

Coding animations and interactivity.

Using CSS & JavaScript. Individual assignment week 43

The Assignment

Overview

You'll be building a web page containing a character (that you draw or find online) and 15 buttons. When the user clicks a button, an animation starts. Consider the exercise split into two parts. Tuesday you'll build the HTML and the CSS for the animations, and Wednesday you'll add JavaScript to control the animations.

Tuesday

1. Set up a webpage with a "scene" (just add a `<div>` with a border)
2. Add a single stylesheet to the HTML file
3. Add the character (``) to the scene
4. Add a number of buttons underneath the scene (`<button>Jump</button>`)
5. Keep reading to see which animations we expect:

Make all your animations in the same file. It can get a bit overwhelming with all these buttons, animations and JavaScript snippets, so try to maintain a clear structure in your documents. Use meaningful names for classes, keyframe-animations, buttons and functions, and add lots of explanatory comments in the CSS, HTML and JavaScript.

Make one animation at the time, following this order

1. Add a meaningful class to the character.
 - a. If the goal is to make the character jump, give it a class of jump
2. Add the CSS that gives it the desired animation.
3. Remove the class from the character so you can continue on the next.
 - a. Leave the class in the stylesheet, you'll use all the classes on Thursday.

Wednesday

Now it's time to hook up the classes dynamically. When a button is clicked, the character should animate. In general, each button will start a specific animation. The animations won't be linked together, therefore you WILL encounter some problems when adding a second or third animation to the character. Don't worry in the first place. Try to figure it out later - or just refresh the web page. You'll be using the JavaScript covered later to finish the assignment. Here are our suggestions on how to proceed:

1. Select the character (`document.querySelector(...)`) and store **that** in a variable.

2. Select a button in the same manner.
3. Add an eventListener to that button.
4. When the button is clicked, call a function that adds a class to the character (thereby animating it).
5. The user should be able to stop the animation.
 - a. Two options
 - i. Repeat the above process to remove the class or
 - ii. Use the same function and classList.toggle to start/stop the animation.
6. Go back to step 2.

Requirements

There are listed a total of 15 animations and you have to complete at least 10. Challenge yourself!

Graphics

Use a character you draw or find online. (PNG, JPG, other)

Animations

Every animation you make must have a start and a stop button (classList.add, classList.remove) beneath the scene, or a toggle button that handles both.

If one of the animations proves to be too much, skip it and go to the next.

The names in parenthesis are suggestions for naming the animations (keyframes, classes, buttons etc., use them as you see fit)

1. (**move**) Make the character move from the left side of the screen (outside) to the right side of the screen (outside) continually.
2. (**moveTo30**) Make the character move once from just outside the left side to 30% in to the scene.
Make sure the character slows down before it stops
Make sure the character stays at its final position
3. (**moveFrom30**) Make the character move from 30% into the scene to just outside the right side.
Make sure the character picks up speed when he starts, and that he stays put at the end.
4. (**oneJump**) Make the character jump once, and make him fall back nicely
5. (**jumping**) Make the character jump up and down.
Hint: Remember that an animation can shift direction (back, forth, back, forth...)
6. (**fade**) Make the character fade in and out
7. (**glow**) Make the character glow, as if it has a light behind. Fade it in and out

8. (**fallDown**) Make the character fall down, as if he's been shot.
Hint: Give the character a reasonable rotation point (transformY origin)
9. (**speak**) Make the character speak, or at least, make it look like he's speaking. Move it a little, rotate a little etc.)
10. (**shake**) Make the character shake 3-4 times
11. (**mirror**) Mirror the character (can be done without an animation)
12. (**moveTo0**) Move the character to the left side of the scene.
Hint: if you make your animation without "from", it doesn't care where it starts

The following suggestions require an additional character on the scene (it could be as simple as a <div> with a background color).

13. (**hitter**) Make one character hit the other (eventually a head butt)
14. (**gotHit**) Make the character that got hit react in some way. Make the animation different than **fallDown**
15. And finally, find something fun or cool to do with CSS & JS

Hand in

Upload your files to your domain, and hand in a link on Fronter in the folder "03.01.02 link to simple animation" before 26/10, 23:59

Have fun, Alan & Julio