CSS Interview Day5 Q&A

1. Show implementation of CSS transitions.

The transition-property specifies the CSS properties to which you want the transition effect. Only these CSS properties are animated.

Code:-

```
div {
     width: 100px;
     height: 100px;
     background: lightblue;
     transition-property: width;
     transition-duration: 2s;
     transition-timing-function: linear;
     transition-delay: 1s;
     }

div: hover{
            width: 300px;
        }
```

2. Show implementation of CSS transform.

CSS **transforms** are a collection of *functions* that allow to **shape elements** in particular ways:

- **translate**: moves the element along up to 3 axis (x, y and z)
- **rotate**: moves the element around a central point
- **scale**: resizes the element
- **skew**: distorts the element

Code:-

```
@keyframes translating {
            0% {
                transform: translate (0, 0);
            }

            25% {
                transform: translate (240px, 0);
            }

            50% {
                transform: translate (240px, 140px);
            }

            75% {
                transform: translate (0, 140px);
            }

            100% {
                transform: translate (0, 0);
            }
        }

        p {
            animation: translating 4s linear infinite;
        }
```

3. Show implementation of Grid.

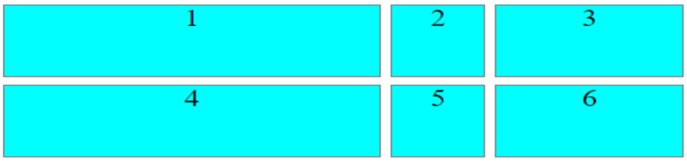
CSS:-

```
.container {
            display: grid;
            grid-template-columns: 200px 50px 100px;
            grid-template-rows: 50px 50px;
            grid-gap: 5px;
            text-align: center;
            justify-content: center;
        }

        .item{
            background-color: aqua;
            border: 1px solid gray;

        }
        .item1 {
            grid-column-start: 1;
            grid-column-end: 3;
        }

        .item3 {
            grid-row-start: 2;
            grid-row-end: 4;
        }

        .item4 {
            grid-column-start: 2;
            grid-column-end: 4;
        }
```

HTML:-

```
 <div class="container">
        <div class="item item-1">1</div>
        <div class="item item-2">2</div>
        <div class="item item-3">3</div>
        <div class="item item-4">4</div>
        <div class="item item-5">5</div>
        <div class="item item-6">6</div>
    </div>
```

Output:-



```
1  2  3
4  5  6
```

4. What are CSS Combinators?

Combinators combine the selectors to provide them a useful relationship and the position of content in the document. This comes in handy when you want to style certain elements all at once rather than styling elements individually. CSS combinator selectors are as follows.

1. Descendant Selector (space)
2. Child Selector (>)
3. Adjacent Sibling Selector (+)
4. General Sibling Selector (~)

Descendant Selector : The descendant selector matches all elements that are descendants of a specified element.

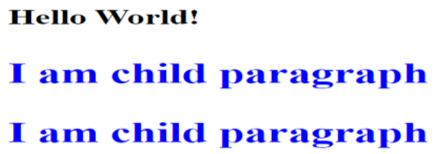
HTML :

```
<div>
  <h2>Hello World!</h2>
  <p>I am child paragraph</p>
  <p>I am child paragraph</p>
 </div>
```

CSS :

```
 div p {  /* --> (div is the parent and the p is the child) */
            color: blue;
            font-family: 'Times New Roman';
            font-weight: bolder;
            font-size: xx-large;
        }
```

Output:


Hello World!

I am child paragraph

I am child paragraph

Child Selector : The child selector will only affect elements under a direct parent. A good way to remember this is that a child will only listen to its direct parents.

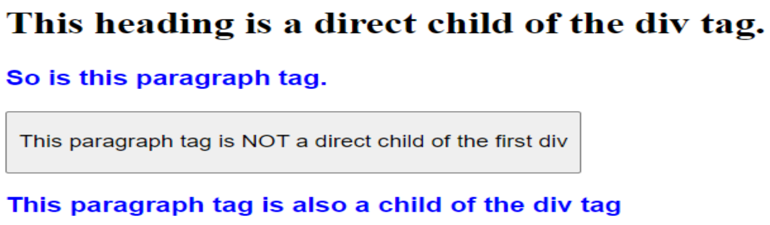
HTML :

```
    <div>
        <h2>This heading is a direct child of the div tag.</h2>
        <p>So is this paragraph tag.</p>
        <button>
            <p>This paragraph tag is NOT a direct child of the first div</p>
        </button>
        <p>This paragraph tag is also a child of the div tag</p>
    </div>
```

CSS :

```
 div > p   {
            color: blue;
            font-family: sans-serif;
            font-weight: bold;
           }
```

Output :


This heading is a direct child of the div tag.

