasului?

- A. `var b = canvas.getElementById("test")`
- B. `var b = document.getElementById("test")`
- C. `var b = document.getCanvas("test")`
- D. `var b = document.getContext("test")`

16. Care este sintaxa corecta pentru introducerea in pagina a unui script extern?

`<script type="text/javascript" src="scriptulmeu.js">`

17. Ce atribut este utilizat pentru specificarea stilurilor CSS intr un element HTML?

A. css

B. link

C. style

D. class

18. Care este varianta corecta pentru introducerea in HTML a unui css extern?

A. <link rel="stylesheet" type="text/css" href="stilulmeu.css">

19. Care dintre urmatoarele secvente respecta sintaxe CSS?

A. body:color=black

B. body(color:black)

C. (body, color.black)

D. (body.color=black(body)

20. Ce elemente HTML vor fi modificate prin intermediul selectorului img.all?

A. Toate imaginile din cadrul documentului

B. toate elementele care au atributul id="img.all"

C toate elementele care au atributul class="img.all"

D.toate imaginile care ai atributul class="all"

20. jQuery este:

a) un limbaj de programare

b) un limbaj de acces la baza de date

c) o biblioteca JavaScript

d) un tag

21. Desenarea unui cerc folosind un context graphic asociat unui element canvas se realizeaza prin intermediul functiei:

a) circle



b) arc

c) eclipse

d) closedPath

22. Care din urmatoarele elemente svg este utilizat pentru gruparea unui set de elemente fara a le afisa?

a) rect

b) defs

c) g

d) svg

23. Care dintre urmatoarele valori nu este o valoare acceptabila pentru proprietatea CSS position?

a) static

b) relative

c) fixed

d) absolute

e) float

24. Elementul HTML utilizat pentru introducerea unui element de grafica raster in cadrul unei pagini este

A. graph

B. svg

C. canvas

D. vector

Au fost 15 intrebari in total

Which is the HTML element used to draw a raster graphic on a web page?

Alegeți o opțiune:

- ☐ a. *image*
- ☐ b. *canvas*
- ☐ c. *svg*
- ☐ d. *raster*

NAVIGARE ÎN TEST

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

[Înceiați încercarea ...](#)

Timp rămas 0:04:52

b

Which of the following tags does include the "stud.jpg" image, in an HTML document?

Alegeți o opțiune:

- ☐ a. `<img alt="Student image">stud.jpg</img>`
- ☐ b. `<image src="stud.jpg" alt="Student image">`
- ☒ c. ``
- ☐ d. `<img href="stud.jpg" alt="Student image">`

cea bifata

Which of the following CSS selectors refers an element with "my" id, in an HTML document?

Alegeți o opțiune:

- ☐ a. `#my`
- ☐ b. `body [my]`
- ☐ c. `.my`
- ☐ d. `my`

#my

Which colour model does use a double-cone representation for its colour space?

Alegeți o opțiune:

- ☐ a. HLS
- ☐ b. RGB
- ☐ c. HSB
- ☐ d. CMYK

HLS

What is the output of the following chunk of HTML & CSS code?

```
<div id="test">
  <span>Text</span>
</div>
<style>
div#test span { color: green; }
div span { color: blue; }
span { color: red; }
</style>
```

Alegeți o opțiune:

- ☐ a. It colours the text in green.
- ☒ b. It colours the text in blue..
- ☐ c. It colours the text in red..
- ☐ d. It generates an error.

It colours the text in green

Check all arguments accepted by the DrawImage JavaScript function:

Alegeți una sau mai multe opțiuni:

- ☐ a. a canvas element
- ☐ b. an image
- ☐ c. a SVG element
- ☐ d. a video element

canvas, image, video

Check all the processing operations involved in JPEG compression algorithm:

Alegeți una sau mai multe opțiuni:

- ☐ a. discrete cosine transform
- ☐ b. block splitting
- ☐ c. LZW coding
- ☐ d. quantization

transformare cosinus discrete si cuantizare

Select ALL statements depicting DOM, from the following:

Alegeți una sau mai multe opțiuni:

- ☐ a. DOM is the acronym for Document Object Model.
- ☐ b.

DOM provides the way to programmatically access HTML structure in JavaScript

- ☐ c.  
HTML of every web page is turned into a DOM representation by the browser
- ☐ d. The attributes of DOM nodes cannot be changed dynamically in JavaScript

a, b, c

Select all characteristics of a raster image, from the following.

Alegeți una sau mai multe opțiuni:

- ☐ a. The raster image takes into account image semantics.
- ☐ b. The raster image is figured as a matrix of points.
- ☐ c. The raster image is memory intensive.
- ☐ d. Raster image file size depends on the image size and the colour depth

b,d

Select ALL solutions for incorporating CSS in HTML.

Alegeți una sau mai multe opțiuni:

- ☐ a. defined within a style block, in the head section of a web page
- ☒ b. coded in a separate file
- ☐ c. Coded in the body of the web page

tag link, tag style (in head), atribut

style pt inline (in body) c e ambiguu

Select ALL correct statements referring to the FlexBox model.

Alegeți una sau mai multe opțiuni:

- ☐ a. It uses grid-column and grid-row properties to change the size of items.
- ☐ b. It is activated using display property
- ☐ c. It is used to change the horizontal or vertical layout of HTML elements, in a web page
- ☐ d. It can be used to programmatically change the order of the HTML elements.
- ☐ e. It is appropriated for creating a responsive web application

b, d, e, c? e ambiguu

Select ALL primary colours of the **subtractive colour model**.

Alegeți una sau mai multe opțiuni:

- ☒ a. yellow
- ☐ b. green
- ☐ c. red
- ☒ d. cyan
- ☐ e. blue
- ☒ f. magenta

cyan, magenta, yellow

What purpose has setTimeout JavaScript function?

Răspuns:

metoda setTimeout apeleaza o functie

sau evalueaza o expresie dupa un numar specificat de milisecunde. 1000ms=1s si functia ce va fi executata va fi rulata o singura data. Pt a repeta executia functiei putem folosi metoda setInterval. Pt a preveni functia din a rula putem folosi metoda clearTimeout

setTimeout(function(){ alert("Hello"); }, 3000); - afiseaza o fereastra de alerta cu textul hello dupa 3 sec

What is HyperVideo?

Răspuns:

## BARBULESCU

1. Care dintre urmatoarele valori reprezinta culoarea **rosu** in modelul de culoare RGB?
  - a) #0000FF
  - b) #00FF00
  - c) #FF0000 -> (255,0,0)
  - d) #000000
2. Care dintre urmatoarele valori reprezinta culoarea **albastru** in modelul de culoare RGB?
  - a) #000000
  - b) #FF0000
  - c) #00FF00
  - d) #0000FF -> (0,0,255)
3. Care dintre urmatorii selectori CSS selecteaza elementul cu id-ul „maria” din document?
  - a) .maria
  - b) body[maria]
  - c) maria
  - d) #maria
4. Care este apelul corect pentru preluarea intr-un obiect a canvas-ului?
  - a) var b = canvas.getElementById(“test”)
  - b) var b = document.getElementById(“test”)
  - c) var b = document.getcanvas(“test”)
  - d) var b = document.getcontext(“test”)
5. Formatul de stocare SVG este specific stocarii:
  - a) fisierelor de tip sunet
  - b) imaginilor necomprimate
  - c) imaginilor vectoriale
  - d) imaginilor de tip raster
6. In CSS prin table#unu se selecteaza:
  - a) Prima linie din table
  - b) Tabelul cu id=unu
  - c) Tabelul cu o linie
  - d) Un buton
7. Formatul de stocare a imaginii JPEG este specific:
  - a) imaginilor vectoriale
  - b) imaginilor de tip raster

- c) fişierelor de tip sunet si video
  - d) imaginilor necomprimate
8. Desenarea unui cerc folosind un context grafic asociat unui element canvas se realizeaza prin intermediul functiei:
- a) `closedPath`
  - b) `circle`
  - c) `ellipse`
  - d) `arc`
9. Elementul HTML utilizat pentru introducerea unei imagini in cadrul unei pagini este:
- a) `image`
  - b) `class`
  - c) `img`
  - d) `graph`
10. Care dintre urmatoarele valori reprezinta culoarea verde in modelul de culoare RGB?
- a) `#000000`
  - b) `#0000FF`
  - c) `#00FF00` -> (0,255,0)
  - d) `#FF0000`
11. Care dintre urmatorii selectori CSS selecteaza elementul cu id-ul „ion” din document?
- a) `body[ion]`
  - b) `#ion`
  - c) `ion`
  - d) `.ion`
12. Ce element HTML este utilizat pentru introducerea unei secvente CSS in interiorul unui document?
- a) `link`
  - b) `class`
  - c) `css`
  - d) `style`
13. Grafica vectoriala foloseste o reprezentare sub forma de:
- a) functii matematice
  - b) matrice de puncte
  - c) transformari Fourier
  - d) coeficienti polinomiali

14. Care dintre urmatoarele modele de culoare este de tip aditiv?
- a) CMYK
  - b) RGB
  - c) HSL
  - d) HSV
15. Grafica raster (bitmap) foloseste o reprezentare sub forma de:
- a) matrice de puncte
  - b) coeficienti polinomiali
  - c) functii matematice
  - d) transformari Fourier
16. Elementul HTML folosit pentru reprezentarea unui paragraf este:
- a) paragraph
  - b) li
  - c) section
  - d) p



## AU RASPUNSURILE DE PE PLATFORMA!

1. Care dintre urmatoarele NU sunt formate de stocare pentru imagini vectoriale?

a) **PNG**

b) SHP

c) DXF

d) EPS

2. Metoda corecta pentru desenarea unui text pe un element de tip canvas este:

a) `appendText(text, x, y);`

b) `writeText(text, x, y);`

c) `drawText(text, x, y);`

d) **`fillText(text, x, y);`**

3. Care dintre urmatoarele attribute nu este utilizat in mod obisnuit pe un element audio?

a) `loop`

b) **`showControls`**

c) `volume`

d) `autoplay`

5. CMYK este un model:

a) aditiv

b) **subtractiv**

c) nu este un model utilizat in multimedia

d) multiplicativ

6. Care dintre urmatoarele nu este o metoda de transformare disponibila pentru `CanvasRenderingContext2D(canvas.getContext("2d"))`:

a) rotatie

b) translatie

c) **deplasare**

d) scalare

4. Care dintre următoarele răspunsuri reflecta conținutul vectorului data, obținut după cum urmează, în cazul imaginii de mai jos.

```
const context = canvas.getContext("2d")
```

```
const imageData = context.getImageData(0,0,2,2)
```

```
const data = imageData.data;
```

4 întrebare

Corect

Marcat 1,00 din 1,00

▼ Întrebare cu flag

Care dintre următoarele răspunsuri reflectă conținutul vectorului data, obținut după cum urmează, în cazul imaginii de mai jos.

```
const context = canvas.getContext("2d")
const imageData = context.getImageData(0,0,2,2)
const data = imageData.data;
```

Pixel 0 Pixel 1

Alegeți o opțiune:

- ☐ a. [255,255,255,255, 197,52,144,255, 255,0,0,255, 0,0,0,255]
- ☐ b. [255,0,0,255, 255,255,255,255, 0,0,0,255, 197,52,144,255]
- ☒ c. [255,0,0,255, 0,0,0,255, 255,255,255,255, 197,52,144,255] ✓
- ☐ d. [255,0,0,255, 0,0,0,255, 255,255,255,255, 197,52,144,255]

The correct answers are: [255,0,0,255, 0,0,0,255, 255,255,255,255, 197,52,144,255], [255,0,0,255, 0,0,0,255, 255,255,255,255, 197,52,144,255]

5 întrebare

7. Care dintre următoarele afirmații nu este adevărată în cazul graficii vectoriale:

- a) conțin forme geometrice precum puncte, linii, curbe etc.
- b) un set de comenzi este folosit pentru a desena imaginea
- c) calitatea lor este afectată atunci când sunt scalate**
- d) pentru imagini simple, acestea ...

