

Arduboy

Starters Guide

Stijn Caerts



JCW

Jeugd, Cultuur en Wetenschap vzw

Copyright © 2019 Stijn Caerts

JEUGD, CULTUUR EN WETENSCHAP VZW

STIJN.CAERTS.BE – WWW.JEUGDCULTUURENWETENSCHAP.BE

Dit werk valt onder een Creative Commons Naamsvermelding-NietCommercieel-GelijkDelen 4.0 Internationaal-licentie (de “Licentie”). Dit document mag enkel gebruikt worden in navolging van de Licentie. De volledige Licentie-tekst is beschikbaar op <https://creativecommons.org/licenses/by-nc-sa/4.0/>.



Eerste versie, augustus 2019

Inhoudsopgave

I	Introductie tot C++	
1	Inleiding	9
1.1	Wat is C++?	9
1.2	Syntax	9
1.2.1	Punktkomma	9
1.2.2	Accolades	9
1.2.3	Commentaar	10
2	Variabelen en types	11
2.1	Datatypes	11
2.2	Variabelen	11
2.2.1	Declareren	11
2.2.2	Scope	12
2.2.3	Operatoren	13
3	Controlestructuren	15
3.1	If-then-else	15
3.1.1	If-then	15
3.1.2	Else	15
3.1.3	Geneste if-statements	16
3.2	Switch	16
3.3	While-lus	16
3.4	For-lus	16

4	Funcities en procedures	17
5	Arrays en lijsten (ADVANCED)	19
6	Klassen en objecten (ADVANCED)	21

II

Arduboy

7	Arduino	25
7.1	Programmastructuur	25
7.1.1	Globale variabelen	25
7.1.2	De setup() procedure	25
7.1.3	De loop() procedure	25
7.2	Importeren van libraries	25
8	Arduboy	27
8.1	Instellingen	27
8.2	De Arduboy2 library	27
8.2.1	Display	27
8.2.2	Buttons	27
8.3	Emulator	27
8.4	Programma op Arduboy plaatsen	27

III

Part One

9	Text Chapter	31
9.1	Paragraphs of Text	31
9.2	Citation	32
9.3	Lists	32
9.3.1	Numbered List	32
9.3.2	Bullet Points	32
9.3.3	Descriptions and Definitions	32
10	In-text Elements	33
10.1	Theorems	33
10.1.1	Several equations	33
10.1.2	Single Line	33
10.2	Definitions	33
10.3	Notations	34
10.4	Remarks	34
10.5	Corollaries	34
10.6	Propositions	34
10.6.1	Several equations	34

10.6.2	Single Line	34
10.7	Examples	34
10.7.1	Equation and Text	35
10.7.2	Paragraph of Text	35
10.8	Exercises	35
10.9	Problems	35
10.10	Vocabulary	35

IV

Part Two

11	Presenting Information	39
11.1	Table	39
11.2	Figure	39
	Bibliografie	41
	Artikels	41
	Boeken	41
	Index	43



Introductie tot C++

1	Inleiding	9
1.1	Wat is C++?	
1.2	Syntax	
2	Variabelen en types	11
2.1	Datatypes	
2.2	Variabelen	
3	Controlestructuren	15
3.1	If-then-else	
3.2	Switch	
3.3	While-lus	
3.4	For-lus	
4	Funcies en procedures	17
5	Arrays en lijsten (ADVANCED)	19
6	Klassen en objecten (ADVANCED)	21



1. Inleiding

1.1 Wat is C++?

Programma's voor **Arduino** en **Arduboy** worden geschreven in de programmeertaal C++. Het is niet nodig om de hele programmeertaal te kennen en begrijpen voor je aan de slag kan gaan met programmeren. Daarom geven we hier een beknopt overzicht van de belangrijkste concepten die je nodig hebt om van start te gaan.

In de volgende hoofdstukken komen variabelen en types, controlestructuren (if-then-else, for, while) en functies en procedures aan bod. Tot slot zijn er nog twee hoofdstukken die dieper ingaan op de mogelijkheden van C++, namelijk arrays en lijsten, en klassen en objecten.

Voor een interactieve en uitgebreidere introductie tot C++, kan je terecht bij W3Schools (<https://www.w3schools.com/cpp/>).

