Pau Casesnoves
Centre Integrat de Formació Professional
C/Joan Miró, 22 07300 Inca
Tel. 971 881711
secretaria@paucasesnovescifp.cat

Animations activity

Curs	22/23	**Grup**	**S2WD**	**Data lliurament**	19/03/23 - 23:55
Mòdul	Disseny d'Interfícies Web				
Títol	Animations Activity				

Work type	Individual
Implementation guidelines	

In this activity you must work with 2 procedures to create animations (not much complex ones): one using an specific software (Gimp) and the other using HTML+CSS.

First, take a look about you can achieve with CSS transforms, transitions and animations:

- **CSS 2D Transforms**
- **CSS Transitions**
- **CSS Animations**

1. **Let's start with a simple animated GIF. Create an animation that could be used as a commercial banner in a webpage following this video tutorial, so you only have to apply simple effects like fade-in and fade-out:**

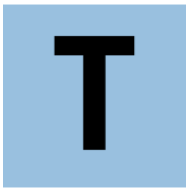
   GIMP 2.10 Tutorial: Create an Animated GIF for Google Adwords

   - **You are free to decide the theme and the whole design.**
   - **The total duration of the animated GIF must be 5 seconds.**
   - **Document your creation process!**
   - **Take in mind that you have recreate (approximately) this animation using HTML+CSS.**

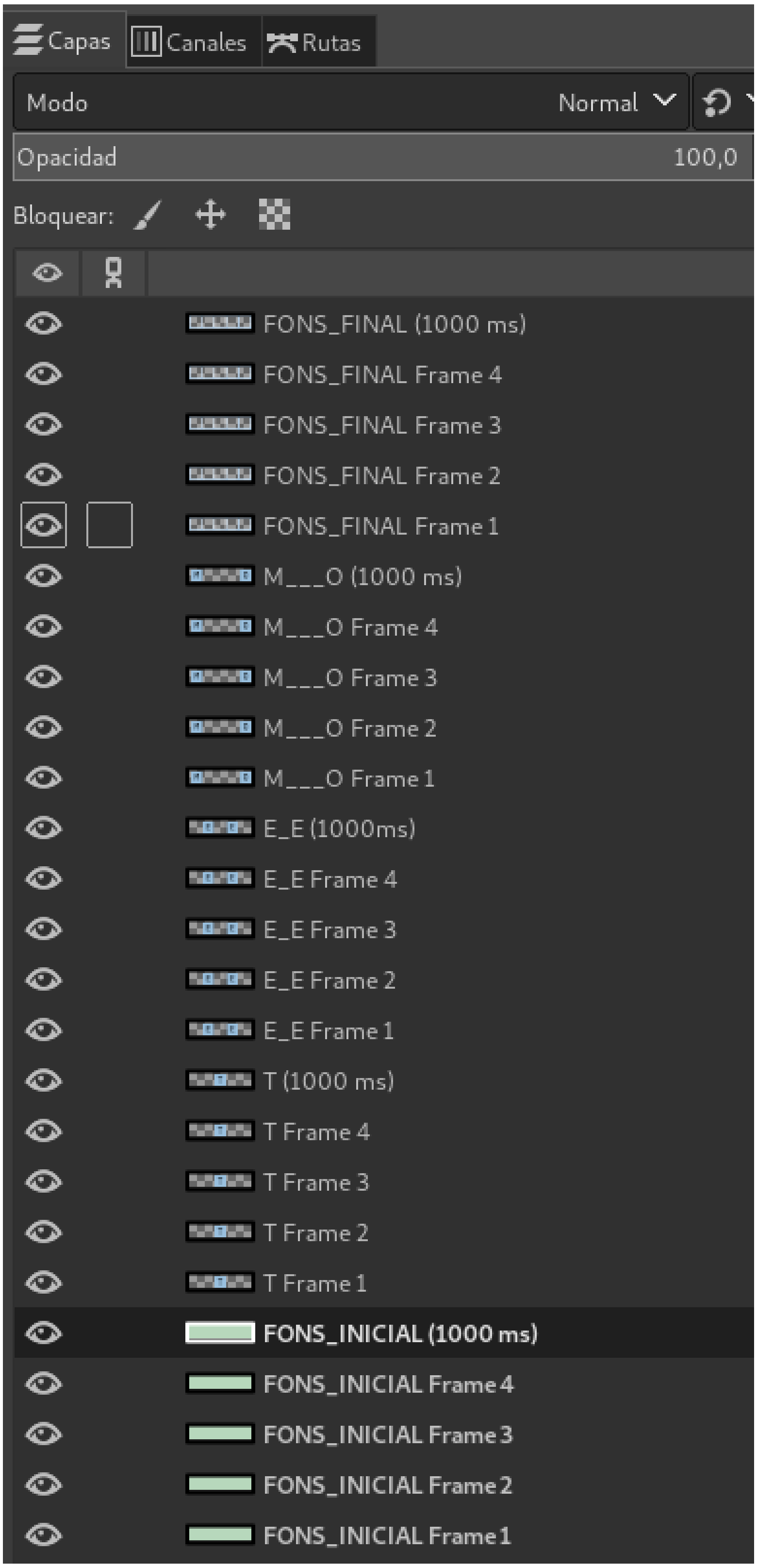
   (3 points)