8. Următoarele proprietăți sunt disponibile pentru CanvasRenderingContext2D(canvas.getContext("2d")):

- a) textAlign**
- b) strokeStyle**
- c) font**
- d) lineWidth**

9. Desenarea conturului unui dreptunghi pe un element de tip <canvas> se poate realiza cu urmatoarele metode aferente CanvasRenderingContext2D:

a) `paintRect(x, y, width, height)`

**b) `strokeRect(x, y, width, height)`**

c) `drawRect(x, y, width, height)`

d) `rect(x, y, width, height)`

10. Care dintre urmatoarele metode nu exista implicit in JavaScript?

a) `document.getElementsByName();`

**b) `document.getElementByTagName();`**

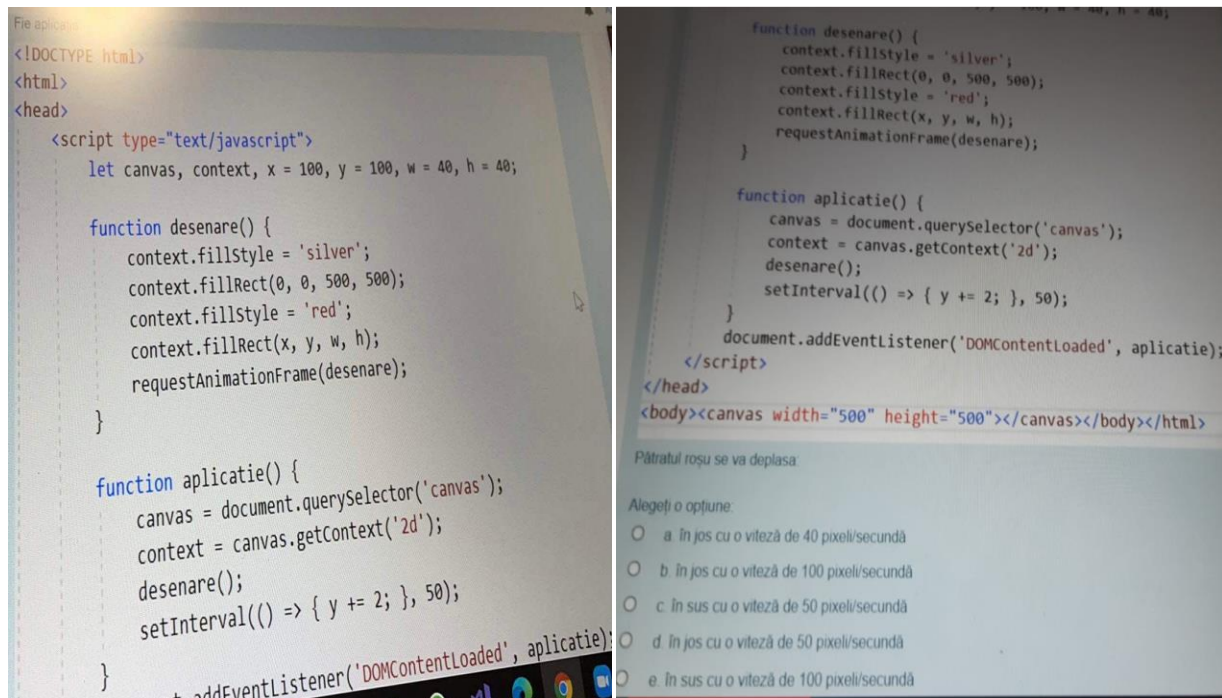
c) `document.getElementsByTagName();`

d) `document.getElementById();`

**!!! DAR exista `document.getElementsByTagName()`**

**CELE DE MAI SUS AVEAU RASPUNSURILE DE PE PLATFORMA!**

## IONITA



Patratul rosu se va deplasa:

- a) in sus cu o viteza de 100 pixeli/secunda
- b) in jos cu o viteza de 50 pixeli/secunda
- c) in jos cu o viteza de 100 pixeli/secunda
- d) in jos cu o viteza de 40 pixeli/secunda
- e) in sus cu o viteza de 50 pixeli/secunda

Care dintre urmatoarele valori reprezinta culoarea rosu in modelul de culoare HSL?

- a) (0, 100%, 0%)
- b) (120, 100%, 100%)
- c) (0, 100%, 50%)
- d) (120, 100%, 0%)
- e) (240, 0%, 0%)

Fie **a** o referinta la un element HTML de tip audio. Pozitionarea la jumatatea secventei audio se realizeaza prin:

- a) `a.time = a.duration * 0.5`
- b) `a.currentTime = 0.5`
- c) `a.time = a.duration ? a.duration * 0.5 : 0` ?????????????
- d) `a.currentTime = a.duration ? a.duration / 2 : 0`
- e) `a.position = 1 / 2`

**C** este o referinta la un canvas de latime 20 pixeli si inaltime 10 pixeli.

Fie codul:

```
let context = C.getContext("2d");
```

```
let imageData = context.getImageData(0, 0, W, H);
```

Componenta de culoare verde pentru pixelul de la linia 3, coloana 4 este accesata prin (numararea liniilor si coloanelor incepe de la 0):

- a) `imageData.data[80]`
- b) `imageData[97]`
- c) `imageData.data[81]`
- d) `imageData[81]`
- e) `imageData.data[97]`

Fie documentul HTML:

```
<!DOCTYPE html>

<html>

<head></head>

<body>

  <svg>

    <rect id="r1" x="10" y="10" width="10" height="10"></rect>

    <rect id="r2" x="10" y="10" width="10" height="10"></rect>

  </svg>

</body></html>
```

Pozitionarea dreptunghiului **r2** imediat sub dreptunghiul **r1** se poate realiza prin urmatoarea secventa JavaScript:

- a) document.querySelector('#r2').setAttribute('y', '10');
- b) document.querySelector('#r1').y = 20;
- c) document.querySelector('r2').setAttribute('y', '20');
- d) document.querySelector('#r1').setAttribute('x', '20');
- e) document.querySelector('#r2').setAttribute('y', '20');

## GRILE ADUNATE

1. Care dintre algoritmi de mai jos este doar fara pierdere de informatie?  
a) **FLAC**  
b) Audio Mpeg Layer 3  
c) WMA  
d) Audio Mpeg-4
2. Care din cele de mai jos NU este cerinta pentru compresia video  
a) Acces random la secvente  
b) Posibilitate editare  
c) Corectie/evitare erori  
d) **Generarea cadrelor intermediare**
3. Care din cele de mai jos NU este cerinta pentru compresia video  
a) Reverse playback  
b) **Compresia sunetului fara pierdere de informatie**  
c) Sincronizare audio video  
d) Viteza mare codare/decodare
4. Compresia video are la baza eliminarea redundantei:  
a) doar audio  
b) doar inter-cadru  
c) **intra si inter-cadru**  
d) doar intra-cadru
5. In cadrul compresiei JPEG pentru imagini este utilizata:  
a) Transformata Fourier Discreta  
b) Compresia Lempel-Ziv-Welch  
c) **Transformata Cosinus Discreta**  
d) Transformata Sinus Discreta
6. Care este atributul care face ca sunetul in tag-ul audio sa ruleze continuu pana cand este oprit de catre utilizator?  
a) controls  
b) **loop**

7. Care este elementul introdus in html5 care permite redarea nativa a fisierelor de sunet?
- a) <sound>
  - b) <audio> ?
  - c) <video>
  - d) <play>
8. Care informatie NU este adevarata cu privire la Group of Pictures (GOP)
- a) Sunt secvente repetitive I,P,B
  - b) Reprezinta cadre in ordinea aparitiei
  - c) Nu sunt comprimate
  - d) Incep mereu cu care I
9. Pentru ce se utilizeaza CSS in contextul HTML5?
- a) Color style sheets
  - b) Clear style sheets
  - c) Computer style sheets
  - d) Cascading style sheets
10. Care dintre urmatoorii selectori CSS selecteaza elementul cu id-ul „ion” din document?
- a) body[ion]
  - b) #ion
  - c) ion
  - d) .ion
11. Care dintre urmatoarele formate este utilizat pentru stocarea sunetului necomprimat?
- a) JPG
  - b) WAVE
  - c) MP3
  - d) BMP
12. Sunetul este definit ca:
- a) energie magnetica intr-un mediu elastic
  - b) o vibratie care se propaga printr-un mediu material
  - c) energie electrica propagata prin vid
  - d) energie electrica statica transmisa prin orice mediu



13. Ce atribut este utilizat pentru specificarea stilurilor CSS intr-un element HTML?

- a) css
- b) link
- c) style
- d) class

14. Care este variabila corecta de includere a imaginii ase.jpg in cadrul unui document HTML?

- a) <image src="ase.jpg" alt="Exemplu imagine">
- b) <img alt="Exemplu imagine">ase.jpg</img>
- c) 
- d) <img href="ase.jpg" alt="Exemplu imagine">

(( Care este VARIANTA corecta de includere a imaginii ase.jpg in cadrul unui document HTML? )))

15. Elementul HTML folosit pentru reprezentarea unui rand in cadrul unui tabel este:

- a) tr
- b) table
- c) cell ---- mai e o varianta cu row aici
- d) td
- e) tc

<tr> de la table row

!!! <td> = celula cu date, celula standard si <th> este celula din header

(( Elementul HTML folosit pentru reprezentarea unei linii in cadrul unui tabel este: )))

16. Elementul HTML utilizat pentru introducerea unui element de grafica raster in cadrul unei pagini este:

- a) graph
- b) svg
- c) canvas
- d) vector

17. Stabilirea culorii rosu pentru **umplerea** urmatoarei **figuri desenate** pe contextul grafic c asociat unui element canvas se realizeaza prin:

- a) c.strokeStyle("red")
- b) c.fillStyle = "red"
- c) c.color = "red"
- d) c.strokeStyle = "red"

18. Stabilirea culorii rosu pentru **desenarea** urmatoarei **linii** pe contextul grafic c asociat unui element canvas se realizeaza prin:

- a) c.strokeStyle("red")
- b) c.fillStyle = "red"
- c) c.color = "red"
- d) c.strokeStyle = "red"

19. Care dintre urmatoarele metode NU exista implicit in JavaScript?

- a) document.getElementsByName();
- b) document.getElementById();**
- c) document.getElementsByTagName();
- d) document.getElementById();

**!!! DAR exista document.getElementsByTagName()**

20. Secventa jQuery `$("#p.class1, #p1").css("border", "solid green");` modifica culoarea chenarului pentru:

- a) toate elementele de tip paragraf (p) din pagina care au id-ul #p1 si clasa CSS class1
- b) selectorul jQuery este incorect
- c) toate elementele de tip paragraf (p) din pagina care au fie id-ul #p1, fie clasa CSS
- d) toate elementele din pagina de tip paragraf (p) avand clasa CSS class1 si elementul din pagina cu id-ul p1

21. Echivalentul jQuery pentru: `document.getElementById('elementId')` este:

- a) `$("#elementId");`
- b) `$(".elementId");`
- c) `$("#elementId");`
- d) `$("<elementId>");`

22. jQuery este:

- a) Un limbaj de programare
- b) Un limbaj de acces la baza de date
- c) O biblioteca JavaScript
- d) Un tag

