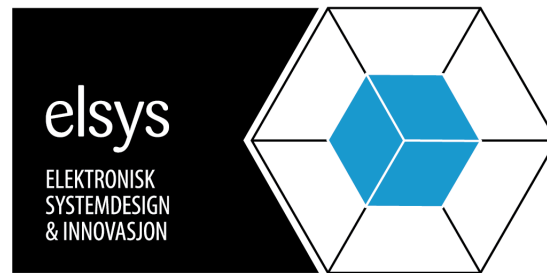




NTNU

Kunnskap for en bedre verden

 NTNU



FAKULTET FOR ELEKTRONISKE SYSTEMER

TTT4255 ELEKTRONISK SYSTEMDESIGN GRUNNKURS 2020

---

# Sammenligning av Arduinokode og python

---

---

## 1 Introduksjon

Dette dokumentet tar for seg en sammenlikning av to programmeringsspråk, wiring og python. Wiring er programmeringen som skrives i Arduino IDE. I slutten av dokumentet er det lagt ved ressurser for å finne formler, oppbygning og syntaks til wiringprogrammering i Arduino IDE.

## 2 Generell sammenlikning av wiring og python

Hovedforskjellen mellom wiring og python er at wiring er et strukturorientert programmeringsspråk, mens python er et objektorientert programmeringsspråk. Generelt er wiring brukt for å utvikle maskindrevne applikasjoner, mens python er et meningsbasert programmeringsspråk. Wiring er et språk basert på C og C++. C og C++ har få innebygde funksjoner, mens python har fult dannede innebygde og forhåndsdefinerte funksjoner. I tillegg bruker python en innrykkssyntaks, mens wiring krever eksplisitt implementasjon av funksjoner. Syntaksen til python er lettere å lese og forstå, mens wiring krever dypere forståelse for å programmere og implementere.

I tabell 1 vises en sammenlikning av syntaksen til begge kodespråkene.

Funksjon	Python	Wiring
Skrive ut	<code>print("hei")</code>	<code>Serial.print("hei");</code>
Definere tallvariabel	<code>tall = 10</code>	<code>int tall = 10;</code>
Definere setning	<code>setning = "Hei på deg"</code>	<code>string setning = "Hei på deg";</code>
if-løkke	<pre>a = 10 b = 20 if a == b:     print('yes') else:     print('no')</pre>	<pre>int a = 10; int b = 20;  if (a==b){     serial.print("yes"); } else{     serial.print("no"); }</pre>
for løkke	<pre>for i in range(5):     print(i)</pre>	<pre>for (i=0; i&lt;5, i++){     serial.print(i); }</pre>
Boolsk operator: og	AND	&&
Boolsk operator: eller	OR	
Boolsk operator: ikke	NOT	!=

Table 1: Sammenlikning av python og wiring.

Flere løkker med eksempelkode: [https://www.tutorialspoint.com/arduino/arduino\\_loops.htm](https://www.tutorialspoint.com/arduino/arduino_loops.htm)

---

## 3 Generell struktur

Strukturen og syntaksen til wiring er mer komplekst enn til python. I wiringprogrammering trenger man *setup()*, *loop()*, *serial.begin()* og som oftest *delay()*.

**setup():** Når koden kjøres kalles *setup()* funksjonen på først. Den initialiserer variabler, pinmoduser og bruker bibliotek. Denne funksjonen kjører kun en gang.

**loop():** Etter *setup()* kjøres *loop()*. Den kjører om igjen og om igjen konsekvent, og lar programmet forandre seg og respondere. Denne funksjonen bruker man for å kontrollere arduinoen.

**serial.begin():** *Serial.begin(value)* angir datahastigheten (*value*) i bits per sekund (baud) for seriell dataoverføring. For å kommunisere med Serial Monitor, må du sørge for å bruke en av baudhastighetene (*value*) som er oppført i menyen nederst til høyre på skjermen. Denne setter man inni til *setup()* og skriver vanligvis 9600 som *value*.

Eksempel i figur 1.

### Example Code

```
void setup() {  
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps  
}  
  
void loop() {}
```

Figure 1: Eksempelkode i wiring som viser *serial.begin()*.

**delay():** Stanser programmet i en periode (i millisekunder) som er spesifisert som parameter. (Det er 1000 millisekunder i løpet av et sekund.) Dette plasseres ofte helt nederst i *loop*-funksjonen.

## 4 Variabler

En stor forskjell i python og wiring er at man definerer variabler ulikt. I wiring må man definere variablene etter hvilke typer de er.

Tall skrives med *int* foran, eks:

**Wiring:** *int tall = 10;*

**Python:** *tall = 10*

Ord skrives med *char* eller *string* foran.

**Wiring:** *string navn = "Ola";*

**Python:** *navn = "Ola"*

Variabler kan også være definert med *unsigned* eller *const* foran. *Unsigned* gir mer plass på dataen, og *const* definerer at variabelen er fastsatt. Mer info om disse og andre ulike typer finnes her:

[https://www.tutorialspoint.com/arduino/arduino\\_data\\_types.htm](https://www.tutorialspoint.com/arduino/arduino_data_types.htm)

---

## 5 Print

For å printe i Arduino IDE, bruker man *serial* foran, mens i python bruker man kun `print()`. Eksempler i wiring:

- `Serial.print(78)` gir "78"
- `Serial.print(1.23456)` gir "1.23"
- `Serial.print('N')` gives "N"
- `Serial.print("Hello world.")` gives "Hello world."

Mer utdypende her: <https://www.arduino.cc/en/pmwiki.php?n=Serial/Print>

## 6 Definere pins

Siden wiringkode kontrollerer et brett, må man definere pinsene også. Dette gjøres ikke i python. For å bruke Arduinopinsene må du definere hvilke pins som blir brukt, og hva de brukes til. En vanlig måte å definere pinsene på er:

`'define pinName pinNumber'`.

Funksjonen er enten input eller output, og er definert ved å bruke `pinMode()` i `setup()`.

## 7 Ressurser

En god side for å sjekke hva ulike uttrykk i wiringkode betyr er: <https://www.arduino.cc/reference/en/>

Denne siden forklarer litt om oppsettet til wiring. <https://www.circuito.io/blog/arduino-code/>

Denne siden forklarer oppsettet til wiring grunnleggende: <https://www.circuito.io/blog/arduino-code/>