My commercial banner design is about the fictitious brand name "METEO". The animation consists of the appearance of the different characters that make up the word METEO successively and that in the end the background is completely uniform.

For the design, first I have first the different images that will appear successively:

initial background	
character T	
characters E_E	
characters M___O	
end background	

Later, in a new file, I have added these images as layers indicating the periods of time in which they should appear as well as their gradual appearance

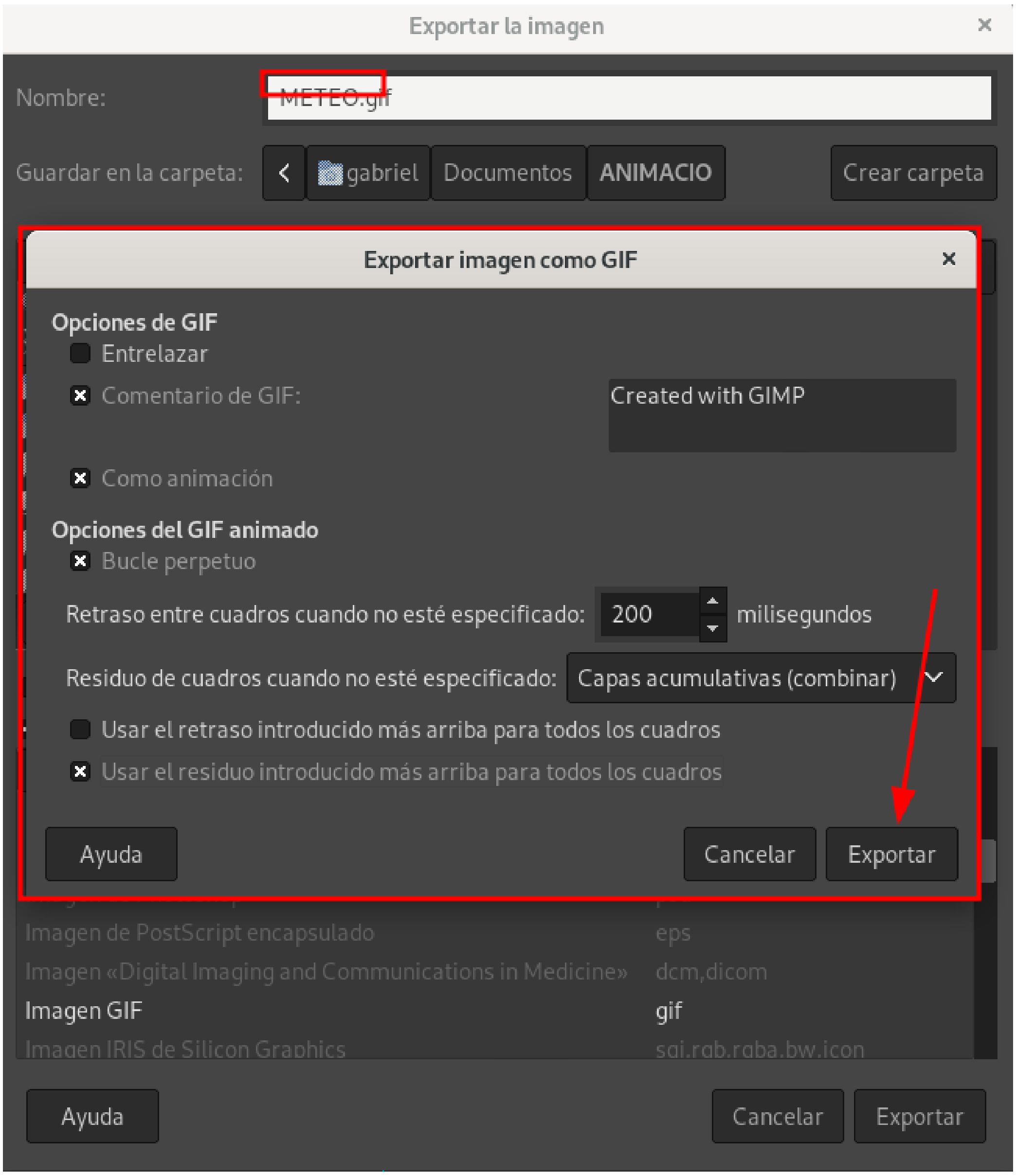


Modo Normal

Opacidad 100,0

FONS_FINAL (1000 ms)
FONS_FINAL Frame 4
FONS_FINAL Frame 3
FONS_FINAL Frame 2
FONS_FINAL Frame 1
M___O (1000 ms)
M___O Frame 4
M___O Frame 3
M___O Frame 2
M___O Frame 1
E_E (1000ms)
E_E Frame 4
E_E Frame 3
E_E Frame 2
E_E Frame 1
T (1000 ms)
T Frame 4
T Frame 3
T Frame 2
T Frame 1
FONS_INICIAL (1000 ms)
FONS_INICIAL Frame 4
FONS_INICIAL Frame 3
FONS_INICIAL Frame 2
FONS_INICIAL Frame 1

Finally, we export the created animation to GIF format:



Exportar la imagen

Nombre: METEO.gif

Guardar en la carpeta: gabriel / Documentos / ANIMACIO

Exportar imagen como GIF

Opciones de GIF

☐ Entrelazar

☒ Comentario de GIF: Created with GIMP

☒ Como animación

Opciones del GIF animado

☒ Bucle perpetuo

Retraso entre cuadros cuando no esté especificado: 200 milisegundos

Residuo de cuadros cuando no esté especificado: Capas acumulativas (combinar)

☐ Usar el retraso introducido más arriba para todos los cuadros

☒ Usar el residuo introducido más arriba para todos los cuadros

Imagen de PostScript encapsulado	eps
Imagen «Digital Imaging and Communications in Medicine»	dcm,dicom
Imagen GIF	gif
Imagen IRIS de Silicon Graphics	sgi,rgb,rgba,bw,icon

This is the final result:

2. **Compare a CSS transform, a CSS transition, and a CSS animation. What are its main properties? When should we use each? Briefly describe how each of these functions can be applied.** (3 points)

CSS Transform, CSS Transition, and CSS Animation are three different techniques that can be used to change the appearance or position of an element on a webpage.

CSS Transform

Transforms are one of the most interesting and powerful CSS features that are introduced into the language to turn style sheets into a system capable of performing 2D and 3D visual effects. With them we can do things like move elements, rotate them, increase or decrease them and other related or combined transformations.