23. Desenarea unui dreptunghi in canvas se realizeaza cu metoda:

- a) `paint(x,y,a,b)`
- b) `fillRect(0,0,50,70)`
- c) `fillStyle(0,0,10,30)`
- d) `square(x,y,a,b)`

24. Care este apelul corect pentru preluarea intr-un obiect a canvas-ului?

- a) `var b = canvas.getElementById("test")`
- b) `var b = document.getElementById("test")`
- c) `var b = document.getcanvas("test")`
- d) `var b = document.getcontext("test")`

25. Care este sintaxa corecta pentru introducerea in pagina a unui script extern?

- a) `<src type="text/javascript" src="scriptulmeu.js">`
- b) `<script type="text/javascript" src="scriptulmeu.js">`
- c) `<script type="text/javascript" href="scriptulmeu.js">`
- d) `<script type="text/javascript" name="scriptulmeu.js">`

26. Care este tagul pentru introducerea in HTML a unui container pentru crearea de grafica?

- a) `<paint>`
- b) `<style>`
- c) `<canvas>`
- d) `<graphics>`

(( Care este tagul pentru introducerea in HTML a unui container pentru crearea grafica? ))

27. Care este metoda corecta pentru scriere in canvas?

- a) `font(x,y,text)`
- b) `fillText(text,x,y)`
- c) `fillstyle(text,x,y)`
- d) `paint(text,x,y)`

28. Formatul de stocare SVG este specific stocarii:

- a) fisierelor de tip sunet
- b) imaginilor necomprimate
- c) imaginilor vectoriale
- d) imaginilor de tip raster

29. Compresia video are la baza eliminarea redundantei:

- a) doar audio
- b) doar inter-cadru
- c) intra si inter-cadru
- d) doar intra-cadru

30. Sunetul este definit ca:

- a) energie magnetica intr-un mediu elastic
- b) o vibratie care se propaga printr-un mediu material
- c) energie electrica propagata prin vid
- d) energie electrica statica transmisa prin orice mediu

31. Compresia JPEG NU utilizeaza:

- a) compresie RLE
- b) compresia Huffman
- c) transformata cosinus discreta
- d) compresia LZW

32. Care dintre urmatoarele caracteristici NU este specifica imaginilor vectoriale:

- a) mentine semnatica imaginii
- b) obiectele componente sunt descrites mathematic
- c) fisierul imagine este mic
- d) imaginea este dependenta de scara de vizualizare

33. Elementul HTML utilizat pentru introducerea unui element de grafica vectoriala in cadrul unei pagini este:

- a) graph
- b) svg
- c) canvas
- d) vector

!!! <canvas> pentru grafica raster

34. Care este varianta corecta pentru introducerea in HTML a unui CSS extern?

- a) `<link rel="stylesheet" type="text/css" href="stilulmeu.css">`
- b) `<style src="stilulmeu.css" />`
- c) `<stylesheet>stilulmeu.css</stylesheet>`
- d) `<script style="stilulmeu.css">`

Care este varianta corecta pentru introducerea in HTML a unui CSS **extern**?

- a) `<link rel="stylesheet" type="text/css" href="stilulmeu.css">`

!!!DAR daca este **intern** avem `<style src="...">`

35. Pierderea de informatie in cazul compresiei JPEG este influentata de:

- a) **alegerea matricei de cuantizare**
- b) calitatea imaginii sursa
- c) parametrii compresiei Huffman
- d) parametrii compresiei RLE
- e) modalitatea de aplicare a transformatei cosinus discreta

36. Ce element HTML este utilizat pentru specificarea stilurilor CSS externe pentru un document?

- a) css
- b) **link**
- c) style
- d) class

1. Grafica vectoriala foloseste o reprezentare sub forma de: => functii matematice
2. Compresia MPEG 1-2 LAYER III foloseste codificarea perceptuala si un model psihoacustic
3. GIF este un format specific => imaginilor de tip raster
4. Care dintre urmatoarele este un format de stocare pentru imagini raster? => JPG
5. Stabilirea culorii pentru culoare urmatoarelor linii din contextul grafic:  
⇒ `context.fillStroke="red"`

37. Care dintre urmatoarele secvente respecta sintaxe CSS?

- a) `body:color=black`
- b) `body(color:black)`
- c) `(body,color.black)`
- d) `(body.color=black(body)`

38. Ce elemente HTML vor fi modificate prin intermediul selectorului `img.all`?

- a) Toate imaginile din cadrul documentului
- b) toate elementele care au atributul `id="img.all"`
- c) toate elementele care au atributul `class="img.all"`
- d) toate imaginile care au atributul `class="all"`

39. Care din urmatoarele elemente svg este utilizat pentru gruparea unui set de elemente fara a le afisa?

- a) `rect`
- b) `defs`
- c) `g`
- d) `svg`

40. Care dintre urmatoarele valori nu este o valoare acceptabila pentru proprietatea CSS `position`?

- a) `static`
- b) `relative`
- c) `fixed`
- d) `absolute`
- e) `float`

Which is the HTML element used to draw a raster graphic on a web page? => **canvas**

Which of the following tags does include the "stud.jpg" image, in an HTML document? =>

****

Which of the following CSS selectors refers an element with "my" id, in an HTML document? => **#my**

Which colour model does use a double-cone representation for its colour space? => **HLS**

Which is the output of the following chunk of HTML & CSS code?

( ceva cu span)

```
What is the output of the following chunk of HTML & CSS code?  
<div id="test">  
  <span>Text</span>  
</div>  
<style>  
div#test span { color: green; }  
div span { color: blue; }  
span { color: red; }  
</style>
```

⇒ **It colours the text in green.**

Check all arguments accepted by the DrawImage JavaScript function:

⇒ **a canvas element, an image, a video element**

Check all the processing operations involved in JPEG compression algorithm:

⇒ **transformare cosinus discreta, cuantizare, block splitting**

Select ALL statements depicting DOM, from the following:

- ⇒ DOM is the acronym for Document Object Model
- ⇒ DOM provides the way to programmatically access HTML structure in JavaScript
- ⇒ HTML of every web page is turned into a DOM representation by the browser

Select all characteristics of a raster image, from the following

⇒ **The raster image is figured as a matrix of points.**

⇒ Raster image file size depends on the image size and the colour depth

Select ALL solutions for incorporating CSS in HTML

- ⇒ ?defined within a style block, in the head section of a web page
- ⇒ (tag link, tag style (in head), atribut style pt inline (in body) )
- ⇒ c) este ambiguu -> Coded in the body of the web page

Select ALL correct statements referring to the FlexBox model

Select ALL correct statements referring to the FlexBox model.

Alegeți una sau mai multe opțiuni:

- ☐ a. It uses grid-column and grid-row properties to change the size of items.
- ☐ b. It is activated using display property
- ☐ c. It is used to change the horizontal or vertical layout of HTML elements, in a web page
- ☐ d. It can be used to programmatically change the order of the HTML elements.
- ☐ e. It is appropriated for creating a responsive web application

b, d, e, c? e ambiguu

Select ALL primary colours of the subtractive colour model.

cyan, magenta, yellow

What purpose has setTimeout JavaScript function?

Răspuns:

metoda setTimeout apeleaza o functie

sau evalueaza o expresie dupa un numar specificat de milisecunde. 1000ms=1s si functia ce va fi executata va fi rulata o singura data. Pt a repeta executia functiei putem folosi metoda setInterval. Pt a preveni functia din a rula putem folosi metoda clearTimeout

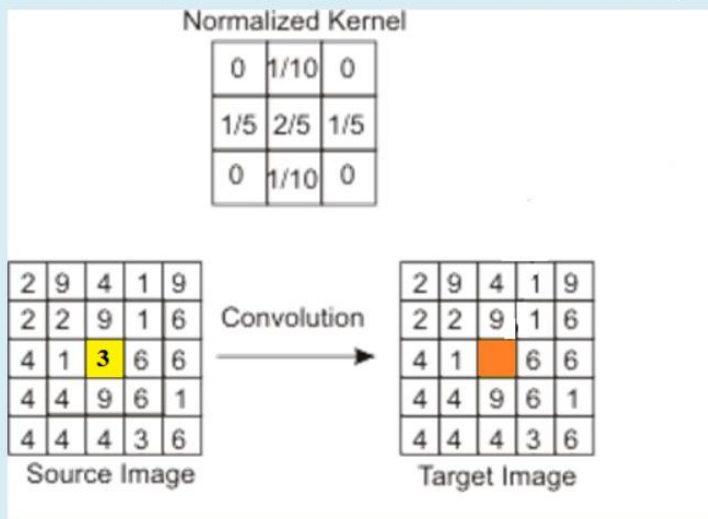
setTimeout(function(){ alert("Hello"); }, 3000); - afiseaza o fereastră de alerta cu textul hello dupa 3 sec

What is HyperVideo?

Răspuns:



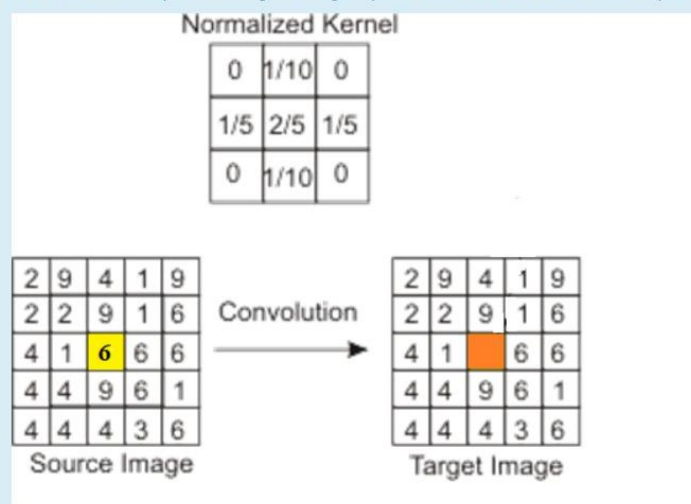
Care este valoarea lipsă din imaginea target aplicând kernelul normalizat centrat pe elementul de culoare galbenă din imaginea sursă?



Alegeți o opțiune:

- ☐ a. 4.4
- ☐ b.  $\frac{2}{5}$
- ☐ c. 3.4
- ☐ d. 4

Care este valoarea lipsă din imaginea target aplicând kernelul normalizat centrat pe elementul de culoare galbenă din imaginea sursă?



Alegeți o opțiune:

- ☐ a. 3.4
- ☐ b. 4.4
- ☐ c.  $\frac{4}{5}$
- ☐ d. 5.6

# TEORIE

## 1. Sunetul - formate de stocare

- formate fara compresie: WAV, AIFF
- formate de compresie fara pierderi: APE, MPEG4, FLAC
- formate de compresie cu pierderi: MP3, AAC, WMA, OGG

Astept continuare...

## 2. Tehnici de animatie

- Tehnica filmului
- Tehnica cadrelor cheie
- Tehnica schimbarii de culoare
- Tehnica schimbarii de forma

Astept continuare...

