



Algoritmer, Loops, Løkker, Programfunktioner mm. for Arduino

Links til følgende i dokumentet:

[Datatyper](#), [Boolean](#)

[Trykknop](#)

[If](#), [If else](#), [If Else If](#)

[While](#), [Do While](#)

[For-Loop](#)

[Break](#),

[PWM](#)

[Serial Read](#)

[Funktioner og subrutinerr](#)

[Switch Case](#)

[Random number, Tilfældigt Nummer](#)

[Array](#), [Matrix](#), [Multidimensionel Array](#)

[Return](#)

Datatyper:

I programmeringsverdenen bruges forskellige data-typer til at gemme variable.

Det er i mange sammenhænge vigtigt at kende den type man gemmer en variabel i. Dels hvor store talværdier, der kan gemmes, og dels forsi compileren har behov for at vide, om et tal starter fra 0 eller en negativ værdi ! Det har noget med måden, negative binære tal gemmes på !! ??

Her er en – foreløbig – oversigt:

Type	Byte	Værdi-område	Forklaring, / adder 1 til max værdi:
bool, Bit, Boolean	1 bit	0 eller 1	false, true HIGH, LOW
int	2	-32768 til 32767	Max værdi + 1 giver -32768; +2 giver -32767 Værdier fra -2^{15} til $(2^{15}) - 1$.
unsigned int	2	0 til 65535	Max værdi + 1 giver 0; +2 giver 1
uint8_t	1	0 til 255	Er det same som en byte. En type af unsigned integer med længden 8 bit.



			"t"-et angiver, at det er navnet på en variabel-"type", ikke fx et funktionsnavn.
byte	1		Brug bare Byte i stedet for int Byte er det samme som uint8_t
char	1	-127 til 128	Til bogstaver, fra ASCII tabellen ! Bruges til at gemme ét enkelt tegn (bogstaver eller tal) i. <i>The compiler will attempt to interpret this data type as a character in some circumstances, which may yield unexpected results. characters are stored as numbers.</i>
unsigned char	1	0 til 255	
long	4	-2147483648 til 2147483647	Max værdi + 1 giver -2147483648; 2 more = -2147483647
unsigned long	4	0 til 4294967296	Max værdi + 1 giver 0; +2 giver 1 0 til (2 ³² - 1).
float	4	1,17E-38 til 3,40E+38	Decimaltal, signed nummer fra -3.4028235E38 til 3.4028235E38.

Fra <http://forum.arduino.cc/index.php?topic=41590.0>>

Omregning mellem datatyper: "Cast"

```
int i;
float f;

f = 3.6;
i = (int) f; // now i is 3. Decimaler forsvinder !! ( truncated )
```

Boolean

Når man læser en pin, kan man gemme den læste værdi, et 0 eller 1, i en Boolean, dvs. en variabel bestående af blot 1 bit.

Eksempler:

```
boolean running = false;

//En Bool kan testes som fx:

void setup() {
  // code here, to run once:
}
```

```

void loop() {
    if (running == true) {
        // do
    };

    if (running == 1){
        // do
    };

    if (running) {
        // do
    };
}

int pinOut = 7;
int pinA = 8;
int pinB = 9;
boolean pinAState = 0; // Variable kan definers her
boolean pinOutState;

void setup()
{
    pinMode(pinOut, OUTPUT);
    pinMode(pinA, INPUT);
    pinMode(pinB, INPUT);
}

void loop()
{
    pinAState = digitalRead(pinA);
    boolean pinBState = digitalRead(pinB); // Eller her
    pinOutState = pinAState & pinBState; // and
    digitalWrite(pinOut, pinOutState);
}

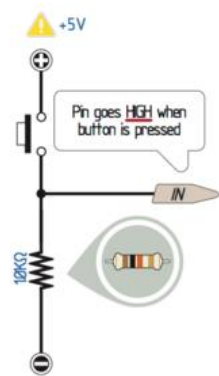
```

Trykknapp: Button, Se eksempler / Digital / Button

```
int buttonpin=2;
```



```
pinMode(buttonpin, INPUT); // I setup !!  
  
// Placeres I Loop:  
  
if (digitalRead(buttonpin) == HIGH) {  
    // do something  
} // EndIF
```



Decision: If:

```
// en blok kode der køres hvis en betingelse er opfyldt  
  
// If:   Eksempel:  
  
if (x > 120){  
    digitalWrite(LEDpin1, HIGH);  
    digitalWrite(LEDpin2, HIGH);  
}  
// EndIF  
  
/* Bemærk:  if ( x == 10 )      // der skal 2 lighedstegn til for at teste om  
x er lig 10. Ellers får x bare værdien 10  
*/
```