1.2 Syntax

1.2.1 Puntkomma

Achter elke instructie wordt in C++ een puntkomma ; geplaatst. Deze puntkomma vertelt de compiler dat op die plaats een instructie eindigt. Als je een puntkomma vergeet te plaatsen, is het programma niet correct en zal je het programma niet kunnen compileren. De compiler zal dan een foutmelding geven.

1.2.2 Accolades

Accolades {} worden gebruikt om een blok code aan te duiden. De accolades moeten gebalanceerd voorkomen in de code, wat betekent dat voor elke openende accolade { er een bijhorende sluitende accolade } moet zijn. Het gebruik van accolades wordt later geïllustreerd bij de controlestructuren (if-then-else, for, ..), functies en klassen.

1.2.3 Commentaar

Om de leesbaarheid van je code te verhogen, is het nuttig om commentaar toe te voegen. In deze commentaar beschrijf je wat dit deel van de code juist doet. Hierdoor is het duidelijk wat je juist hebt geprogrammeerd, ook als je later opnieuw je code bekijkt.

In C++ zijn er twee verschillende manieren om commentaar toe te voegen. Commentaar op één lijn wordt aangeduid met `//`. Alle tekst na `//` tot het einde van de lijn wordt beschouwd als commentaar en zal bijgevolg niet uitgevoerd worden.

■ Voorbeeld 1.1 — Commentaarlijn.

```
// Dit is een lijn commentaar  
int a = 42;
```

◀

Commentaar over meerdere lijnen start met `/*` en eindigt met `*/`. Alle tekst tussen `/*` en `*/` wordt door de compiler genegeerd.

■ Voorbeeld 1.2 — Commentaar over meerdere lijnen.

```
/*  
Deze commentaar neemt  
meerdere lijnen in beslag.  
*/  
int a = 42;
```

◀

2. Variabelen en types

Variabelen worden gebruikt om informatie of gegevens bij te houden in het programma. Een variabele is een plaats in het geheugen, met een zelfgekozen naam, waar gegevens in opgeslagen kunnen worden. In C++ heeft elke variabele een type. Een type komt overeen met het soort van gegevens dat opgeslagen kunnen worden in zo'n variabele.

2.1 Datatypes

Op basis van het datatype van een variabele wordt er geheugen gereserveerd om informatie in op te slaan. In C++ zijn de volgende primitieve datatypes gedefinieerd:

Definitie 2.1.1 — Primitieve datatypes.

bool	Booleaanse waarde: true of false
char	Karakter
int	Geheel getal
float	Kommagetal (met enkele precisie)
double	Kommagetal (met dubbele precisie)

Deze types worden gebruikt bij variabelen, maar ook bij functies. Deze primitieve datatypes kunnen ook gebruikt worden om complexere types samen te stellen, zoals bijvoorbeeld bij arrays en klassen.

2.2 Variabelen

Een variabele kan gezien worden als een opslagplaats waaraan een naam verbonden is. In deze opslagplaats kan een waarde geplaatst worden.

2.2.1 Declareren

Alvorens informatie kan opgeslagen worden in een variabele, moet de variabele eerst *gedefinieerd* of *gedeclareerd* worden. De declaratie van een variabele vertelt de compiler dat er plaats

in het geheugen gereserveerd moet worden om gegevens van het gegeven datatype in op te slaan.

Een nieuwe variabele declareren doe je met een instructie van deze vorm:

```
type naam_variabele;
```

■ **Voorbeeld 2.1 — Declareren van een *integer* variabele.**

```
// Declareer een nieuwe variabele van het type int met naam a
int a;
```

Je kan bij het definiëren van een nieuwe variabele er ook meteen een waarde aan toekennen.

■ **Voorbeeld 2.2 — Declareren van een *integer* variabele en een waarde toekennen.**

```
/*
  Declareer een nieuwe variabele van het type int met naam a,
  en ken de waarde 42 toe
*/
int a = 42;
```

Bij het kiezen van een variabelenaam moet je met de volgende regels rekening houden:

- namen moeten beginnen met een letter of een laag streepje (`_`);
- namen kunnen letters, cijfers en lage streepjes bevatten;
- namen zijn hoofdlettergevoelig;
- spaties of speciale karakters zijn niet toegelaten;
- C++-keywords (zoals `int`, `if`, ...) kunnen niet gebruikt worden als variabelenaam.

eld van elk type
de

2.2.2 Scope

De *scope* van een variabele is de plaats in het programma waar je een gedefinieerde variabele kan gebruiken. Je kan dit ook beschouwen als de levensduur van de variabele.