## 3. Grafica raster (BITMAP) - reprezentare si caracteristici

Reprezentata ca o matrice de puncte

Fiecare punct(pixel) retine informatii despre culoarea sa

Culorile sunt stocate conform modelului de culoare al imaginii

Caracteristici:

1. Rezolutia : reprezinta nr de linii si coloane ale matricii de reprezentare sau numarul total de pixeli
2. Adancimea de culoare : cat de multa informatie este retinuta despre culoare

Utilizare:

- Reprezentarea imaginilor pe monitor
- Reprezentarea capturilor externe

Avantaje:

- Poate reprezenta orice fel de imagine

Dezavantaje:

- Nu este scalabila
- Informatia este saracacioasa, nu tine cont de semantica
- Dimensiune mare

Formate: BMP, JPEG, GIF, TIFF

## TEHNICI DE ANIMATIE | GRAFICA RASTER (BITMAP)

### SUNET – FORMATE DE STOCARE – COMPRESIE / FARA COMPRESIE

### GRAFICA RASTER (BITMAP) - Reprezentata ca o matrice de puncte

Fiecare punct (pixel) retine informatii despre culoarea sa

Culorile sunt stocate conform modelului de culoare al imaginii

Caracteristici:

1. Rezolutia : reprezinta nr de linii si coloane ale matricii de reprezentare sau numarul total de pixeli
2. Adancimea de culoare : cat de multa informatie este retinuta despre culoare

Utilizare:

-Reprezentarea imaginilor pe monitor

-Reprezentarea capturilor externe

A/D

+ poate reprezenta orice fel de imagine

- nu este scalabila

- informatia este saracacioasa, nu tine cont de semantica

- dimensiune mare

Formate: BMP / JPEG / GIF / TIFF

SUNETUL

TEHNICI DE ANIMATIE

## **MODELE DE CULOARE**

### **Model de culoare**

Model matematic care descrie modalitatea de reprezentare a culorilor sub formă de tupluri numerice

Exemple: RGB, CMY

### **Spațiu de culori**

Modelul de culoare împreună cu instrucțiunile de reprezentare fizică, Exemple: sRGB, AdobeRGB, Pantone

### **Caracteristici**

Țin seama de modalitatea de percepere a luminii de către ochiul uman, Culori de bază albastru (S), verde (M) și roșu (L)

## **MODELUL ADITIV**

Bazat pe culorile de bază roșu, verde și albastru, Culoarea variază în funcție de dispozitiv, (în lipsa unui spațiu de culoare)

### **Spatii de culoare RGB**

sRGB - Dezvoltat de HP și Microsoft, Standard pentru monitoare / imprimante / web, Utilizat ca standard implicit

AdobeRGB - Dezvoltat de Adobe, Acoperă aproape complet spațiul de culori CMYK

## **MODELUL HSL**

Reprezentare sub formă de coordonate cilindrice, Bazat pe aceleași culori de bază: Red – 0grade, Green – 120grade, Blue – 240grade, Axa centrală – tonuri de gri, Utilizat în special pentru selecție de culoare

### **Modelul substractiv**

Modelul CMY(K) - Culori utilizate: cyan, magenta, yellow, Maschează culorile pe o suprafață albă, Utilizare: materiale tipărite

## **CSS – SPECIFICATOR CULORI**

**Format hexazecimal:** #rgb sau #rrggbb , r,g,b sunt cifre în baza 16

**Format RGB:** rgb(red, green, blue) sau rgba(red, green, blue, alpha), red, green, blue – numere de la 0 la 255 sau procente, alpha – număr între 0 – transparent și 1 – opac sau procent

**Format HSL:** hsl(hue, saturation, lightness) sau hsla(hue, saturation, lightness, alpha), hue – număr de la 0 la 360, saturation, lightness – procente, alpha – număr între 0 – transparent și 1 – opac sau procent

## PALETE DE CULORI

**Tabel de corespondență:** index – tuplu (R, G, B), Utilizare: Reducerea cantității de informație necesară pentru reprezentarea culorilor, Pretabilă pentru imagini cu un număr redus de culori (exemplu: diagrame fără gradienti)

## GRAFICA RASTER

Grafica raster (bitmap, matriceală): Reprezentare sub formă de matrice de puncte, Fiecare punct (denumit pixel) stochează informația de culoare, Culorile sunt tocate conform unui model de culoare, direct sau prin intermediul unei palete de culori

Caracteristici: Rezoluție (numărul de linii și coloane stocate în matrice, numărul total de pixeli sau densitate), Adâncime de culoare (cantitatea de informație stocată de către fiecare pixel)

Utilizare: Reprezentare imagine pe monitor, Captare imagini din surse externe

Avantaje și dezavantaje: Poate reprezenta orice imagine, Codaj sărac în informație (nu ia în considerare semantica imaginii), Dimensiune mare, Nu se pot adapta unei scări variabile de vizualizare

## FORMATE DE STOCARE

**BMP** (Microsoft Windows Bitmap): Formatul standard de stocare pe platforma Microsoft Windows, Suportă date necomprimate sau comprimate folosind algoritmul RLE, Monocromă sau în culori pe 4, 8, 16, 24 sau 32 de biți, Suportă palete de culori

**JPEG** (Joint Photographics Experts Group): Stocare comprimată cu pierdere de informație conform standardului JPEG, Rate de compresie diferite selectabile de către utilizator, Utilizat pentru imagini fotografice (cu gradații fine de culoare),

Nu este potrivit pentru text, linii sau alte imagini care prezintă un contrast foarte mare, Editări multiple (se pierde calitate la fiecare etapă de compresie / decompresie)

**GIF** (Graphics Interchange Format): Folosit în special pentru transferul imaginilor de maxim maxim 64K x 64K, Pretabil pentru diagrame, text logo-uri (contrast puternic și număr limitat de culori), Suportă maxim 256 culori prin intermediul unei palete de culori, Poate stoca mai multe cadre (pentru animație), Algoritm de compresie fără pierdere de informație Lempel-Ziv-Welch (LZW)

**TIFF** (Tag Image File Format): Format portabil și extensibil utilizat în special pentru imagini scanate, Suportă stocarea mai multor imagini într-un singur fișier

Suportă mai mulți algoritmi de compresie (RLE, LZW sau JPEG)

**Compresia RLE:** (nr aparitii, valoare), Rată mică de compresie, Se pretează pentru imagini cu zone mari de aceeași culoare, Utilizat în special pentru fișiere BMP cu paletă de culori

**Compresia LRW:** Algoritm de compresie universal bazat pe dicționar,

Descriere compresie: Se construiește dicționarul inițial (toate șirurile de lungime 1), Se caută cel mai lung șir W din dicționar care se potrivește cu șirul de la intrare

Se elimină W din șirul de intrare, Se adaugă W urmat de următorul caracter în dicționar, Se continuă cu pasul 2

Decompresie: se parcurge șirul codificat și se reconstruiește dinamic dicționarul

Utilizat pentru fișiere de tip GIF

**Compresia Huffman:** Algoritm universal de compresie, Codificare optimă de lungime variabilă pentru fiecare simbol în funcție de frecvența de apariție, Datele salvate: dicționarul, datele originale recodificate; Decodificare: translație simbol cu simbol pe baza dicționarului salvat.

**Compresia JPEG:** Compresie specializată cu pierdere de informație, Rezultate foarte bune pentru fotografii (variații fine de luminozitate și culoare)

Tipuri de compresie: **secvențial** – codaj bazat pe transformarea cosinus discretă cu blocurile procesate în ordinea apariției; **progresiv** – codaj bazat pe transformarea cosinus discretă cu blocurile procesate prin mai multe treceri asupra imaginii; **progresiv fără pierdere** – folosește doar algoritmi de compresie fără pierdere de informație, **progresiv ierarhic** – codifică imaginea la rezoluții din ce în ce mai mari

Etape: 1. Translatarea modului de culoare din RGB în Y'CBCR, 2. Reducerea rezoluției pentru componentele CB și CR, 3. Imaginea se descompune în blocuri de dimensiune 8x8 pixeli, 4. Se aplică transformata cosinus discretă pe fiecare bloc în parte, 5. Aplicarea matricei de cuantizare (pierdere de informație), 6. Blocurile rezultate în urma cuantizării sunt comprimate folosind RLE și Huffman; Decodificare: se aplică pașii în ordine inversă

## GRAFICA VECTORIALA

Bazătă pe descrierea matematică a obiectelor componente ale imaginii

Avantaje: Menține semantica – editare la nivel de obiect graphic, Dimensiune redusă, Independente de scara de vizualizare

Dezavantaj: Nu poate reprezenta fidel orice fel de informație

**Formate de stocare:** SVG (Scalable Vector Graphics) - Format generic bazat pe XML pentru reprezentări vectoriale 2D, Suportă animație și interactivitate

**DXF (Drawing Exchange Format)**-formatul vectorial lansat de firma Autodesk pentru produsul software AutoCAD

**EPS (Encapsulated Post Script)** - formatul firmei Adobe pentru imagini vectoriale, se bazează pe un limbaj de descriere numit Post Script

**SHP (Shapefile)** - formatul firmei ESRI pentru descrierea datelor spațiale de tip: punct, polilinie și polygon, utilizat la reprezentarea elementelor geografice în sisteme de tip GIS

## **ANIMATIE**

Modificarea rapidă a imaginii vizualizate prin modificarea poziției, formei sau dimensiunii unui obiect din imagine; Stocarea numerică a animației presupune reținerea elementelor independente ce compun mișcarea în raport cu factorul timp. Crearea iluziei de mișcare se realizează prin afișarea rapidă de imagini statice ușor modificate

Tehnici principale: Tehnica filmului, Cadre cheie, Schimbarea culorii

## **SUNETUL**

Reprezentare: Axa X: timp, Axa Y: presiune (0 – presiunea aerului în repaus), Amplitudine: măsoară dimensiunea vibrației / volumul sunetului, Frecvență: măsoară viteza vibrației / tonul sunetului

**Numerizarea sunetului:** Presupune stocarea și prelucrarea sunetului în format digital; Etape: Convertirea sunetului în semnal electric, Eșantionarea și cuantificarea semnalului, Stocarea informației numerice pe un suport de memorie externă conform unui format

Avantaje: Stocare mai ușoară, Permite analiza și procesarea numerică a sunetului, nu se degradează în timp sau la copieri repetate

**Esantionare:** Prin eşantionare se înțelege procesul de segmentare, cu o perioadicitate fixă, a semnalului audio analog. Frecvența de eşantionare – rezoluția orizontală. Se determină pe baza teoremei lui Nyquist (minim dublul frecvenței maxime a sunetului) Rate de eşantionare uzuale: 8 kHz – semnal telefonic, 11 kHz – radio AM, 22 kHz – radio FM, 44 kHz – audio CD

**Cuantificare:** Pp asocierea unei valori numerice corespunzătoare amplitudinii semnalului pentru fiecare interval de timp. Calitatea este influențată de numărul de biți alocați pentru fiecare eşantion (uzual 8 sau 16 biți pentru stocare și 16 – 32 pentru procesare). Redarea sunetului digital: Se reconstruiește sinusoida originală prin interpolarea valorilor numerice stocate, Prin intermediul unui convertor digital

**Formate audio:** **WAVE** – formatul standard de fișier audio pentru Microsoft și IBM; conține sunet în reprezentare PCM necomprimat; **AIFF** (Audio Interchange File Format) – formatul standard pentru audio digital utilizat pe platformele Apple (varianțe: necomprimat / comprimat); **MPEG** (Moving Picture Experts

Group) Audio - format standard pentru sunetul digital comprimat; parte a standardului MPEG de codificare a semnalului audio-video; cea mai cunoscută variantă a lui este MP3.