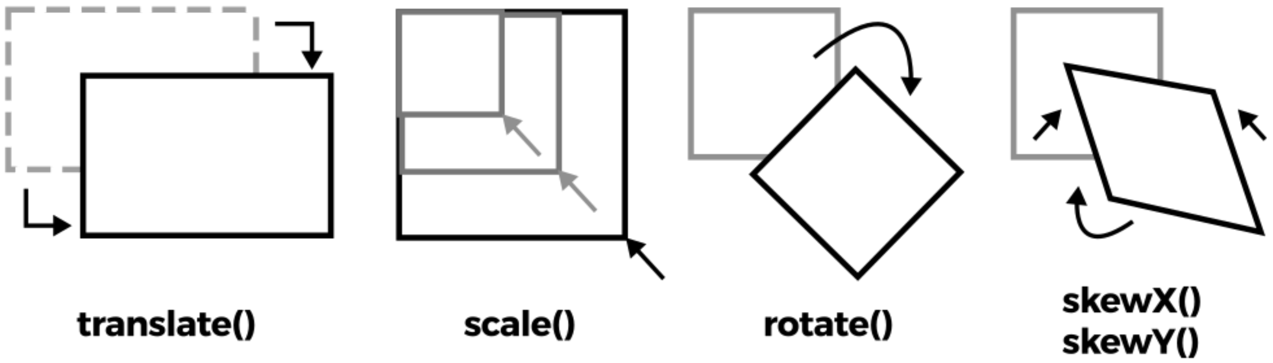
These transformations can be carried out in CSS through the transform property that allows receiving a specific transformation function, which will be applied to the HTML element in question selected through CSS. The HTML element will be visually transformed.

Thus, once we choose one of the possible transformations, we can apply it through this property, choosing its related transformation function. For example, imagine that we want to transform the HTML element with class item and move it 25 pixels to the right and 150 pixels down:

```
.item {
   transform: translate(25px, 150px);
}
```

In this example, a transformation function is applied, specifically translate(), with specific values for X and Y.

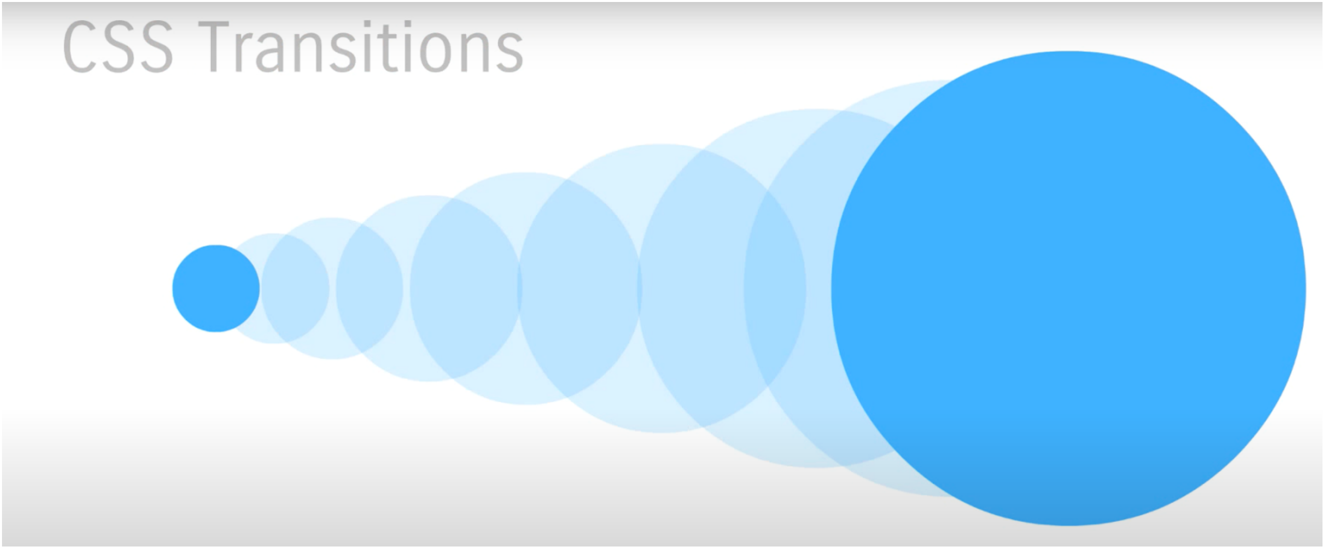
We can perform the following transformations in 2D:



We can also perform transformations in the 3D field.