De scope van een variabele wordt bepaald door waar de variabele werd gedeclareerd. Dit kunnen we opdelen in 3 verschillende soorten plaatsen.

1. in een functie of blok: **lokale variabelen**,
2. in de definitie van een functie: **parameters**,
3. buiten alle functies: **globale variabelen**.

In het algemeen kan gesteld worden dat de scope van de variabelen beperkt is tot het blok, aangeduid met accolades `{}`, waarin de variabele gedefinieerd is.

■ **Voorbeeld 2.3 — Scope van variabelen.**

```
// Globale variabele g
int g = 12;

void setup() {
  // Lokale variabelen a en b
  int a = 42;
```

```

/*
  Globale variabelen kunnen na declaratie
  overal gebruikt worden
*/
int b = a + g;

// Na declaratie moet het type van variabelen niet gespecificeerd worden
g = b;
}

// Variabelen a en b kunnen buiten de functie setup() niet gebruikt worden

```

2.2.3 Operatoren

Om de waarde in de variabelen aan te passen, maken we gebruik van *operatoren*. Hieronder geven we een overzicht van de belangrijkste operatoren in C++.

Rekenkundige operatoren

Rekenkundige operatoren worden gebruikt om veelgebruikte wiskundige operaties uit te voeren. In deze voorbeelden gebruiken we de variabelen $A = 10$ en $B = 3$.

Operator	Beschrijving	Voorbeeld
+	Optelling	$A + B \rightarrow 13$
-	Vermindering	$A - B \rightarrow 7$
*	Vermenigvuldiging	$A * B \rightarrow 30$
/	Deling	$A / B \rightarrow 2$
%	Modulus (rest na deling)	$A \% B \rightarrow 1$
++	Verhoogt de waarde van de variabele met 1	$A++ \Rightarrow A \rightarrow 11$
--	Verlaagt de waarde van de variabele met 1	$A-- \Rightarrow A \rightarrow 9$

Toewijzende operatoren

De operator voor de toewijzing van een waarde aan een variabele is het gelijkheidsteken (=). Naast deze toewijzingsoperator bestaan er nog andere varianten, die kortere notaties zijn voor een rekenkundige operatie gevolgd door een toewijzing.

Operator	Voorbeeld	Equivalent
=	$A = 12$	$A = 12$
+=	$A += 2$	$A = A + 2$
-=	$A -= 3$	$A = A - 3$
*=	$A *= 4$	$A = A * 4$

Vergelijkende operatoren

Vergelijkende operatoren worden gebruikt om twee waardes met elkaar te vergelijken. Het resultaat van deze operatie is een Booleaanse waarde, true of false.

Operator	Beschrijving	Voorbeeld
<code>==</code>	Gelijk aan	<code>10 == 3 → false</code>
<code>!=</code>	Niet gelijk aan	<code>10 != 3 → true</code>
<code>></code>	Groter dan	<code>10 > 10 → false</code>
<code><</code>	Kleiner dan	<code>5 < 10 → true</code>
<code>>=</code>	Groter dan of gelijk aan	<code>10 >= 10 → true</code>
<code><=</code>	Kleiner dan of gelijk aan	<code>5 <= 10 → true</code>

Logische operatoren

Logische operatoren worden gebruikt om meerdere Booleaanse waarden te combineren in een formule tot één waarheidswaarde.

Operator	Beschrijving	Voorbeeld
<code>&&</code>	Logische conjunctie (AND)	<code>(true && false) → false</code>
<code> </code>	Logische disjunctie (OR)	<code>(true false) → true</code>
<code>!</code>	Logische negatie (NOT)	<code>! true → false</code>

Voor meer voorbeelden van het gebruik van logische operatoren, kan je een kijkje nemen naar waarheidstabellen [3].

3. Controlestructuren

Controlestructuren bepalen de loop van het programma. Zo is het mogelijk om stukken code enkel in specifieke gevallen uit te voeren, of om een blok code meerdere keren te herhalen.