**Compresia:** Cel mai utilizat algoritm de compresie: MPEG-1 sau 2 Audio Layer III (MP3), Folosește codificare perceptuală - Elimină din rezultat sunetele care nu pot fi percepute de către urechea umană, Sunetele imperceptibile sunt eliminate pe baza unui model psihoacustic care exploatează fenomenele de: Mascare a frecvențelor, Mascare temporală

Mascarea frecvențelor: Sunt eliminate sunetele cu frecvența mai mare de 16-18 KHz, Sunt eliminate sunetele de intensitate scăzută, care apar concomitent cu sunete de intensitate înaltă, dacă sunt în benzi de frecvență alăturate (cele cu intensitate scăzută sunt mascate de cele cu intensitate înaltă)

Mascarea temporală: Se elimină sunetele de intensitate mică care urmează după sunete de intensitate puternică în cadrul unui interval de timp, Sunetele de intensitate mică nu pot fi percepute după sunete de intensitate puternică datorită inerției timpanului

**Compresia MP3**- etape: 1. Utilizarea de filtre pentru separarea sunetului în 32 sub-benzi de frecvență, 2. Aplicarea modelului psiho-acustic pentru eliminarea sunetelor imperceptibile, 3. Determinarea numărului de biți pentru coeficienți, 4. Prelucrarea valorilor obținute și compunerea fluxului final de biți

## VIDEO

Video digital – cuprinde totalitatea tehnicilor de captură, procesare și stocare a imaginilor în mișcare (precum și a sunetului asociat) prin intermediul unui dispozitiv de calcul.

Avantaje video digital: Poate fi procesat prin intermediul calculatorului, Păstrare în timp și rezistență la copieri repetate, Poate fi transmis la distanță

**Caracteristici:** Rezoluția, Spațiul de culoare și numărul de biți per pixel, Numărul de cadre pe secundă, Modul de afișare (întrețesut sau progresiv), Calitatea compresiei

**Formate:** Container – specifică structura de stocare a componentelor video (imagini + audio) și a datelor asociate (metadate, subtitrări, ...)

**Advanced Systems Format – ASF:** container dezvoltat de Microsoft care poate conține fluxuri codate cu orice codec (Extensii: .asf, .wma, .wmv), **Audio Video Interleave – AVI:** container mai vechi dezvoltat de Microsoft pe baza Resource **Interchange File Format – RIFF** (stochează datele în secțiuni identificate prin markere FourCC), **MP4 – MPEG-4 Part 14:** dezvoltat de către Motion Pictures Expert Group și utilizat inițial de către QuickTime (video H.264, audio AAC),

**AVCHD** – format utilizat în special de către camerele video (video H.264 AVC și sunet AC3 sau PCM), **Matroska / OGG:** formate deschise; pot conține mai multe fluxuri audio / video

Codec – specifică modalitatea de compresie / decompresie pentru un flux video / audio în cadrul unui container



**H.264 / MPEG-4 AVC** – cel mai popular (utilizat pentru Web, BluRay, camere video), **H.262 / MPEG-2** – formatul standard pentru DVD, **Windows Media Video** – format dezvoltat de către Microsoft, **MJPEG (Motion JPEG)** – format mai vechi bazat pe compresia JPEG

**Compresia:** Se bazează pe reducerea redundanței din cadrul fluxului video

Redundanța spațială (intra-cadru) - tipul de redundanță identificat și eliminat de algoritmi de compresie a imaginilor, Redundanța temporală (inter-cadru) - redundanță identificată între două cadre consecutive (de exemplu, prin compararea a două cadre se observă că majoritatea pixelilor își păstrează valoarea)

### **Algoritm de compresie video**

**Hibrid** - Transformata Cosinus Discretă – similar JPEG pentru reducerea redundanței spațiale, Codaj Huffman – pentru comprimarea coeficienților TCD, Codificarea mișcării – pentru reducerea redundanței temporale, Codaj **RLEAsimetric** - Timpul de codare este mult mai mare decât cel de decodare

**Etape compresie:** Împărțirea imaginii în blocuri: 16x16 luminanță, 8x8 crominanță (culoare); Compresie pe baza DCT pentru reducere spațială, Aplicarea tehnicilor de compensare a mișcării pentru reducere temporală, Faza finală de codare pe două dimensiuni folosind Run Length Encoding

**Tipuri de cadre:** <I> Intra-picture/frame/image - Cadrele cheie, Necesare pentru căutare și poziționare, Compresie moderată

<P> Predicted pictures - Codate cu referință la un cadru anterior, Folosite ca referință pentru cadre ulterioare

<B> Bi-directional prediction (interpolated pictures) - Necesită cadre anterioare și viitoare pentru refacere, Compresie mare

# NOTITE SEMINAR MULTIMEDIA

## SEMINAR 2:

### 1. Adaugare fisier css extern:

```
<link rel="stylesheet" type="text/css" href="agenda.css">
```

### 2. Adaugare fisier javascript:

```
<script type="text/javascript"> ->daca scriut direct aici
```

```
<script src="cod.js" type="text/javascript"></script> ->daca iau din alt fisier
```

### 3. Tabel:

```
<table>
<caption>Person List </caption>
<thead> //header
<tr>
<th>Last Name</th> //celula din header
<th>First Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody> //body
<tr>
<td>Popescu</td> //celula din body
<td>Ion</td>
<td>0732555</td>
</tr>
</tbody>
<tfoot> //footer
<tr>
<td colspan="3">Number of persons:1 </td> //ocupa 3 coloane
</tr>
</tfoot>
</table>
```

#### 4. Formular:

```
<form action="#">
<label for="lastName">Last name:</label>
<input type="text" id="lastName" name="lastName" placeholder="last name">
<label for="phone">Phone:</label>
<input type="tel" id="phone" name="phone" placeholder="phone">

<input type="button" value="Add person">
</form>
```

#### 5. CSS:

Selector de tip element:  
Body, h1, table, p, etc...  
Color->culoarea textului  
Background-color->culoarea fundalului  
Font-family:tahoma->fontul  
Font-size:11pt; ->marime font  
Font-weight:bold ->stil font  
Text-align:left ->orientare in pagina  
Border: 1px solid black  
Width: 60%  
Border-radius:5ps; ->colturi rotunjite  
input[type="button"] ->selector

### SEMINAR 3:

#### 1. Validare de null:

```
if (lastName.value === '' || lastName.value == null)
```

#### 2. Validare doar cu cifre:

```
if (!/^[0-9]+$/.test(phone.value))
```

#### 3. Adaugare linii in tabel folosind javascript:

```
//adaugam randuri noi in tabel cu informatia citita
```

```
let tr = document.createElement("tr"); //l. fac randul dorit
```

```
let tdLastName = document.createElement("td"); //1. se face coloana
tdLastName.innerText = lastName; //2. Se pune text in coloana noua
tr.appendChild(tdLastName); //3. Se lipeste coloana la randul creat
```

```
let tdDelete = document.createElement("td")
let Button = document.createElement("input")
Button.type="button"
Button.value="Delete"
```

```
Button.addEventListener("click",deletePerson2);
tdDelete.appendChild(Button)
```

```
tr.appendChild(tdDelete);
```

```
let tBody = document.querySelector("tbody"); // II. selectam body ptr a lipii la el noul
rand creat
tBody.appendChild(tr); // III. Punem randul in body
```

#### 4. Sterge un rand din tabel:

```
function deletePerson2(){
    let input=this;
    let tdToDelete=input.parentNode;
    let trToDelete=tdToDelete.parentNode;
    let tBody = document.querySelector("tbody");
    tBody.removeChild(trToDelete);
    document.getElementById("noOfPersons").innerText
=document.getElementById("myTable").rows.length-2;
}
```

## SEMINAR 4: CANVAS-BAR CHART

### 1. Canvas-lucru initial:

```
//inainte de a lucra cu un canvas trebuie sa obtinem un context
let context = this.canvas.getContext("2d")

context.lineWidth = 2; //ptr grosimea liniilor
```

```

context.textAlign="center" //ptr alinierea in cadrul canvasului

context.strokeStyle="#dedede" //ptr a schimba culoarea contur
context.fillStyle="#dedede" //ptr a schimba culoarea de umplere
context.fillRect(0, 0, this.canvas.width, this.canvas.height) //ptr a umple un dreptunghi, x,y-
//coordonatele de la care desenam, w,h- latime si inaltime

```

## 2. Desenare grafic cu bare dupa un vector de valori:

```

draw(values){
let context = this.canvas.getContext("2d");

let barWidth = this.canvas.width / values.length;

let maxValue = Math.max(...values)
let f = this.canvas.height / maxValue;

for (let i = 0; i < values.length; i++) {

let barHeight = values[i] * f * 0.9;
let barX = i * barWidth;
let barY = this.canvas.height - barHeight;

let barWidthVisible=barWidth * 0.9

context.fillStyle = "#FF0000"
context.fillRect(barX, barY, barWidthVisible, barHeight)
context.strokeRect(barX, barY, barWidthVisible, barHeight)

context.fillStyle="#000000"
context.fillText(values[i],barX+barWidthVisible/2,barY-10)
}
}

```

## 3. Download imagine desenata pe canvas la click pe canvas:

HTML:

```
<a href="#" download="barChart">
```

```
<canvas id="canvas" style="width: 800px; height: 600px;">
```

Your browser does not support the canvas element!

```
</canvas>
</a>
```

JS:

```
canvas.addEventListener("click", function(){
    let a = this.parentNode; // this este elementul care genereaza click deci canvas si canvas
    este pus intr-un link a asadar this.parentNode este linkul a
    let dataUrl=this.toDataURL("image/png")
    console.log(dataUrl);
    a.href=dataUrl;
})
```

#### 4. Eveniment de click pe un element:

```
btnDrawChart.addEventListener("click", function () {...});
```

#### 5. Extragere valoare dintr-un camp:

```
let tbValues = document.getElementById('tbValue');
let values = tbValues.value;
```

#### 6. Transformare String de forma „[12,2,24]” intr-un vector cu elementele 12,2,24:

```
let valuesArray = eval('[ ' + values + ' ]');
barChart.draw(valuesArray)
```

### SEMINAR 5 – CANVAS – HISTOGRAMA DUPA POZA

#### 1. Desenare histograma dupa un set de valori date in vector:

```
draw(values){
    let context = this.canvas.getContext("2d");

    let maxValue = Math.max(...values)
    let f = this.canvas.height / maxValue;

    let barWidth = this.canvas.width / values.length;

    context.save();
    context.rotate(Math.PI); //rotatie la 180 de grade
```

```

context.translate(0,-this.canvas.height)//translatie pe verticala, in sus, cu valoarea inaltimii

context.scale(-1,f); //-1 adica face flipp pe verticala, oglinda

for(let i=0;i<values.length;i++)
{
    context.fillRect(i*barWidth,0,barWidth*0.9,values[i]);
}
context.clearRect(0,0,this.canvas.width,this.canvas.height)
context.restore();
}

```

2. `scale(-1, 1)` to flip the context horizontally and `scale(1, -1)` to flip it vertically. The `translate()` method adds a translation transformation to the current matrix by moving the canvas and its origin `x` units horizontally and `y` units vertically on the grid.

