1. What is the difference between transition and animation in CSS?

Answer:

- **Transition** changes properties smoothly over time, requires a trigger (hover, focus, etc.), and defines only start & end states.
- **Animation** uses @keyframes to define multiple states, can run automatically, loop infinitely, and does not require user interaction.

2. How do you write a basic CSS transition?

```
.button {

  transition: background-color 0.3s ease-in-out;

}

.button:hover {

  background-color: red;

}
```

Parts of transition:

- Property (background-color)
- Duration (0.3s)
- Timing function (ease-in-out)
- Delay (optional)

3. What are transform functions in CSS?

Answer:
transform changes an element's shape, size, or position in 2D/3D space.
Common functions:

- translate(x, y) → Move
- rotate(angle) → Rotate
- scale(x, y) → Resize
- skew(x, y) → Tilt

4. Difference between translate() and position: relative movement?

Answer:

- translate() moves visually but does not affect document flow.
- position: relative moves the element and shifts its layout position.

5. What is the difference between @keyframes and animation shorthand?

Answer:

- @keyframes → Defines animation stages.
- animation → Applies animation to an element (name, duration, timing, delay, iteration, direction, fill-mode).

6. How do you pause and play CSS animations?

```
.element {

  animation-play-state: paused; /* or running */

}
```

7. What are CSS variables?

Answer:
Custom properties defined with --var-name and accessed using var().
Example:

```
:root { --main-color: #333; }

p { color: var(--main-color); }
```

8. Difference between local and global CSS variables?

Answer:

- **Global:** Declared in :root, available everywhere.
- **Local:** Declared inside a selector, available only in that selector.

9. What is CSS specificity order?

Answer:
Inline styles > IDs > Classes/Attributes/Pseudo-classes > Tags/Pseudo-elements > Inherited styles.
!important overrides all (avoid if possible).

10. How does the :is() pseudo-class work?

Answer:
Matches any selector in a list and takes highest specificity of them.
Example:

```
:is(h1, h2, h3) { color: red; }
```

11. How does the :has() pseudo-class work?

Answer:
Selects an element if it contains another element matching a selector.
Example:

```
article:has(img) { border: 1px solid #000; }
```

12. How does the :not() pseudo-class work?

Answer:
Excludes elements matching a selector.
Example:

```
p:not(.highlight) { color: gray; }
```

13. How does the :where() pseudo-class work?

Answer:
Same as :is() but has **zero specificity**.
Example:

```
:where(h1, h2, h3) { margin: 0; }
```

14. What is accent-color in CSS?

Answer:
Changes default form controls' highlight color (checkboxes, radios, etc.).
Example:

```
input[type="checkbox"] { accent-color: green; }
```

15. What are container queries?

Answer:
Allow component styles based on **container size**, not viewport.

```
@container (min-width: 500px) {

  .card { font-size: 1.2rem; }
```

}

16. Difference between container queries and media queries?

Answer:

- **Media queries** → React to **viewport** size.
- **Container queries** → React to **parent container** size.

17. What is float in CSS?

Answer:
Moves elements left/right allowing inline content to wrap around them.
Best for text wrapping, not layouts.

18. How do you clear floats?

.clearfix::after {

content: "";

display: block;

clear: both;

}

19. What is clip-path in CSS?

Answer:
Defines a visible shape, hiding parts outside it.
Example:

img { clip-path: circle(50%); }

20. What are some common clip-path shapes?

- circle()
- ellipse()
- polygon()
- inset()

21. Difference between clip-path and mask?

Answer:

- clip-path → Shape-based clipping.

- mask → Pixel-based transparency control.

22. What are vendor prefixes?

Answer:
Browser-specific keywords for experimental CSS features.
Example:

-webkit-transition: all 0.3s;

-moz-transition: all 0.3s;

23. Common vendor prefixes?

- -webkit- → Chrome, Safari
- -moz- → Firefox
- -o- → Old Opera
- -ms- → IE/Old Edge

24. Why use Autoprefixer?

Answer:
Automatically adds/removes prefixes based on browser support using PostCSS.

25. How do you center an element horizontally using Flexbox?

```
.parent {

  display: flex;

  justify-content: center;

}
```

26. How do you center an element both horizontally and vertically with Flexbox?

```
.parent {

  display: flex;

  justify-content: center;

  align-items: center;

}
```

27. How do you create smooth hover effects with transitions?

```
a {
  color: black;
  transition: color 0.3s ease;
}
a:hover {
  color: red;
}
```

28. How do you make an infinite CSS animation?

```
@keyframes spin {
  100% { transform: rotate(360deg); }
}
.loader {
  animation: spin 2s linear infinite;
}
```

29. What are timing functions in CSS transitions/animations?

Answer:
They control speed changes over time.
Examples: linear, ease, ease-in, ease-out, ease-in-out, cubic-bezier().

30. How do you make reusable themes using CSS variables?

```
:root {
  --bg-color: white;
  --text-color: black;
}
.dark-theme {
  --bg-color: black;
```

```
  --text-color: white;
}
body {
  background: var(--bg-color);
  color: var(--text-color);
}
```