3.1 If-then-else

De eenvoudigste controlestructuur is het *if*-statement. Dit kan verder uitgebreid worden met een *else*-blok voor verdere controle over de uitvoeringsvolgorde.

3.1.1 If-then

Definitie 3.1.1 — If-statement.

```
if (conditie) {  
    /*  
        instructies tussen deze accolades worden enkel  
        uitgevoerd als de conditie tot true evalueert  
    */  
}
```

3.1.2 Else

Na een if-statement kan optioneel een else-blok geplaatst worden. De instructies in dit blok zullen enkel uitgevoerd worden als aan de conditie van het if-statement **niet** voldaan is. De code ziet er dan uit als volgt.

Definitie 3.1.2 — If-then-else.

```
if (conditie) {  
    // conditie evalueert tot true  
} else {  
    // conditie evalueert tot false  
}
```

- 3.1.3** Geneste if-statements
- 3.2** Switch
- 3.3** While-lus
- 3.4** For-lus



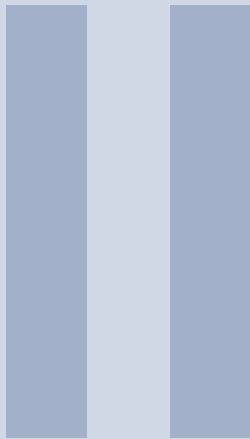
4. Funcies en procedures



5. Arrays en lijsten (ADVANCED)



6. Klassen en objecten (ADVANCED)



Arduboy

7	Arduino	25
7.1	Programmastructuur	
7.2	Importeren van libraries	
8	Arduboy	27
8.1	Instellingen	
8.2	De Arduboy2 library	
8.3	Emulator	
8.4	Programma op Arduboy plaatsen	



7. Arduino

7.1 Programmastructuur

7.1.1 Globale variabelen

7.1.2 De `setup()` procedure

7.1.3 De `loop()` procedure

7.2 Importeren van libraries



8. Arduboy

8.1 Instellingen

8.2 De Arduboy2 library

8.2.1 Display

8.2.2 Buttons

8.3 Emulator

8.4 Programma op Arduboy plaatsen



Part One

9	Text Chapter	31
9.1	Paragraphs of Text	
9.2	Citation	
9.3	Lists	
10	In-text Elements	33
10.1	Theorems	
10.2	Definitions	
10.3	Notations	
10.4	Remarks	
10.5	Corollaries	
10.6	Propositions	
10.7	Examples	
10.8	Exercises	
10.9	Problems	
10.10	Vocabulary	

9. Text Chapter

9.1 Paragraphs of Text

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi.

Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

9.2 Citation

This statement requires citation [1]; this one is more specific [2, pagina 162].

9.3 Lists

Lists are useful to present information in a concise and/or ordered way¹.

9.3.1 Numbered List

1. The first item
2. The second item
3. The third item

9.3.2 Bullet Points

- The first item
- The second item
- The third item

9.3.3 Descriptions and Definitions

Name Description

Word Definition

Comment Elaboration

¹Footnote example...

10. In-text Elements

10.1 Theorems

This is an example of theorems.

10.1.1 Several equations

This is a theorem consisting of several equations.

Theorem 10.1.1 — Name of the theorem. In $E = \mathbb{R}^n$ all norms are equivalent. It has the properties:

$$||\mathbf{x}|| - ||\mathbf{y}|| \leq ||\mathbf{x} - \mathbf{y}|| \quad (10.1)$$

$$||\sum_{i=1}^n \mathbf{x}_i|| \leq \sum_{i=1}^n ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \quad (10.2)$$

10.1.2 Single Line

This is a theorem consisting of just one line.

Theorem 10.1.2 A set $\mathcal{D}(G)$ is dense in $L^2(G)$, $|\cdot|_0$.

10.2 Definitions

This is an example of a definition. A definition could be mathematical or it could define a concept.

Definition 10.2.1 — Definition name. Given a vector space E , a norm on E is an application,

denoted $\|\cdot\|$, E in $\mathbb{R}^+ = [0, +\infty[$ such that:

$$\|\mathbf{x}\| = 0 \Rightarrow \mathbf{x} = \mathbf{0} \quad (10.3)$$

$$\|\lambda \mathbf{x}\| = |\lambda| \cdot \|\mathbf{x}\| \quad (10.4)$$

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \quad (10.5)$$

10.3 Notations

Notation 10.1. Given an open subset G of \mathbb{R}^n , the set of functions φ are:

1. Bounded support G ;
2. Infinitely differentiable;

a vector space is denoted by $\mathcal{D}(G)$.

