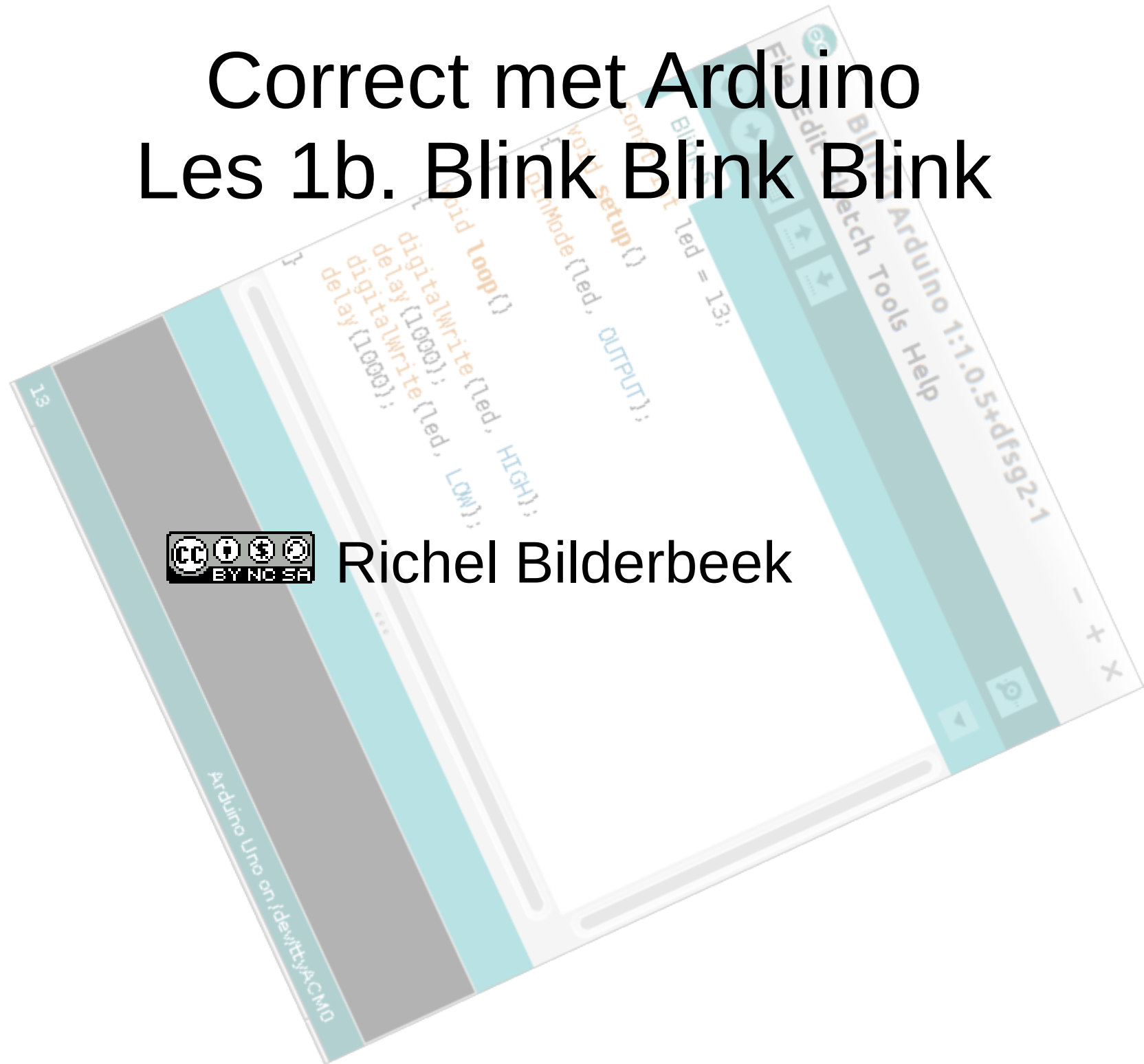


Correct met Arduino

Les 1b. Blink Blink Blink



Richel Bilderbeek

Les 1b: Blink Blink Blink

```
int rodePin = 10;
int groenePin = 11;
void setup() {
    pinMode(rodePin, OUTPUT);
    pinMode(groenePin, OUTPUT);
}
void loop() {
    digitalWrite(rodePin, HIGH);
    digitalWrite(groenePin, HIGH);
    delay(1000);
    digitalWrite(rodePin, LOW);
    digitalWrite(groenePin, LOW);
    delay(1000);
}
```

