## (1.) Why is it called a pseudo-class?

Ans. A pseudo-class is a CSS selector that is used to select elements based on their state or attributes, rather than their type or ID. It is called "pseudo" because it does not represent a real class in the HTML document, but instead selects elements based on their virtual state.

Examples of pseudo-classes include :hover, :active, and :focus. They are used to style elements when they are in a certain state, such as when a user hovers over a link or a button is being clicked.

## (2.) What are gradients in CSS?

Ans. In CSS, gradients are used to create a smooth transition between two or more colors in an element.

=> Gradients can be applied to backgrounds, borders, and other visual properties of an element.

=> CSS defines three types of gradients:

- Linear Gradients (goes down/up/left/right/diagonally)
- Radial Gradients (defined by their center)
- Conic Gradients (rotated around a center point)

**(i) *Linear Gradients :*** A linear gradient is a transition between two or more colors along a straight line.

=> The linear-gradient function is used to define a linear gradient, and it takes a number of arguments to specify the direction and colors of the gradient.

=> To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

**Syntax :** background-image: linear-gradient(direction, color-stop1, color-stop2, ...);

**(ii) *Radial Gradients :*** A radial gradient is a transition between two or more colors that emanates from a single point.

=> The radial-gradient function is used to define a radial gradient, and it takes arguments to specify the position and size of the gradient, as well as the colors.

=> A radial gradient is defined by its center.

=> To create a radial gradient you must also define at least two color stops.

**Syntax :** background-image: radial-gradient(shape size at position, start-color, ..., last-color);

By default, shape is ellipse, size is farthest-corner, and position is center.

**(iii) *Conic Gradients :*** A conic gradient is a gradient with color transitions rotated around a center point.

=> To create a conic gradient you must define at least two colors.

**Syntax :** background-image: conic-gradient([from angle] [at position,] color [degree], color [degree], ...);

By default, angle is 0 degree and position is center.

If no degree is specified, the colors will be spread equally around the center point.

**NOTE :** Gradients can be used to create a wide range of visual effects, such as backgrounds with a subtle color transition, buttons with a glossy effect, and more.

## (3.) What are different types of transitions in CSS?

Ans. In CSS, transitions are used to create smooth changes in the visual properties of an element, such as its color, size, or position, over a period of time.

=> There are four types of transitions in CSS:

**(i) Property transition:** This type of transition applies to a specific CSS property, such as the color or width of an element. For example, you can use a transition to smoothly change the color of a button when a user hovers over it.

**(ii) Height and Width transition:** This type of transition applies to the height and width of an element.
**(iii) Transform transition:** This type of transition applies to the transformation of an element, such as rotation, scaling, or skewing.
(iv) Translate transition: This type of transition applies to the position of an element by moving it along the x and y-axis.
**=>** These transitions are defined using the transition property and its associated sub-properties such as transition-property, transition-duration, transition-timing-function, and transition-delay.
**=>** In addition to these basic types of transitions, CSS also provides some additional ways to create more complex animations, such as the animation property and the @keyframes rule.

**(4.) Animations in CSS :** An animation lets an element gradually change from one style to another.
**=>** You can change as many CSS properties you want, as many times as you want.
**=>** To use CSS animation, you must first specify some keyframes for the animation.
**=>** Keyframes hold what styles the element will have at certain times.
**=> The @keyframes Rule :** When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
**=>** To get an animation to work, you must bind the animation to an element.
**This property is a shorthand for the following CSS properties:**

- animation-delay.
- animation-direction.
- animation-duration.
- animation-fill-mode.
- animation-iteration-count.
- animation-name.
- animation-play-state.
- animation-timing-function.

**=>** In CSS, animations are used to create complex motion effects that can be applied to elements on a web page.
**=>** Animations can be used to animate the changes in visual properties of an element, such as its color, size, or position, over a period of time.
**=>** The animation property is used to define the basic properties of an animation, such as its duration, delay, and iteration count.
**=>** The @keyframes rule is used to define the specific steps of the animation, including the starting and ending states of the element and the intermediate states that will be animated.
**=>** CSS animations can be used to create a wide range of effects, such as simple hover effects, complex scrolling animations, and more.
**=>** The animations can be controlled by using javascript or by using css pseudo-classes such as :hover, :active, and :focus.
For example, you can create an animation that makes an element move across the screen by defining a keyframe animation that starts with the element at one position, and ends with the element at another position. And then you can apply the animation to an element by setting the animation-name property to the name of the keyframe animation and the animation-duration property to specify how long the animation should take to complete.