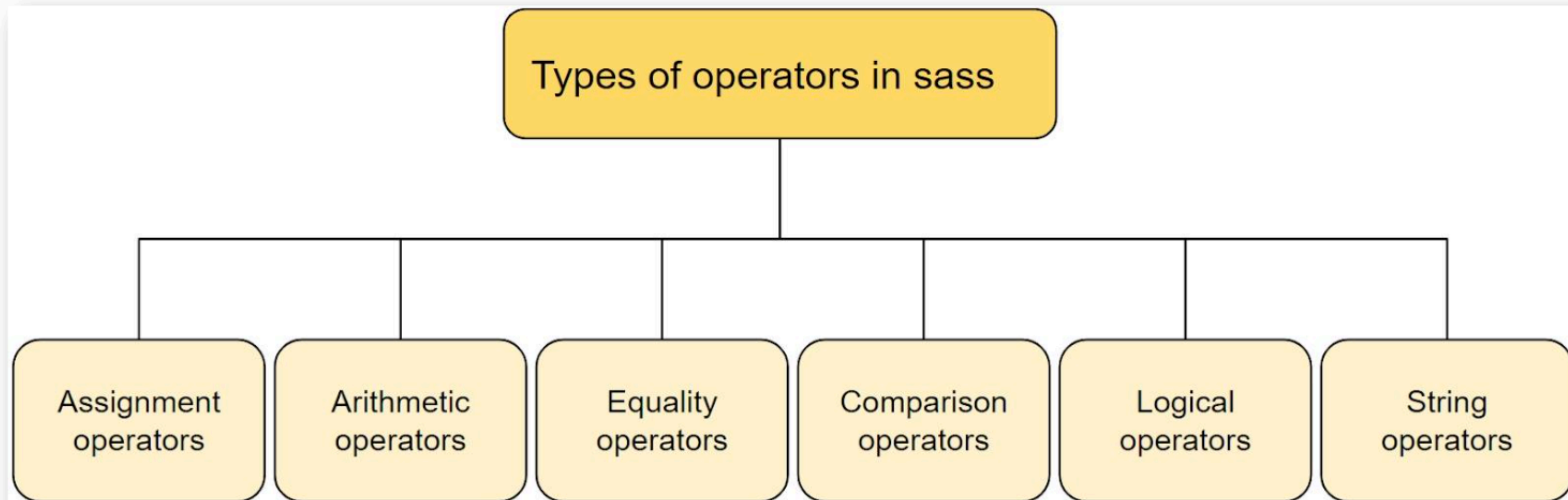# Operators in Sass

Operators are useful tools that enable you to perform calculations and manipulate values in your stylesheets. They simplify the process of developing styles and allow you to create dynamic styles based on various factors.

Types of operators in sass

| Assignment operators | Arithmetic operators | Equality operators | Comparison operators | Logical operators | String operators |

Operators in Sass – Arithmetic Operators

- Sass supports common arithmetic operators for numeric values:

  - **Addition (+):** Combine values to create new ones.

  - **Subtraction (-):** Calculate the difference between values.

  - **Multiplication (*):** Scale values by multiplying them.

  - **Division (/):** Divide values to create proportional changes.

  - **Modulo (%):** Get the remainder of a division operation.

Example:

```scss
$base-width: 100px;

.element {
    width: $base-width * 2;
    // Results in 200px
    margin-left: $base-width % 3;
    // Results in the remainder of  division
}
```

edureka!

Operators in Sass – Comparison Operators

- These operators compare values and return a boolean result:

  - **Equal to (==):** Check if two values are equal.

  - **Not equal to (!=):** Determine if two values are not equal.

  - **Greater than (>) and Less than (<):** Compare values' magnitudes.

Example:

```
$font-size: 14px;
.text {
    font-size: $font-size;
    color: if($font-size > 16px, ■red, ■blue);
    // Conditionally set color
}
```

edureka!

Operators in Sass – Logical Operators

- These operators are used to combine

  conditions and create complex expressions:

  - **AND (&&):** Both conditions must be true

    for the expression to be true.

  - **OR (||):** At least one condition must be true

    for the expression to be true.

  - **NOT (!):** Negates the truth value of a

    condition.

Example:

```scss
$is-bold: true;
$is-italic: false;


.styled-text {
    font-weight: if($is-bold and $is-italic, bold, normal);
}
```

edureka!

SASS Superpowers

edureka!

Nesting in Sass

- Nesting is used to nest the child CSS selectors inside the parent selector. In Sass, you can nest CSS rules with each other.

- The sass example shows that the **p-1**, **p-2**, **p-3**, and **p-4** selectors are nested inside the main div selector.

```scss
html > 𝓢 s.scss > ...
1   // define variables for the primary colors
2   $color_1: ☐ black;
3   $color_2: 🟥 rgb(221, 36, 36);
4   $color_3: 🟩 rgb(45, 192, 31);
5   $color_4: ☐ blanchedalmond;
6   #main_div{
7       #p-1{
8           background-color: $color_1;
9       }
10      #p-2{
11          background-color: $color_2;
12      }
13      #p-3{
14          background-color: $color_3;
15      }
16      #p-4{
17          background-color: $color_4;
18          font-size: 10px + 20px;
19      }
20  }
21
```

With Sass

```css
html > 𝟛 s.css > ⬡ #main_div #p-4
1   #main_div #p-1 {
2       background-color: ☐ black;
3   }
4
5   #main_div #p-2 {
6       background-color: 🟥 #dd2424;
7   }
8
9   #main_div #p-3 {
10      background-color: 🟩 #2dc01f;
11  }
12
13  #main_div #p-4 {
14      background-color: ☐ blanchedalmond;
15      font-size: 30px;
16  }
17  /*# sourceMappingURL=s.css.map */
```

Without Sass

edureka!