10.4 Remarks

This is an example of a remark.



The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

10.5 Corollaries

This is an example of a corollary.

Gevolg 10.5.1 — Corollary name. The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

10.6 Propositions

This is an example of propositions.

10.6.1 Several equations

Propositie 10.6.1 — Proposition name. It has the properties:

$$\left| \|\mathbf{x}\| - \|\mathbf{y}\| \right| \leq \|\mathbf{x} - \mathbf{y}\| \quad (10.6)$$

$$\left\| \sum_{i=1}^n \mathbf{x}_i \right\| \leq \sum_{i=1}^n \|\mathbf{x}_i\| \quad \text{where } n \text{ is a finite integer} \quad (10.7)$$

10.6.2 Single Line

Propositie 10.6.2 Let $f, g \in L^2(G)$; if $\forall \varphi \in \mathcal{D}(G)$, $(f, \varphi)_0 = (g, \varphi)_0$ then $f = g$.

10.7 Examples

This is an example of examples.

10.7.1 Equation and Text

■ **Voorbeeld 10.1** Let $G = \{x \in \mathbb{R}^2 : |x| < 3\}$ and denoted by: $x^0 = (1, 1)$; consider the function:

$$f(x) = \begin{cases} e^{|x|} & \text{si } |x - x^0| \leq 1/2 \\ 0 & \text{si } |x - x^0| > 1/2 \end{cases} \quad (10.8)$$

The function f has bounded support, we can take $A = \{x \in \mathbb{R}^2 : |x - x^0| \leq 1/2 + \epsilon\}$ for all $\epsilon \in]0; 5/2 - \sqrt{2}[$. ◀

10.7.2 Paragraph of Text

■ **Voorbeeld 10.2 — Example name.** Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. ◀

10.8 Exercises

This is an example of an exercise.

■ **Oefening 10.1** This is a good place to ask a question to test learning progress or further cement ideas into students' minds. ◀

10.9 Problems

■ **Probleem 10.1** What is the average airspeed velocity of an unladen swallow?

10.10 Vocabulary

Define a word to improve a students' vocabulary.

■ **Vocabulaire 10.1 — Word.** Definition of word.

IV

Part Two

11	Presenting Information	39
11.1	Table	
11.2	Figure	
	Bibliografie	41
	Artikels	
	Boeken	
	Index	43

11. Presenting Information

11.1 Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Tabel 11.1: Table caption

Referencing Table 11.1 in-text automatically.

11.2 Figure



Figuur 11.1: Figure caption

Referencing Figure 11.1 in-text automatically.



Bibliografie

Artikels

- [1] James Smith. "Article title". In: 14.6 (mrt 2013), pagina's 1–8 (zie pagina 32).

Boeken

- [2] John Smith. *Book title*. 1ste editie. Deel 3. 2. City: Publisher, jan 2012, pagina's 123–200 (zie pagina 32).

Websites

- [3] Wikipedia. *Waarheidstabel*. <https://nl.wikipedia.org/wiki/Waarheidstabel> (zie pagina 14).

Index

B

Booleaanse waarde . . . *zie ook* Datatype, bool

C

C++ 9
Citation 32
Commentaar 10
Corollaries 34

D

Datatype 11
 bool 11
 char 11
 double 11
 float 11
 int 11
Definitions 33

E

Examples 34
 Equation and Text 35
 Paragraph of Text 35
Exercises 35

F

Figure 39

L

Lists 32
 Bullet Points 32
 Descriptions and Definitions 32
 Numbered List 32

N

Notations 34

O

Operatoren 13
 Logisch 14
 Rekenkundig 13
 Toewijzend 13
 Vergelijkend 13

P

Paragraphs of Text 31
Problems 35
Propositions 34
 Several Equations 34
 Single Line 34

R

Remarks 34

T

Table	39
Theorems	33
Several Equations.....	33
Single Line	33
Type	<i>zie ook</i> Datatype

V

Variabele	11
Declareren	11
Scope	12
Vocabulary	35