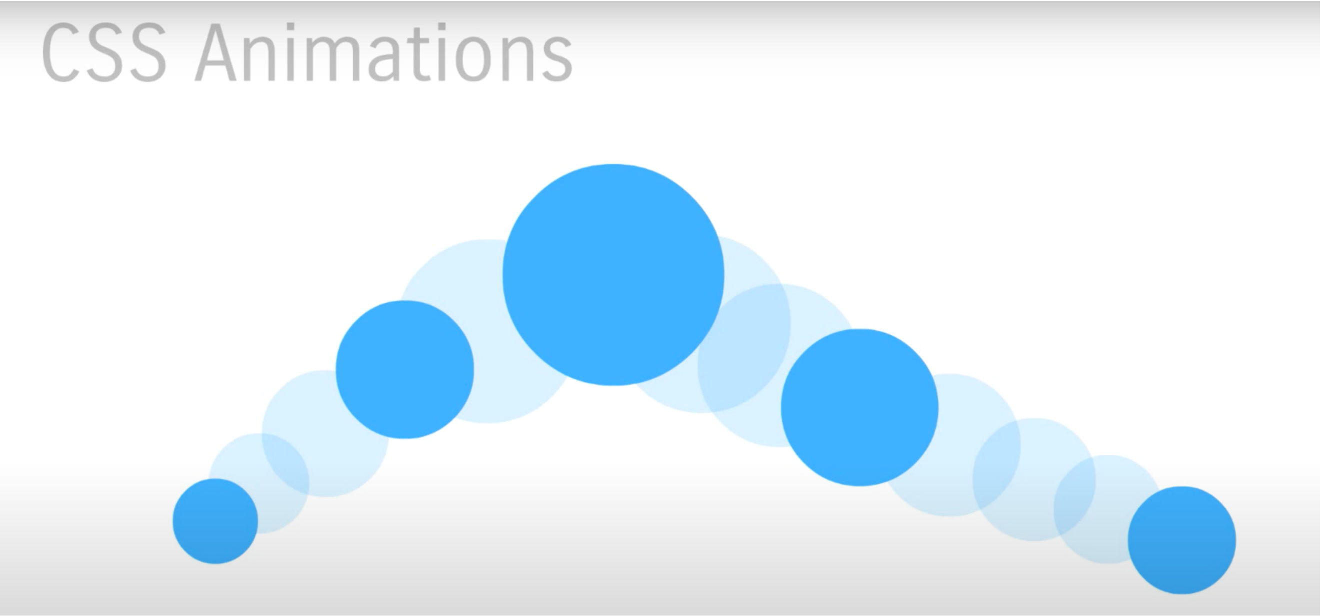
CSS Transition

CSS transitions allow us to change the values of the properties during a certain period of time and whenever they are activated (by placing the cursor over the element). They are used to create smooth transitions between two states of an element, such as when you hover over a button and the color changes. We can configure the duration, the delay and the timing function of the transition.



CSS Animation

CSS animation allows the animation of HTML elements, unlike the transition, which only performs an operation from point A to point B, animations allow intermediate operations, providing them with greater complexity and possibilities. Animations consist of two parts: animation properties to control the time, direction, and duration of the animation, and keyframes to specify the intermediate stages of the animation.



Difference between transitions and animations:

TRANSITION	ANIMATIONS
Transitions cannot be repeated.	Animations have no problem looping.
Transitions need a trigger to execute like hover.	The animation does not need any activation, it starts executing based on the established properties
Transitions are easy to work with in JavaScript.	Animations are difficult to work with in JavaScript.
Transitions animate an object from one point to another.	Animation allows you to define keyframes that vary from state to state with various properties and time frames.

In summary, CSS Transform is used to change the position, rotation, scale, or skew of an element. CSS Transition is used to create smooth transitions between two states of an element, and CSS Animation is used to create more complex animations. You should use CSS Transform when you want to manipulate an element's position, CSS Transition when you want to create smooth transitions between two states, and CSS Animation when you want to create more complex animations.

3. **Now, using HTML+CSS try to reproduce the animated GIF that you create at exercise 1. You can do it using just a single HTML file.** (4 points)

In this case, I have modified the order in which the characters of the trademark "METEO" appear and I have also increased the duration of the animation. Even so, the concept of the animation and the final result is the same as with the GIF file.