So is this paragraph tag.

This paragraph tag is NOT a direct child of the first div

This paragraph tag is also a child of the div tag

Adjacent Sibling Selector : To use the adjacent sibling selector in your CSS stylesheet, you simply need the plus (+) sign. The adjacent sibling selector is used to select an element that is directly after another specific element.

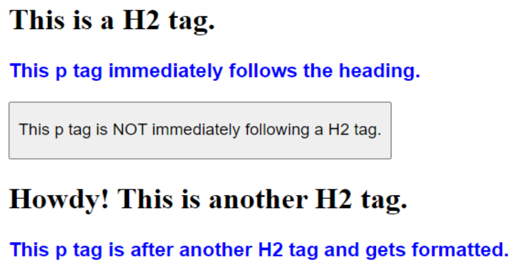
HTML :

```
    <div>
        <h2>This is a H2 tag.</h2>
        <p>This p tag immediately follows the heading.</p>
        <button>
            <p>This p tag is NOT immediately following a H2 tag.</p>
        </button>
        <h2>Howdy! This is another H2 tag.</h2>
        <p>This p tag is after another H2 tag and gets formatted.</p>
    </div>
```

CSS :

```
h2+p {
            color: blue;
            font-family: sans-serif;
            font-weight: bold;
        }
```

Output :



This is a H2 tag.

This p tag immediately follows the heading.

This p tag is NOT immediately following a H2 tag.

Howdy! This is another H2 tag.

This p tag is after another H2 tag and gets formatted.

General Sibling Selectors :The general sibling selector follows through on all elements under the targeted HTML element selected. This means that if another element tag interrupts the direct adjacent flow, the style rules can still follow through.

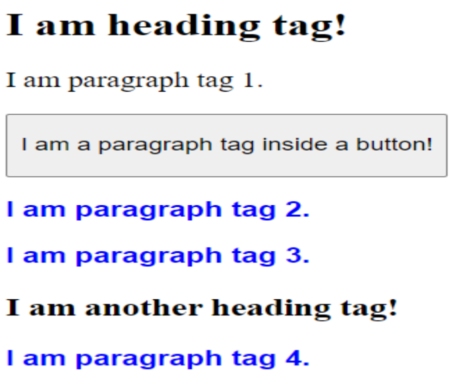
HTML :

```
<div>
        <h2>I am heading tag!</h2>
        <p>I am paragraph tag 1.</p>
        <button>
            <p>I am a paragraph tag inside a button!</p>
        </button>
        <p>I am paragraph tag 2.</p>
        <p>I am paragraph tag 3.</p>
        <h3>I am another heading tag!</h3>
        <p>I am paragraph tag 4.</p>
    </div>
```

CSS :

```
 button~p {
            color: blue;
            font-family: sans-serif;
            font-weight: bold;
        }
```

Output :



I am heading tag!

I am paragraph tag 1.

I am a paragraph tag inside a button!

I am paragraph tag 2.

I am paragraph tag 3.

I am another heading tag!

I am paragraph tag 4.

5. What are CSS Pseudo Selectors?

CSS pseudo-classes are used to add styles to selectors, but only when those selectors meet certain conditions. A pseudo class is expressed by adding a colon (:) after a selector in CSS, followed by a pseudo-class such as "hover", "focus", or "active", like this:

```
a:hover {
```

```
           /*add your style here */
        }
```

Syntax:

```
selector:pseudo-class {
           property : value;
        }
```

```
Some Examples of Pseudo-Class:
```

:active	It is used to add style to an active element.
:hover	It adds special effects to an element when the user moves the mouse pointer over the element.
:link	It adds style to the unvisited link.
:visited	It adds style to a visited link.
:lang	It is used to define a language to use in a specified element.
:focus	It selects the element which is focused by the user currently.
:first-child	It adds special effects to an element, which is the first child of another element.