Variabele naam

- De variabele namen zijn in camelCase: 'rodePin'
- Veel gebruikt alternatief: 'rode_pin'

Coding standaard

- Kies een coding standaard en volg deze
 - Bijvoorbeeld de 'C++ Core Guidelines',
<https://github.com/isocpp/CppCoreGuidelines>
- Meng niet 'rodeLed' en 'groene_led'

Programmeren

- Druk je ideeën direct/juist/zonder omwegen uit in code [1]
- Als je iets niet zegt, zeg je ook iets

Const

- `'int rodePin = 11;'` betekent dat `rodePin` veranderd mag/zal worden
- In het echt ziet dat er raar uit!
- `'const int rodePin = 11;'` betekent het pinnummer naar de LED 11 is en blijft
- Gebruik `const` waar mogelijk [1-4]

[1] C++ Core Guidelines. ES.25: Declare an objects `const` or `constexpr` unless you want to modify its value later on

[2] Bjarne Stroustrup. The C++ Programming Language (3rd edition). 7.9.3: 'Use `const` extensively and consistently'.

[3] Scott Meyers. Effective C++ (3rd edition). Item 3: 'Use `const` whenever possible'.

[4] Herb Sutter, Andrei Alexandrescu. C++ coding standards: 101 rules, guidelines, and best practices. Item 15: 'Use `const` proactively'.

Les 1b: Blink Blink Blink

```
const int rodePin = 10;
const int groenePin = 11;
void setup() {
    pinMode(rodePin, OUTPUT);
    pinMode(groenePin, OUTPUT);
}
void loop() {
    digitalWrite(rodePin, HIGH);
    digitalWrite(groenePin, HIGH);
    delay(1000);
    digitalWrite(rodePin, LOW);
    digitalWrite(groenePin, LOW);
    delay(1000);
}
```

Globale variabelen, 'globals'

- De code gebruikt twee globale variabelen
- Probeer het gebruik van globale variabelen te voorkomen, geef de voorkeur aan lokale variabelen [1-6]
- In dit geval minder erg, omdat de globals constanten zijn

[1] Andrei Alexandrescu. Modern C++ Design

[2] Stephen C. Dewhurst. C++ Gotchas

[3] Bjarne Stroustrup. The C++ Programming Language (3rd edition)

[4] Andrei Alexandrescu. Modern C++ Design

[5] Joint Strike Fighter Air Vehicle C++ Coding Standards

[6] etcetera

Les 1b: Blink Blink Blink

```
void setup() {  
    const int rodePin = 10;  
    const int groenePin = 11;  
    pinMode(rodePin, OUTPUT);  
    pinMode(groenePin, OUTPUT);  
    while (1) {  
        digitalWrite(rodePin, HIGH);  
        digitalWrite(groenePin, HIGH);  
        delay(1000);  
        digitalWrite(rodePin, LOW);  
        digitalWrite(groenePin, LOW);  
        delay(1000);  
    }  
}  
  
void loop() {}
```

Waarom zijn globals slecht?

- Het overzicht vermindert
 - Je kunt ze overal veranderen
 - Je kunt ze overal lezen
- Globals moeten een unieke naam hebben
 - Wat als je code samen wilt voegen met iemand die als 'jouw' variabelenaam gebruikt?
- Ze schalen niet mee als je code groeit
- Mythe: globals zijn sneller