If - Else

```
// If else  
  
// en blok kode køres hvis en betingelse er opfyldt, ellers køres en anden  
blok  
  
if (x < 500)  
{  
    // action A  
}  
else      // Else-delen kan udelades  
{  
    // action B. Udføres hvis x >= 500.  
} // EndIF
```

If – Else If



(efter en analog Read)

```
if (pinFiveInput < 500)
{
  // do Thing A
}
else if (pinFiveInput >= 1000) // Udføres kun hvis den første "if" er falsk,
                               // og denne test er sand.
{
  // do Thing B
}
else
{
  // do Thing C
} // EndIF
// Se: https://www.arduino.cc/en/Reference/Else
```

En if - statement kan efterfølges af en optional "else if"...og eller "else" statement. Dette er meget brugbart til at teste forskellige betingelser.

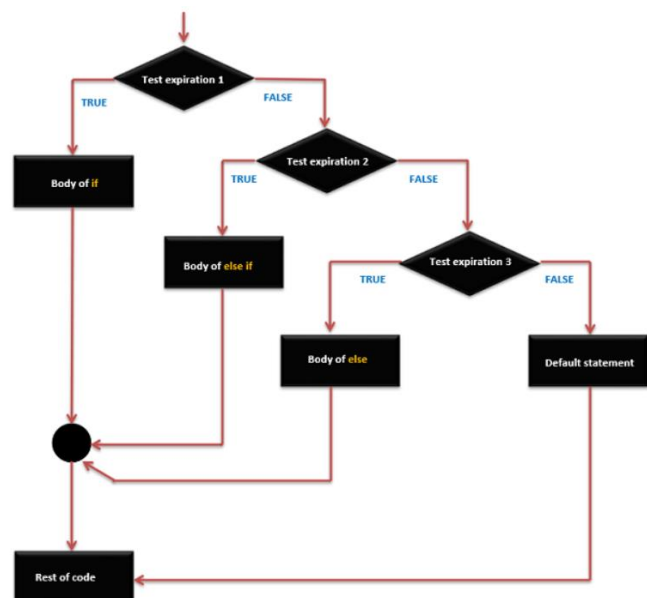
En if kan have ingen eller 1 "else" statement, og den skal komme efter eventuelle "else if".

En if kan have fra 0 op til mange "else if" statements, og de skal komme før "else".

Hvis en "if" lykkes, skippes resten i strukturen.

Hvis en "else if" lykkes, bliver ingen af de følgende "else if" eller "else" statements testet.

Hvis hverken "if" eller "else if" lykkes, udføres "else" hvis defineret.



Kilde: https://www.tutorialspoint.com/arduino/arduino_if_else_if_else_statement.htm

Think of these questions´:

If she is hungry
give her food
else if she is thirsty
give her drink
else
do nothing

Test conditions and do different things



- Can be:
- `if {}`
- `if {} else {}`
- `if {} else if {} else {}`
- Can have as many “else if” as you Want

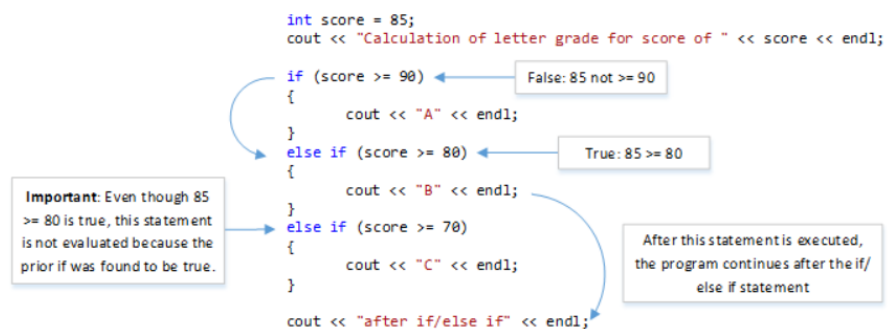
- Use `if` to specify a block of code to be executed, if a specified condition is true
- Use `else` to specify a block of code to be executed, if the same condition is false
- Use `else if` to specify a new condition to test, if the first condition is false

An `else` statement is matched to the closest previous `if` statement that does not already have its own `else` statement.

Proper indenting makes it easier to see the matching. Just remember that in an `if` or `else` statement, if the expression is true, the program executes the following curly bracketed code block or single statement.