### 3. Drag and drop imagine pe canvas:

```

document.addEventListener("dragover", function (e) {
    e.preventDefault();
})
document.addEventListener("drop", function (e) {

    e.preventDefault();

    let data = e.dataTransfer;
    let files = data.files;

    let fileReader = new FileReader();
    fileReader.addEventListener("load", function (e) {

        let dataUrl = e.target.result;// e.target ==> FileReader //refera obiectul de tip
        filereader

        let img = document.createElement("img");
        img.addEventListener("load", function (e) {

            canvasImage.width = img.naturalWidth;
            canvasImage.height = img.naturalHeight;

            let context = canvasImage.getContext("2d");
            context.drawImage(img, 0, 0);

```

```

    })
    img.src = dataUrl;

    });
    FileReader.readAsDataURL(files[0]);
  }
})

```

#### 4. Extragere pixeli din imaginea de pe canvas – ptr histograma

```

let imageData = context.getImageData(0, 0, canvasImage.width, canvasImage.height)
let data = imageData.data;

for (let i = 0; i < data.length; i += 4) { //fiecare pixel are 4 pozitii ptr canalele rgb
  let r = data[i]; //rosu
  let g = data[i + 1]; //verde
  let b = data[i + 2]; //albastru
  let a = data[i + 3]; // transparenta, daca exista

  let average = Math.round((r + g + b) / 3);

  v[average]++; //de fiecare data cand gasim un pixel care are o tonalitate din
  vector vom incrementa acea tonalitate cu nr

}

```

### SEMINAR 6 – CANVAS – FILTRE PE IMAGINE

#### 1. Tab de „Adauga fisier” care deschide file explorer ptr incarcare imagine:

HTML:

```
<input id="fileBrowser" type="file" accept="image/*">
```

JS:

```

document.getElementById("fileBrowser").addEventListener("change", function (ev) {
  const files = ev.target.files;

  const reader = new FileReader();
  reader.addEventListener("load", function (ev) {

```



```

const dataURL = ev.target.result;

const img = document.createElement("img");
img.addEventListener("load", function (ev) {

    app.visibleCanvas.width = img.naturalWidth;
    app.visibleCanvas.height = img.naturalHeight;
    const oContext = app.offscreenCanvas.getContext("2d");
    oContext.drawImage(ev.target, 0, 0);
    })
    img.src = dataURL;
    })
    reader.readAsDataURL(files[0]);
});

```

## 2. Filtre:

- *Greyscale:*

```

for (let i = 0; i < data.length; i += 4) {
    const r = data[i];
    const g = data[i + 1];
    const b = data[i + 2];
    // const a=data[i+3];

    const avg = Math.round((r + g + b) / 3);
    data[i] = data[i + 1] = data[i + 2] = avg;
}

```
- *Threshold*

```

for (let i = 0; i < data.length; i += 4) {
const r = data[i];
    const g = data[i + 1];
    const b = data[i + 2];
    // const a=data[i+3];

    const v = (0.2126 * r + 0.7152 * g + 0.0722 * b >= threshold) ? 255 : 0;
    data[i] = data[i + 1] = data[i + 2] = v;
}

```
- *Sepia*

- ```

for (let i = 0; i < data.length; i += 4) {
    const r = data[i];
    const g = data[i + 1];
    const b = data[i + 2];
    data[i] = (r * .393) + (g * .769) + (b * .189);
    data[i + 1] = (r * .349) + (g * .686) + (b * .168);
    data[i + 2] = (r * .272) + (g * .534) + (b * .131);
}

```
- *Invert:*

```

for (let i = 0; i < data.length; i += 4) {
    const r = data[i];
    const g = data[i + 1];
    const b = data[i + 2];
    data[i] = 255 - r;
    data[i + 1] = 255 - g;
    data[i + 2] = 255 - b;
}

```
  - *Pixelate:*

```

const blocksize = 10;
const oContext = app.offscreenCanvas.getContext("2d");
const vContext = app.visibleCanvas.getContext("2d");

for (var x = 1; x < app.offscreenCanvas.width; x += blocksize) {
    for (var y = 1; y < app.offscreenCanvas.height; y += blocksize) {
        var pixel = oContext.getImageData(x, y, 1, 1);
        vContext.fillStyle = "rgb(" + pixel.data[0] + "," + pixel.data[1] + "," + pixel.data[2] +
        ")";
        vContext.fillRect(x, y, x + blocksize - 1, y + blocksize - 1);
    }
}

```
  - *2Channels:*

```

for (let i = 0; i < data.length; i += 4) {

    const g = data[i + 1];
    data[i + 2] = g;
}

```
  - *Red:*

```

for (let i = 0; i < data.length; i += 4) {

    data[i + 1] = 0;
    data[i + 2] = 0;
}

```

3. Cum sa selectezi mai multe butoane care au aceeasi proprietate:

HTML:

```
<button data-effect="normal">Normal</button>
<button data-effect="grayscale">Grayscale</button>
```

JS:

```
document.querySelectorAll("button[data-effect]");
```

## SEMINAR 7 – CANVAS – FILTRE PE IMAGINE 2

1. Sa se faca butoanele cu scris rosu cand sunt apasate:

CSS:

```
.active{
  color:red;
}
```

JS:

```
const previousButton = document.querySelector(".active");
if (previousButton != null) {
  previousButton.classList.remove("active");
}
button.classList.add("active");
```

2. Pozitionare div in mijlocul ecranului, sus, folosind css:

```
<div style="
  background-color:rgba(200,200,200,0.8);
  padding: 10px; position: fixed;top:10px; left:50%; transform: translate(-50%,0);">
```

3. Facem un buton invizibil cu css:

HTML: <button id="btnDownload" style="display: none;">Download</button>  
SAU JS: btn.style.display = "none";

4. Aplicare filtru doar pe jumatatea stanga a pozei-preluare pixeli in sectiuni, nu in totalitate

```
const oContext = offscreenCanvas.getContext("2d");
```

```
const imageData = oContext.getImageData(0, 0, offscreenCanvas.width,
offscreenCanvas.height);
const data = imageData.data;
```

```
for (let y = 0; y < offscreenCanvas.height; y++) {
  for (let x = 0; x < offscreenCanvas.width/2; x++) {
    const i = (y * (offscreenCanvas.width) * 4) + x * 4;

    const r = data[i];
    const g = data[i+1];
    const b = data[i+2];
    const transparency = data[i+3];
    const average = Math.round((r + g + b) / 3);

    data[i] = average;
    data[i+1] = average;
    data[i+2] = average;
  }
}
```

##### 5. Filtru Darker si Brighter cu o valoare data:

- *Darker:*

```
for (let i = 0; i < data.length; i += 4) {
  const r = data[i];
  const g = data[i + 1];
  const b = data[i + 2];
```

```
  data[i] = r - v;
  data[i + 1] = g - v;
  data[i + 2] = b - v;
}
```

- *Brighter:*

```
for (let i = 0; i < data.length; i += 4) {
  const r = data[i];
  const g = data[i + 1];
  const b = data[i + 2];
```

```
  data[i] = r + v;
  data[i + 1] = g + v;
  data[i + 2] = b + v;
}
```

##### 6. Buton pentru download imagine de pe canvas:

```
document.querySelector("#btnDownload").addEventListener("click", (ev) => {
    const a = document.createElement("a");
    a.href = app.visibleCanvas.toDataURL();
    a.download = "output.png";
    a.click();
})
```

## 7. Preluare culoare din punctul in care ne aflam cu mouse-ul:

```
app.visibleCanvas.addEventListener("mousemove", (ev) => {
    const x = ev.offsetX * app.visibleCanvas.width / app.visibleCanvas.clientWidth;
    const y = ev.offsetY * app.visibleCanvas.width / app.visibleCanvas.clientWidth;

    const vContext = app.visibleCanvas.getContext("2d");
    const imageData = vContext.getImageData(x, y, 1, 1);
    const data = imageData.data;
    const r = data[0];
    const g = data[1];
    const b = data[2];

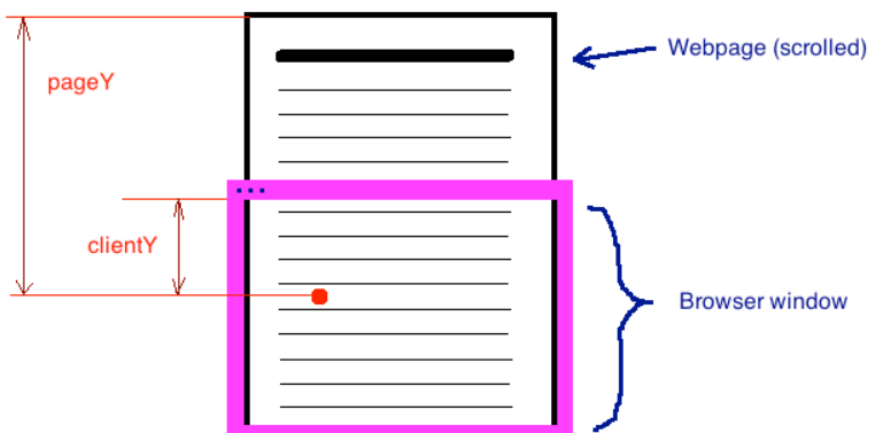
    const span = document.getElementById("color");
    const color = "rgb(" + r + ", " + g + ", " + b + ")"; //sau const color=`rgb(${r},${g},${b})`;

    span.innerText = color;
    span.style.backgroundColor = color;
})
```

## 8. Pozitie in pagina:

**pageX**: pozitia in pagina relativa la inceputul intregii pagini

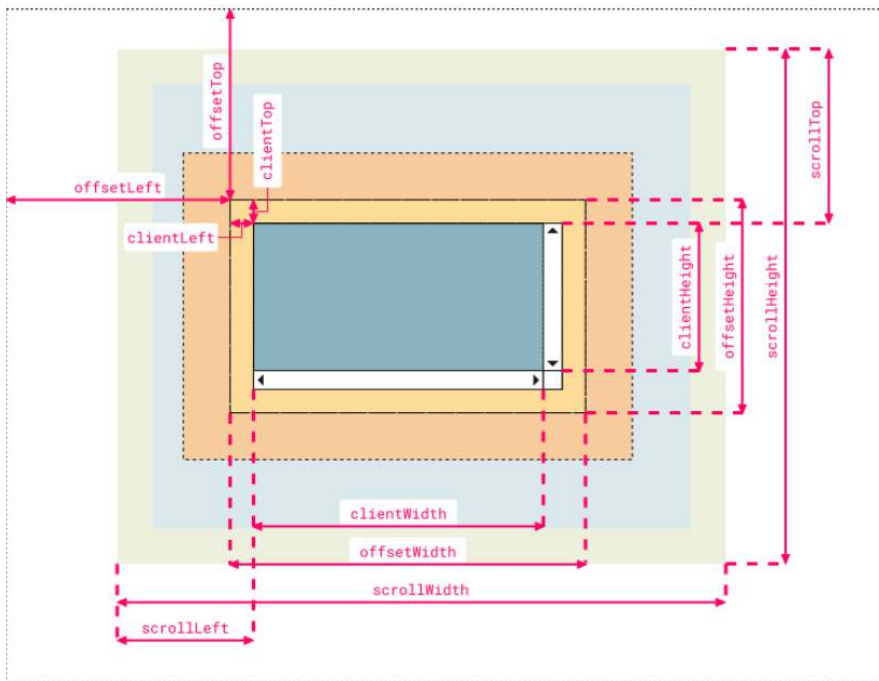
**clientX**: pozitia in pagina relativa la ce vede utilizatorul(viewport), chiar daca pagina este scrollata si incepe de mai sus.



clientLeft: 5  
clientTop: 5  
offsetLeft: 220  
offsetTop: 70

clientLeft and clientTop only show the width of the border. No margin. No padding. No left/top position. Just border-width.

offsetLeft and offsetTop are the left/top specified values + the margin. No border. No padding.