In the HTML file we can highlight:

```
<body>
        <div class="missatge">
            <div class="flex-items c1">M</div>
            <div class="flex-items c2">E</div>
            <div class="flex-items c3">T</div>
            <div class="flex-items c4">E</div>
            <div class="flex-items c5">O</div>
        </div>
</body>
```

In the div with class="missatge" I have grouped the 5 characters of "METEO", each one in an individual div.

In the CSS block it is worth mentioning the animation property:

div	**content**
missatge	animation: canviafons 4s linear 16s 1 forwards;
c1	animation: mostracaracter 2s ease-in-out 2s 7 forwards;
c2	animation: mostracaracter 2s ease-in-out 4s 6 forwards;
c3	animation: mostracaracter 2s ease-in-out 6s 5 forwards;
c4	animation: mostracaracter 2s ease-in-out 8s 4 forwards;
c5	animation: mostracaracter 2s ease-in-out 10s 3 forwards;

The parameters contained in animation refer to:

- animation-name
- animation-duration
- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-fill-mode

Another important aspect is the keyframes that mark the beginning and end of each animation:

div	keyframe

missatge	`@keyframes canviafons {`
	`    from {`
	`        background-color: #bbd6ba;`
	`    }`
	`    to {`
	`        background-color: #99c0df;`
	`    }`
	`}`

c1, c2, c3, c4 i c5	`@keyframes mostracaracter {`
	`    from {`
	`        background-color: #bbd6ba;`
	`        color: #bbd6ba;`
	`    }`
	`    to {`
	`        background-color: #99c0df;`
	`        color: #000000;`
	`    }`
	`}`

This is the content of the html file

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .missatge {
                margin: 50px;
                display: flex;
                flex-direction: row;
                text-align: center;
                align-items: center;
                background-color: #bbd6ba;
                color: #000000;
                font-weight: bold;
                justify-content: space-around;
                font-family: monospace;
                width: 1120px;
                height: 240px;
                animation: canviafons 4s linear 16s 1 forwards;
            }

            .flex-items {
                width: 180px;
                height: 180px;
                font-size: 154px;
                background-color: #bbd6ba;
            }

            .c1 {
                background-color: #bbd6ba;
                color: #bbd6ba;
                animation: mostracaracter 2s ease-in-out 2s 7 forwards;
            }

            .c2 {
                background-color: #bbd6ba;
                color: #bbd6ba;
                animation: mostracaracter 2s ease-in-out 4s 6 forwards;
            }

            .c3 {
                background-color: #bbd6ba;
                color: #bbd6ba;
                animation: mostracaracter 2s ease-in-out 6s 5 forwards;
            }

            .c4 {
                background-color: #bbd6ba;
                color: #bbd6ba;
                animation: mostracaracter 2s ease-in-out 8s 4 forwards;
            }

            .c5 {
                background-color: #bbd6ba;
                color: #bbd6ba;
                animation: mostracaracter 2s ease-in-out 10s 3 forwards;
            }

            @keyframes mostracaracter {
                from {
                    background-color: #bbd6ba;
                    color: #bbd6ba;
                }
                to {
                    background-color: #99c0df;
                    color: #000000;
                }
            }

            @keyframes canviafons {
                from {
                    background-color: #bbd6ba;
                }
                to {
                    background-color: #99c0df;
                }
            }
        </style>
    </head>
    <body>
        <div class="missatge">
            <div class="flex-items c1">M</div>
            <div class="flex-items c2">E</div>
            <div class="flex-items c3">T</div>
            <div class="flex-items c4">E</div>
            <div class="flex-items c5">O</div>
        </div>
    </body>
</html>
```

To deliver:

- A single document with all the statements, your answers and explanations.
- An animated GIF for exercise 1
- An HTML file with the solution to exercise 3

Justifying your decisions is very important. Always explain how you get to the delivered solution for each statement. Without your explanation it can not be evaluated.

Create a ZIP with all the files and upload to the activity space.