

# Eisen

Per CH onderdeel kun je 0 of de aangegeven punten krijgen.

ID	Gedaan	Omschrijving	Prio	Punten	Competentie VT
CH00	✓	Minimaal vier van onderstaande checks <b>moeten</b> zijn geïmplementeerd	Must	0	APP-2, APP-6, APP-7,
CH01	✓	Controleer of er geen variabelen worden gebruikt die niet gedefinieerd zijn.	Should	5	
CH02	✓	Controleer of de operanden van de operaties plus en min van gelijk type zijn. Je mag geen pixels bij percentages optellen bijvoorbeeld. Controleer dat bij vermenigvuldigen minimaal een operand een scalaire waarde is. Zo mag 20% * 3 en 4 * 5 wel, maar mag 2px * 3px niet.	Should	5	
CH03	✓	Controleer of er geen kleuren worden gebruikt in operaties (plus, min en keer).	Should	5	
CH04	✓	Controleer of bij declaraties het type van de value klopt met de property. Declaraties zoals width: #ff0000 of color: 12px zijn natuurlijk onzin.	Should	5	
CH05	✓	Controleer of de conditie bij een if-statement van het type boolean is (zowel bij een variabele-referentie als een boolean literal)	Should	5	
CH06	✓	Controleer of variabelen enkel binnen hun scope gebruikt worden	Must	5	

Eigen uitbreidingen (20 punten)

ID	Gedaan	Omschrijving	Punten
1	✓	Implementeren van booleaanse expressies zoals 3<5, Value==5, !AdjustWidth (deze kunnen worden gebruikt in if statement) en &&	8
2	✓	Checken van booleaanse expressies. Bij > dat de waardes van hetzelfde type zijn en alleen 20% 20px 22. Hiernaast dat ! en && alleen op booleaanse waardes (of expressies) kan	4
3	✓	Line comments // en block comments /* */	0
4	✓	Controleer of er geen booleans worden gebruikt in de operaties (plus, min en keer).	2

ID	Gedaan	Omschrijving	Punten
5	✓	( operaties ) geeft voorrang	3
6	✓	2^2 machten toevoegen. Machten kunnen alleen SCALAIR zijn en volgen regel CH02	5

## Test voorbeelden

### ##CH01 en CH06 Input

```
LinkColor := #ff0000;
MeaningOfLife := MeaningOfLife;

p {
  ParWidth := 20px;
  background-color: LinkColor;
  width: ParWidth;
}

#menu {
  width: ParWidth;
}
```

### Output

```
ERROR: Variable has not been defined
ERROR: Variable has not been defined
```

### ##CH02 Input

```
Plusslagen := 20px + 20px;
Minfalen := 20px + 10%;
Keerhalen := 20% * 2;
Keerfalen := 10px * 10px;
```

### Output

```
ERROR: Add operation values must be off same type
ERROR: One of the values of a multiply operation must be scalar
```

### ##CH03 Input

```
Plusmetkleuren := #808080 + #808080;
```

Output

```
ERROR: Don't use colours in operations
```

##CH04 Input

```
LinkColor := #ff0000;

p {
  background-color: LinkColor;
  width: LinkColor; // Error
}

#menu {
  height: 50 * 20; // Error
  width: 2 * 30%;
  color: 10px; // Error
}
```

Output

```
ERROR: size declarations must be assigned percentage or pixel values
ERROR: size declarations must be assigned percentage or pixel values
ERROR: color declarations must be assigned hexadecimal values
```

## CH05

Input

```
Boolean := TRUE;
NotBoolean := 10px;

.777 {
  if [TRUE] {
    width:50%;
  }
  if [NotBoolean]{
    height:10px;
  }
  if[Boolean] {
```

```
        color: #ffff22;
    }
}
```

Output

ERROR: If statement condition is not boolean

## Eigen 1

Start 14:53 pauze bij 15:59 Nog later ook 1.5 uur. Volgens mij. Input

```
Precedence := 22 < 20 && 3 == 2; // FALSE
p {
    if[Precedence] {
        width: 100px;
    }
    if[!FALSE] {
        height: 200px;
    } else {
        height: 300px;
    }
}
Scalar := 2;
Correct := 1 < Scalar; // True
Colorref := #ffffff;
Correctt := #ffffff == Colorref; // True

#bovenaan {
    if[Correct] {
        if[Correctt] {
            background-color: Colorref;
        } else {
            color: #000000;
        }
    } else {
        color: #001100;
    }
}
```

Output

```
p {
    height: 200px;
}
```

```
#bovenaan {  
  background-color: #ffffff;  
}
```

## Eigen 2

35 min Input

```
Precedence := 22 < 20 && 3 == 2px; // FALSE  
  
Scalar := 2px;  
Correct := 1 < Scalar; // Error  
  
AND := 33 && 33;  
  
NOT := !8;
```

Output

```
ERROR: The values of the comparison operation must be of the same type  
ERROR: The values of a boolean comparison must be boolean.  
ERROR: Only boolean values can be used with Not Operator
```

## Eigen 3

Start 11:07 tot 11:30

Input

```
p {  
  //width: 10px;  
  width: 20px;  
}  
  
/*p {  
  color: 222222222222px;  
} */
```

Output

```
p {  
  width: 20px;  
}
```

## Eigen 4

Input

```
Keermetbool := TRUE + FALSE;

Bool := TRUE;

Test2 := Bool * Bool;

Test3 := 2 * 2 * FALSE * FALSE;
```

Output

```
ERROR: Don't use booleans in operations
ERROR: Don't use booleans in operations
ERROR: Don't use booleans in operations
ERROR: Don't use booleans in operations
```

## Eigen 5

11:43 start tot 12:15 Input

```
div {
  width: 50px + 2 * 10px - 2px; // dit is dus 68px
  height: 50px + 2 * (10px - 2px); // dit is dus 66px
}
```

Output

```
div {
  width: 68px;
  height: 66px;
}
```

## Eigen 6

14:01 start tot 14:46

Scalaire waarde exponent. Input

```
p {  
  width: 20 ^ 2 * 20px;  
}
```

Output

```
p {  
  width: 8000px;  
}
```

Check only scalar values Input

```
p {  
  width: 20 ^ 2px;  
  height: 2px ^ 3;  
}
```

Output

```
ERROR: Exponentiation operations can only be done on scalar values and/or use  
scalar values  
ERROR: Exponentiation operations can only be done on scalar values and/or use  
scalar values
```

## Los

Ben toch niet helemaal blij dat dit niet werd geparsed. Dus opgelost. Input

```
#223344 {  
  color: #223344;  
}
```

Output

```
#223344 {  
  color: #223344;  
}
```