**offsetTop/Left:** distanta fata de sus sau stanga relativa la punctul 0.0 din care incepe toata pagina, indiferent daca avem scroll

**getBoundingClientRect().top/left:** distanta fata de sus sau stanga relativa la fereastra pe care o vede utilizatorul(viewport), chiar daca pagina este scrollata si incepe de mai sus

Daca vrem sa luam pozitia mouse-ului in canvas la eventul de mousedown putem scrie:

→ `pozitie.x = event.pageX - canvas.offsetLeft;`

`pozitie.y = event.pageY - canvas.offsetTop;`

→ `pozitie.x = event.clientX - canvas.getBoundingClientRect().left;`

`pozitie.y = event.clientY - canvas.getBoundingClientRect().top;`

## SEMINAR 8 – SVG – BARCHART

### 1. Schimbam stilul cand suntem deasupra unui element; util pentru butoane:

CSS:

```
.button:hover{  
    fill:yellow;  
}
```

### 2. Pentru a lucra cu svg avem nevoie de un namespace:

```
this.svgns = "http://www.w3.org/2000/svg";  
this.svg = document.createElementNS(this.svgns, "svg");
```

...PROCESARE SVG...

```
this.element.appendChild(this.svg); //element este un div in html
```

### 3. Modificari CSS in JS:

```
→this.svg.setAttribute("style", "border: 1px solid black");
```

SAU

```
→ this.svg.style.borderColor = "black";  
   this.svg.style.borderWidth = "1px";  
   this.svg.style.borderStyle = "solid";
```

### 4. Desenare rect in SVG:

```
const rect = document.createElementNS(this.svgns, "rect");  
rect.setAttribute("x", 0);  
rect.setAttribute("y", 0);  
rect.setAttribute("width", this.width);  
rect.setAttribute("height", this.height);  
rect.style.fill = "whitesmoke";  
  
this.svg.appendChild(rect);
```

### 5. Desenare Barchart pe un SVG dupa un vector cu valori:

```

const data=[
    ['Label 1',1],
    ['Label 2',2],
    ['Label 3',3]
]

....
const barWidth = this.width / this.data.length;
const maxValue = Math.max(...this.data.map((x) => x[1])); //map pt a obtine doar valorile
numerice
const f = this.height / maxValue;

for (let i = 0; i < this.data.length; i++) {
    const bar = document.createElementNS(this.svgns, "rect"); //desenare rect
    const text=document.createElementNS(this.svgns,"text");    //desenare text

    const label = this.data[i][0];
    const value = this.data[i][1];

    const barHeight = value * f * 0.9;

    const barX = i * barWidth;
    const barY = this.height - barHeight; //ca sa nu le afiseze de sus

    bar.classList.add("bar"); //ptr prelucrarea in css cu selectorul .bar

    bar.setAttribute("x", barX + barWidth / 4);
    bar.setAttribute("y", barY);
    bar.setAttribute("width", barWidth / 2);
    bar.setAttribute("height", barHeight);

    bar.addEventListener("click",()=>{
        alert(value);
    });

    text.setAttribute("x",barX+barWidth/4);
    text.setAttribute("y",barY-10);
    const labelValue=document.createTextNode(label);
    text.appendChild(labelValue);
}

```



```
// bar.style.fill = "red"; -> aici daca vrem sa facem aia cu hover pe galben are  
specificitatea prea mare si raman rosii  
//daca am pune cu bar.setAttribute("fill","red"); //ar fi functionat ptr ca nu mai are  
specificitate mai mare decat hover
```

```
bar.style.stroke = "black";  
//bar.setAttribute("stroke-width", "10px");  
bar.style.strokeWidth = "2px";  
//sau bar.style["stroke-width"]="2px"  
  
this.svg.appendChild(bar);  
this.svg.appendChild(text);  
  
}
```

## SEMINAR 9 – AUDIO.1

### 1. Adaugare librarii externe-Bootstrap si fonturi Cloudflare

```
<!-- Required meta tags -->  
<meta charset="utf-8">  
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
  
<!-- Bootstrap CSS -->  
<link rel="stylesheet"  
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"  
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5ulKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"  
crossorigin="anonymous">  
  
<!-- Fonturi-->  
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/5.15.1/css/all.min.css">
```

### 2. Element de tip audio in HTML:

```

<audio id="audio" src="media/Beethoven-MoonlightSonata.mp3">
  Your browser does not support the "audio" element
</audio>
SAU
<audio
  controls //ne face automat butoane de control
  src="/media/cc0-audio/t-rex-roar.mp3">
  Your browser does not support the audio element.
</audio>

```

### 3. Buton de play/pause pentru muzica din elementul audio:

```

app.btnPlayPause.addEventListener("click", () => {
  if (app.audio.paused)
    app.audio.play();
  else
    app.audio.pause();
});

```

### 4. Cum sa afisam durata totala a melodiei si pozitia curenta in melodie:

→ *functie de formatare float in min si sec ptr afisare corecta:*

```

function formatTime(seconds) {

  let minutes = Math.floor(seconds / 60);
  let secondsRemaining = Math.floor(seconds - minutes * 60);

  let result = "" + (minutes < 10 ? "0" : "") + minutes + ":" + (secondsRemaining < 10 ? "0" : "") +
  secondsRemaining;

  return result;
}

```

→ *eventimentele care trebuie tratate:*

```

app.audio.addEventListener("durationchange", () => { //cand se schimba durata totala

  app.spDuration.innerHTML = formatTime(app.audio.duration); // audio.duration ne da
  lungimea totala a melodiei

});

```

```

app.audio.addEventListener("timeupdate", () => {//cand se schimba valoarea din currentTime

    app.spCurrentTime.innerHTML = formatTime(app.audio.currentTime);//audio.currentTime
ne da pozitia curenta in melodie

});

```

## SEMINAR 10 – AUDIO.2->playlist

### 1. Playlist-ul poate fi o simpla lista in care pastram sursa ptr fiecare melodie cu un atribut:

```

<ul id="playlist" class="list-group">
  <li class="list-group-item"
    data-url="media/Bolero.mp3">
    Ravel - Bolero
  </li>
  <li class="list-group-item"
    data-url="media/Beethoven-MoonlightSonata.mp3">
    Beethoven - Moonlight Sonata
  </li>
  <li class="list-group-item"
    data-url="media/CanoninD.mp3">
    Pachelbel - Canon in D
  </li>
</ul>

```

### 2. Cum selectam o melodie din playlist ptr a da play la click pe ea:

→ `const elements = document.querySelectorAll("li[data-url]");`*//selectam toate randurile din lista, adica toate melodiile si ptr fiecare am salvat in atributul data-url link-ul sursa*

```

for (let i = 0; i < elements.length; i++) {

    const element = elements[i]
    const url = element.dataset.url;

    element.addEventListener("click", function () {
        app.play(url);

    })
}

```

→app.play = function (url) {*//asta e functia de play din aplicatia scrisa de noi*

```
const previouslySelectedElement = document.querySelector("li.active");  
if (previouslySelectedElement !== null)  
    previouslySelectedElement.classList.remove("active");
```

```
const selector = 'li[data-url="' + url + '"]';  
const selectedElement = document.querySelector(selector);
```

```
selectedElement.classList.add("active");
```

```
app.audio.src = url;  
app.audio.play();  
}
```

→app.audio.addEventListener("play", () => {*//asta e functia de play a elementului audio*  
 const selectedElement=document.querySelector("li[data-url='"+app.audio.getAttribute('src')+"']");  
 selectedElement.classList.add("active");  
});

### 3. Buton de +10 secunde in melodie:

```
app.btnForward.addEventListener("click", () => {  
    app.audio.currentTime += 10;  
});
```

## SEMINAR 11 – AUDIO.3 -> MICROFON SI SOUND BAR ANALYZAR-Frequency Bars

### 1. Dropdown de tip select in HTML:

HTML:

```
<label for="visualisation"> Visualizer settings</label>  
<select id="visualisation" >  
    <option disabled> Choose</option>  
    <option value="sinewave">Sinewave</option>  
    <option value="bars">Frequency Bars</option>
```

```
</select>
```

JS:

```
display(visualisation) {  
    this.audioContext.resume();  
  
    if (visualisation === "bars") {  
        this.displayBars();  
    } else if (visualisation === "sinewave") {  
        this.drawSineWave();  
    }  
}
```

2. The `MediaDevices.getUserMedia()` method prompts the user for permission to use a media input which produces a `MediaStream` with tracks containing the requested types of media. That stream can include, for example, a video or an audio.

```
stream = await navigator.mediaDevices.getUserMedia;
```

3. **Cum sa preluam audio-ul din microfon:**

```
→ navigator.mediaDevices.getUserMedia({  
    audio:true//daca exista un input de tip audio  
}).then(function(stream){  
    soundAnalyzer.setStreamSource(stream)//atunci stream-ul creat de input se trimite  
    mai departe in aceasta metoda care construiesc un nod sursa?  
})  
.catch(function(err){  
    alert(err.message);  
})
```

→ Undeva in constructor :

```
this.audioContext = new AudioContext();  
this.analyserNode = this.audioContext.createAnalyser();
```

```
→ setStreamSource(stream){  
    const sourceNode = this.audioContext.createMediaStreamSource(stream);  
    sourceNode.connect(this.analyserNode);  
}
```

!!! daca vrem, putem transmite aici un element audio care are o melodie in el, nu neaparat un stream de la microfon DAR PUNEM `MediaElementSource`

```
setMediaElementSource(mediaElement) {
  this.sourceNode = this.audioContext.createMediaElementSource(mediaElement);
  this.sourceNode.connect(this.analyserNode);
}
```

[https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Audio\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API)

### AudioContext

The **AudioContext** interface represents an audio-processing graph built from audio modules linked together, each represented by an **AudioNode**. You need to create an **AudioContext** before you do anything else, as everything happens inside a context.

### AnalyserNode

The **AnalyserNode** interface represents a node able to provide real-time frequency and time-domain analysis information, for the purposes of data analysis and visualization. **->audioContext.createAnalyser()**

### MediaStreamAudioSourceNode

The **MediaStreamAudioSourceNode** interface represents an audio source consisting of a **MediaStream** (such as a webcam, microphone, or a stream being sent from a remote computer). It is an **AudioNode** that acts as an audio source. **->audioContext.createMediaStreamSource(stream)**

## 4. Construirea graficului cu bare ptr analiza audio: sound bar analyser-Frequency Bars

```
drawBars(){
  this.context.fillStyle = "black";
  this.context.fillRect(0, 0, this.canvas.width, this.canvas.height);

  this.analyserNode.fftSize = 256;

  const bufferLength = this.analyserNode.frequencyBinCount;
  const buffer = new Uint8Array(bufferLength);
  this.analyserNode.getByteFrequencyData(buffer);

  const barWidth = this.canvas.width / buffer.length;
  const f = this.canvas.height / 255;

  this.context.fillStyle = "red";
  for(let i=0; i< buffer.length; i++){
    const barHeight = buffer[i] * f;
    const barX = i*barWidth;
    const barY = this.canvas.height - barHeight;
```

```

        this.context.fillRect(barX, barY, barWidth, barHeight);
    }

    //setInterval
    requestAnimationFrame(=> this.drawBars());
}

```

The `window.requestAnimationFrame()` method tells the browser that you wish to perform an animation and requests that the browser calls a specified function to update an animation before the next repaint. The method takes a callback as an argument to be invoked before the repaint.

