# :root-declaratie

### Voorbeeld

### Te gebruiken bij:

-toepassen door middel van de CSS functie var().

### Eigenschappen:

-

### Info

#### overzicht

#### aantekening

De root-declaratie wordt meestal gedefinieerd in de CSS-bestanden van een website en bevat algemene stijlinstellingen die van toepassing zijn op de gehele pagina. Hier zijn enkele voorbeelden van veelvoorkomende CSS-regels die in de root-declaratie kunnen worden opgenomen:

css

:root {

--primary-color: #007bff; /\* Definieer een aangepaste CSS-variabele voor de primaire kleur \*/

font-family: Arial, sans-serif; /\* Stel het standaardlettertype in voor de hele pagina \*/

font-size: 16px; /\* Stel de standaardlettergrootte in voor de hele pagina \*/

background-color: #f1f1f1; /\* Stel de standaard achtergrondkleur in voor de hele pagina \*/

}

De :root-selector selecteert het root element van het document, in dit geval <html>. De stijlregels binnen de :root-selector worden toegepast op alle elementen op de pagina, tenzij ze worden overschreven door specifiekere stijlregels.

Deze regels stellen onder andere een aangepaste CSS-variabele in voor de primaire kleur (--primary-color), definiëren het standaardlettertype en de standaardlettergrootte voor de hele pagina, en stellen een standaardachtergrondkleur in.

Door stijlinstellingen in de root-declaratie te definiëren, kun je consistentie in stijl en vormgeving over de hele pagina behouden en de mogelijkheid hebben om de waarden op één centrale locatie te wijzigen.

#### Gebruik

1. Maak een CSS-bestand: Maak een apart CSS-bestand waarin je je stijlregels zult definiëren. Je kunt dit bestand bijvoorbeeld "styles.css" noemen.
2. Selecteer het root element: Gebruik de :root-selector in je CSS-bestand om het root element (meestal <html>) te selecteren waarop je de stijlen wilt toepassen.
3. Definieer stijlinstellingen: Binnen de :root-selector kun je verschillende stijlinstellingen definiëren, zoals kleuren, lettertypen, groottes, marges, enzovoort. Deze stijlinstellingen worden dan toegepast op alle elementen op de pagina.

Hier is een voorbeeld van hoe je de root-declaratie kunt gebruiken in je CSS-bestand:

css

:root {

--primary-color: #007bff; /\* Definieer een aangepaste CSS-variabele voor de primaire kleur \*/

font-family: Arial, sans-serif; /\* Stel het standaardlettertype in voor de hele pagina \*/

font-size: 16px; /\* Stel de standaardlettergrootte in voor de hele pagina \*/

background-color: #f1f1f1; /\* Stel de standaardachtergrondkleur in voor de hele pagina \*/

}

/\* Voorbeeld van het toepassen van de stijlen \*/

body {

background-color: var(--primary-color); /\* Gebruik de aangepaste CSS-variabele \*/

color: white;

}

In dit voorbeeld definieert de root-declaratie de aangepaste CSS-variabele --primary-color, het standaardlettertype, de standaardlettergrootte en de standaardachtergrondkleur voor de hele pagina. Vervolgens kun je deze stijlen toepassen op specifieke elementen zoals body, waarbij je de aangepaste variabele gebruikt.

Door de root-declaratie te gebruiken, kun je algemene stijlinstellingen centraliseren en eenvoudig wijzigingen aanbrengen die van invloed zijn op de hele pagina.

#### Root kleur aanpassen:

div {

border-color: var(--main-bg-color);

}

Wil je jouw kleur nu aanpassen? Dan hoef je dat alleen maar bij de declaratie van de variabele te doen in :root. Handig hé?

css

:root {

background-color: #f1f1f1; /\* Hexadecimale waarde \*/

}

css

:root {

background-color: red; /\* Kleurnaam \*/

}

css

:root {

background-color: rgb(255, 0, 0); /\* RGB-waarde \*/

}

#### calc

Stel, je wil dat jouw input-element dezelfde breedte heeft als zijn container, minus de breedte van het label. Je weet dat het label altijd 30 pixels breed is, maar de breedte van de container zal variëren op basis van schermgrootte. Dus hoe groot moet het input-element dan zijn... ? In plaats van ons hoofd hierover te breken, lost de calc()-functie dit voor je op:

input {

width: calc(100% - 30px);

}

div {

width: calc(100% / 3);

}

.column {

height: calc(20px \* 5);

}

.row {

margin: calc(var(--default-margin) + 20px);

}