CSS Functions

1. **CSS Functions:**
    a. CSS functions are used to set the various CSS property.
    b. Functions means some predefined set of instructions to perform some specific task.

2. **Types of CSS Functions:**
    a. calc()
    b. attr()
    c. gradients → linear, radial, conic, repeating.
    d. colors → rgb(), rgba(), hsl(), hsla()
    e. max()
    f. min()
    g. var()
    h. url()
    i. drop-shadow()
    j. blur()

3. **calc():**
    a. The *calc()* function performs a calculation to be used as the property value.
    b. **Syntax**:

```
calc(experssion)
```

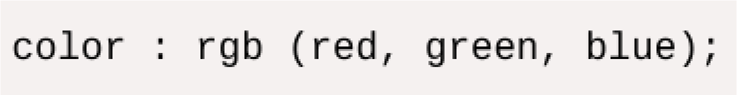
4. **attr():**
    a. The *attr()* function returns the value of the selected elements.

5. **linear-gradient():**
    a. The *linear-gradient()* function sets a linear gradient as the background image.

6. **radial-gradient():**
    a. The *radial-gradient()* function sets a radial gradient as the background image.

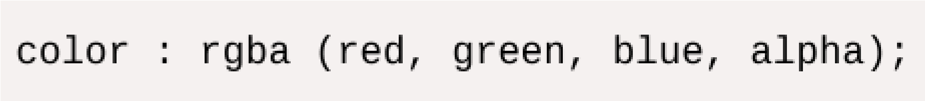
7. **conic-gradient():**
    a. The *conic-gradient()* function sets a conic gradient as the background image.

8. **rgb():**
    a. It is used to define the colors using the Red Green Blue RGB model.
    b. **Syntax**:



```
color : rgb (red, green, blue);
```

9. **rgba():**
    a. It is used to define the colors using the Red-Green-Blue-Alpha RGBA model.
    b. **Syntax**:



```
color : rgba (red, green, blue, alpha);
```

10. **hsl():**
    a. Define the colors using the Hue-saturation-lightness model HSL.

b. **Syntax**:

```
color : hsl (hue, saturation, lightness);
```

11. **hsla():**

a. Define the colors using the Hue Saturation Lightness Alpha HSLA model.

b. **Syntax**:

```
color : hsla (hue, saturation, lightness, alpha);
```

12. **min():**

a. *min()* function in CSS is used to extract the minimum value from a set of comma-separated values.

b. It can take two parameters and a min function can be used inside another min function if the comparison is to be made between multiple values.

c. **Syntax**:

```
min(value1, value2);
min(value1, min(value2, min(value3, value4)));
```

d. **Returns**: This function returns the smallest value from a set of comma separated values.

13. **max():**

a. The *max()* function in CSS is used to set the maximum of the numbers given.

b. It is used to return the largest value from a set of comma-separated values.

c. It can accept length, frequency, integer, angle, and time types of values.

d. **Syntax**:

```
max(value1, value2...);
```

e. **Return Value**: This function returns the largest value from a set of comma separated values.

14. var():

a. The *var()* function is used to insert the value of a CSS variable.

b. **Syntax**:

```
var( custom_property, value )
```

c. **custom_property** : It is the required parameter. The name of the custom property must start with two dashes(-).

d. **value** : It is an optional parameter. It is used if the custom property is invalid.

15. url():

a. The *url()* function is used to include a file.

b. **Syntax**:

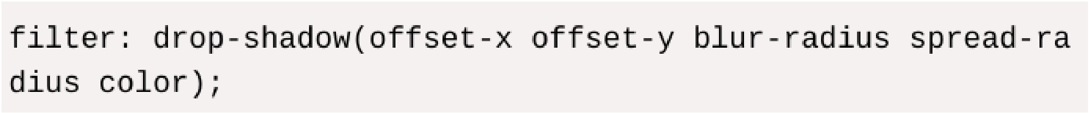
```
url(path);
```

16. **drop-shadow():**

a. The CSS *drop-shadow()* function adds a shadow effect to elements, like images, using horizontal and vertical offsets, blur radius, spread radius, and color parameters.

b. It enhances visual depth and prominence in web design.

c. **Syntax**:



```
filter: drop-shadow(offset-x offset-y blur-radius spread-ra
dius color);
```

d. **Parameters** ⇒

i. *offest-x* → Sets the horizontal offset of the shadow. Positive values move to the right; negative values move to the left.

ii. *offset-y* → Sets the vertical offset of the shadow. Positive values move it down; negative move it up.

iii. *blur-radius* → (Optional parameter). Sets the blur radius of the shadow, creating a softer edge.

iv. *Spread-radius* →(Optional parameter). Sets the spread radius of the shadow, expanding or contracting its size.

v. *color* → Optional parameter. Sets the color of the shadow. Default is **black**.

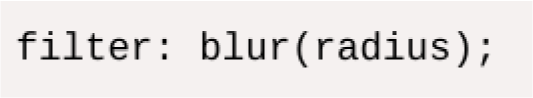
17. **blur():**

a. The CSS *blur()* function applies a Gaussian blur effect to an element, making it appear out of focus.

b. It is used with the filter property and accepts a length value defining the blur radius.

c. CSS *blur()* function is part of the CSS *filter* property, which allows you to apply graphical effects like blurring, sharpening, or color shifting to elements.

d. **Syntax**:



```
filter: blur(radius);
```

e. "*radius*" specifies the blur radius. The higher the value, the more blurred the element becomes.

f. It can be specified in various units like *pixels (px)* or *percentage (%)*.