## SEMINAR 12.1 – AUDIO.4 -> MICROFON SI SOUND WAVE ANALYZAR- Sinewave

### 1. Construirea graficului cu wave ptr analiza audio: sound waveanalyser-Sinewave

```

drawSineWave() {
    let bufferLength = this.analyserNode.fftSize;
    let dataArray = new Float32Array(bufferLength);
    this.analyserNode.getFloatTimeDomainData(dataArray);

    this.context.fillStyle = 'rgb(200, 200, 200)';
    this.context.fillRect(0, 0, this.canvas.width, this.canvas.height);
    this.context.lineWidth = 2;
    this.context.strokeStyle = 'rgb(0, 0, 0)';

    this.context.beginPath();

    let sliceWidth = this.canvas.width * 1.0 / bufferLength;
    let x = 0;
    for (let i = 0; i < bufferLength; i++) {

```

```

let v = dataArray[i] * 200.0;
let y = this.canvas.height / 2 + v;

if (i === 0) {
  this.context.moveTo(x, y);
} else {
  this.context.lineTo(x, y);
}
x += sliceWidth;
}
this.context.lineTo(this.canvas.width, this.canvas.height / 2);
this.context.stroke();

this.drawVisual = requestAnimationFrame(() => this.drawSineWave()); }

```

## SEMINAR 12.2 – VIDEO

### 1. Element de tip video:

```

<video
  id="video"
  src="media/tears-of-steel-battle-clip-medium.mp4"
  muted
  controls>
  Not supported
</video>

```

!!!Daca punem video ca atare, va fi vizualizat din prima. Daca il punem intr-un canvas trebuie sa facem `context.drawImage(video,0,0);`

### 2. Captura de ecran din video:

```

btnCapture.addEventListener("click", ()=>{
  canvasCapture.width = video.videoWidth;//!aparent nu merge cu
  video.clientWidth;
  canvasCapture.height = video.videoHeight;

  const context = canvasCapture.getContext("2d");
  context.drawImage(video,0,0);
});

```



### 3. Mut si unmute pe video:

```
btnMuted.addEventListener("click", ()=> {  
    if(video.muted === true){  
        video.muted = false;  
        btnMuted.innerHTML = "Mute";  
    }  
    else{  
        video.muted = true;  
        btnMuted.innerHTML = "UnMute";  
    }  
});
```

### 4. Pause si unpause pe video:

```
btnPlayPause.addEventListener("click", ()=> {  
    if(video.paused === true){  
        video.play();  
        btnPlayPause.innerHTML = "Pause";  
    }  
    else{  
        video.pause();  
        btnPlayPause.innerHTML = "Play";  
    }  
});
```

### 5. Canvas cat un video:

```
video.addEventListener("canplay", ()=>{  
    canvasVideo.width = video.videoWidth; // aparent nu merge cu video.clientWidth;  
    canvasVideo.height = video.videoHeight;  
});
```

### 6. Animatie pe video: filtru si secunde

```
function draw(){  
    contextVideo.drawImage(video, 0, 0);  
  
    const imageData = contextVideo.getImageData(0, 0,  
        canvasVideo.width, canvasVideo.height);  
    const data = imageData.data; //merge doar pe microsoft edge, nu si pe chrome
```

```

/*for(let i=0; i<data.length; i+=4){
  const average = (data[i] + data[i+1] + data[i+2])/3;
  data[i] = data[i+1] = data[i+2] = average;
},*/

```

→aplicare filtru

```

for (let y=0;y<canvasVideo.height;y++)
  for (let x=0;x<canvasVideo.width;x+=2)
  {
    let i= y*canvasVideo.width*4+x*4;

    const average=(data[i]+data[i+1]+data[i+2])/3;
    data[i]=data[i+1]=data[i+2];
  }

```

```

contextVideo.putImageData(imageData, 0, 0);

```

```

contextVideo.fillStyle = "white";
contextVideo.font = "100px sans-serif";
contextVideo.textAlign = "center"
contextVideo.fillText(

```

```

video.currentTime,
  canvasVideo.width/2, 100)

```

```

requestId = requestAnimationFrame(draw); //ptr a desena la fiecare secunda, nu doar
odata
}

```

7. Daca vrem ca animatia cu filtru pe video si scris sa se opreasca atunci cand punem pauza:

```

let requestId=null;
video.addEventListener("play", ()=>{
  requestId = requestAnimationFrame(draw);
})
video.addEventListener("pause", ()=>{
  cancelAnimationFrame(requestId);
})

```

8. Activare si dezactivare controls

```

const btnControls = document.getElementById("btnControls");
btnControls.addEventListener("click", () => {

```

```
        if (video.controls === true)
            video.controls = false;

        else
            video.controls = true;
    });
```

## SEMINAR 13-JOC CU CARAMIZI

...

## PROIECTUL CU CANVAS EDITOR FOTO

### 1. Stilul textului

```
editor.context.fillStyle = 'red';
```

```
editor.context.font = "20px Georgia"
```

```
editor.context.fillText('Trage o imagine pe canvas', editor.canvas.width / 2 - 90, editor.canvas.height / 2, 200);
```

### 2. Canvas proportionat, pastrarea proportiilor la redimensionare

```
ratio = editor.image.naturalHeight / editor.image.naturalWidth;
editor.canvas.width = 400;
editor.canvas.height = editor.canvas.width * ratio;
editor.context.drawImage(editor.image, 0, 0, editor.canvas.width,
editor.canvas.height);
```

### 3. Canvas cat imaginea:

```
canvasImage.width = img.naturalWidth;

canvasImage.height = img.naturalHeight;
```

### 4. Drag and drop:

```
editor.canvas.addEventListener("dragover", function (event) {
    event.preventDefault();
```

```

});

editor.canvas.addEventListener("drop", function (event) {
    var fisierePrimite = event.dataTransfer.files;
    var fisier = fisierePrimite[0]; //primul fisier

    var cititor = new FileReader();
    cititor.onload = function (event) {

        editor.image = document.createElement("img");
        editor.image.addEventListener("load", function () {

            editor.context.clearRect(0, 0, editor.canvas.width, editor.canvas.height);

            ratio = editor.image.naturalHeight / editor.image.naturalWidth;
            editor.canvas.width = 400;
            editor.canvas.height = editor.canvas.width * ratio;
            editor.context.drawImage(editor.image, 0, 0, editor.canvas.width,
            editor.canvas.height);

            salveazaStareCurenta();

        });

        editor.image.src = event.target.result; //cand se termina functia load se deseneaza poza
        pe canvas
    };
    cititor.readAsDataURL(fisier); //cand se termina functia load ptr cititor

    event.preventDefault();
});

```

##### 5. **Incarcare poza din fileBrowser:**

```

<input id="fileBrowser" type="file" accept="image/*">

//INCARCARE POZA DIN FILE BROWSER
document.getElementById("fileBrowser").addEventListener("change", function (event) {
    var fisierePrimite = event.target.files;

    var cititor = new FileReader();
    cititor.addEventListener("load", function (event) {

```

```

        editor.image = document.createElement("img");
        editor.image.addEventListener("load", function (event) {

            editor.context.drawImage(editor.image, 0, 0, editor.canvas.width,
            editor.canvas.height);

        });

        editor.image.src = event.target.result;
    });
    cititor.readAsDataURL(fisierePrimate[0]);

});

```

#### 6. DrawImage a treia implementare cu source image:

```

//si desenez in canvas doar zona selectata
editor.context.drawImage(editor.image,
    editor.selectiaMea.x, editor.selectiaMea.y,
    editor.selectiaMea.width, editor.selectiaMea.height,
    0, 0,
    editor.canvas.width, editor.canvas.height);

```

#### 7. Desenare elipsa sau cerc

```

let x = canvas.width / 2 - 25;
    let y = canvas.height / 2 - 25;
    context.beginPath();
    context.arc(x, y, 50, 0, 2 * Math.PI);
    context.stroke();

    let x1 = canvas.width / 2 - 30;
    let y1 = canvas.height / 2 - 50;
    context.beginPath();
    context.ellipse(x1, y1, 60,100,0, 0, 2 * Math.PI);
    context.stroke();

```

#### 8. Rotit canvas la 180 de grade

```

context.translate(canvas.width,canvas.height);
    context.rotate(Math.PI);
context.rotate(Math.PI); //rotatie la 180 de grade

```

```
context.translate(0,-this.canvas.height) //translatie pe verticala, in sus, cu valoarea inaltimei
```

9. Rotire scris in canvas la 90 de grade:

```
context.translate(canvas.width, -canvas.height/2);  
context.rotate(Math.PI/2)
```

10. Afisare video pe canvas cu filtru gri

```
let requestId = null;  
video.addEventListener("play", function () {  
    requestId = requestAnimationFrame(draw);  
})  
video.addEventListener("pause", function () {  
    cancelAnimationFrame(requestId);  
})  
  
function draw() {  
    canvas.width = video.videoWidth;  
    canvas.height = video.videoHeight;  
    context.drawImage(video, 0, 0);  
  
    let imageData = context.getImageData(0, 0, canvas.width, canvas.height);  
    let data = imageData.data;  
  
    for (let i = 0; i < data.length; i += 4) {  
        const r = data[i];  
        const g = data[i + 1];  
        const b = data[i + 2];  
  
        const avg = Math.round((r + g + b) / 3);  
        data[i] = data[i + 1] = data[i + 2] = avg;  
    }  
  
    context.putImageData(imageData,0,0);  
    requestId = requestAnimationFrame(draw);  
}
```

11. Functie cu parametru:

```
buton.addEventListener("click", function () {  
    /**  
     * @param {HTMLImageElement} img  
     */
```

```

        draw(img)

    })
    function draw(img) {

        canvas.width = img.naturalWidth;
        canvas.height = img.naturalHeight;
        context.drawImage(img, 0, 0);

        let imageData = context.getImageData(0, 0, canvas.width, canvas.height);
        let data = imageData.data;

        for (let i = 0; i < data.length; i += 4) {
            let r = data[i];
            let g = data[i + 1];

            data[i] = g;
            data[i + 1] = r;
        }
        context.putImageData(imageData, 0, 0);

        context.strokeStyle="green";
        context.strokeRect(canvas.width/2-200, canvas.height/2-300,
        400,600);

    }

```

12. Coltul din stanga jos al canvasului

```

for (let x = canvas.height - 200; x < canvas.height; x++) {
    for (let y = 0; y < 200; y++) {

        let i = x * canvas.width * 4 + y * 4;
        let r = data[i];
        let g = data[i + 1];
        let b = data[i + 2];

        data[i] = data[i + 1] = data[i + 2] = Math.round((r + g + b)
/ 3);
    }
}

```

13. Button pe play pause audio:

```

btn.addEventListener("click", function () {

```

```

        if (audio.paused) {
            audio.play();
            btn.innerHTML = "Pause";
        }
        else {
            audio.pause();
            btn.innerHTML = "Play";
        }
    });

```

#### 14. Afisare video pe canvas:

```

video.addEventListener("play", function () {
    requestAnimationFrame(draw);
});

function draw() {

    canvas.width = video.videoWidth;
    canvas.height = video.videoHeight;

    context.drawImage(video, 0, 0);

    context.textAlign = "center";
    context.font="25px Georgia"
    context.fillText(video.src,canvas.width/2,canvas.height/2);

    requestAnimationFrame(draw);
}

```

#### 15. Histograma cu rosu

```

btn.addEventListener("click", function () {
    let value = [];
    for (let i = 0; i < 256; i++)
        value[i] = 0;

    canvas.width = img.naturalWidth;
    canvas.height = img.naturalHeight;
    context.drawImage(img, 0, 0);

    let imageData = context.getImageData(0, 0, canvas.width, canvas.height);
    let data = imageData.data;

```



```
for (let i = 0; i < data.length; i += 4) {  
    value[data[i]]++;  
}  
  
context.clearRect(0, 0, canvas.width, canvas.height);  
  
context.font = '25px Arial'  
context.fillStyle = 'green';  
context.fillText('Alexandra', 50, 50)  
  
let max = Math.max(...value);  
let f = canvas.height / max;  
  
let barWidth = canvas.width / value.length  
  
context.rotate(Math.PI);  
context.translate(0, -canvas.height);  
context.scale(-1, f);  
  
for (let i = 0; i < value.length; i++) {  
    context.fillRect(i * barWidth, 0, barWidth * 0.9, value[i]);  
}  
  
})
```