Eksempler fra derude:

```
if (isRaining)
{
    Inner if/
    else
    {
        if (temp > 45)
            cout << "Wear light weight rain coat" << endl;
        else
            cout << "Wear fleece and rain coat " << endl;
    }
    else if (isSnowing)
    {
        Inner if/
        else if
        {
            if (temp > 20)
            {
                cout << "Wear soft shell jacket" << endl;
            }
            else if (temp > 0)
            {
                cout << "Wear down jacket" << endl;
            }
            else
            {
                cout << "Wear base layers and down jacket" << endl;
            }
        }
    }
    else
        cout << "It is hard to come up with interesting examples" << endl;
}
```



Fra https://www.w3schools.com/java/java_conditions.asp

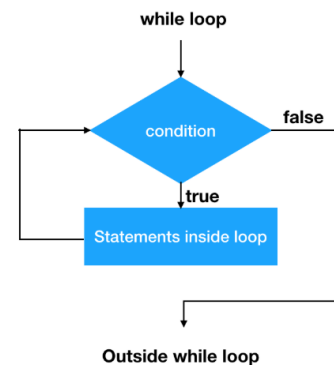


While:

Eksempler:

```
var = 0;
while(var < 200){
    // gentag indtil testen er falsk.
    // I dette tilfælde 200 gange
    var++;
} // EndWhile
```

Fra <<https://www.arduino.cc/en/Reference/While>>



```
while (digitalRead(10) == HIGH)
{
    // Kode der skal køres i løkken
}
```

// Fra <http://www.htx-arduino.dk/index.php?title=Software_og_Udviklingsmilj%C3%B8#Analog_I.2FO>

Der testes i begyndelsen af delprogrammet. Hvis betingelsen er falsk, udføres delprogrammet ikke.

```
void loop()
{
    while( digitalRead(5) == 1 )           // while the button is pressed
    {
        //blink
        digitalWrite(3,HIGH);
        delay(1000);
        digitalWrite(3,LOW);
        delay(1000);
    } // EndWhile
}
```

When you want to check the condition at the beginning of the loop, simply negate the condition on a standard while loop

```
while(!cond) { ... }
```



While med break –eksempel:

```
void alarm()
{
    int AlarmState = 0; //Just to get while loop going
    int count = 1;
    while(AlarmState==0)
    {
        AlarmState = digitalRead(alarmStop);
        digitalWrite(buzz,HIGH);
        delay(500);
        digitalWrite(buzz,LOW);
        delay(500);
        count++;
        if(count>=5){
            break;
        }
    } // Hop hertil
}
```

Break:

En Break-dtatement bruges til at hoppe ud af en ”do”, ”for” eller ”while” -loop, og fra en switch-statement.

Man kan fx anbringe en if-statement inde i en while-loop, for at tjekke, om en loop skal afbrydes

```
if (statement) break;
```

Men det er jo også en mulighed at gøre betingelsen for loopet falsk !

Do – while: Svarer til ”Repeat Until ” som compileren dog ikke forstår

Her er et par eksempler:

Delprogrammet udføres mindst 1 gang: Gentages indtil en betingelse er falsk.

```
do
{
    Statements // do stuf
} while ( Condition )
// EndDoWhile
```

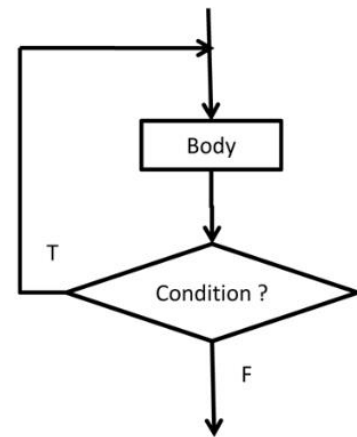
```
do
{
    delay(50);           // wait a bit
```



```
x = x+1;           // do stuf
} while (x < 100);  // Until x >= 100
// EndDoWhile
```

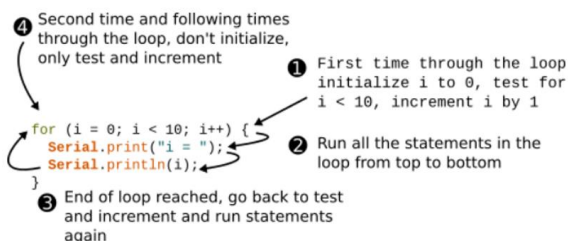
```
do
{
    // Kode der skal køres i løkken
} while (analogRead(A0) < 100);
```

```
// Repeat until - alike:
do
{
    // whatever
} while ( !condition );
```

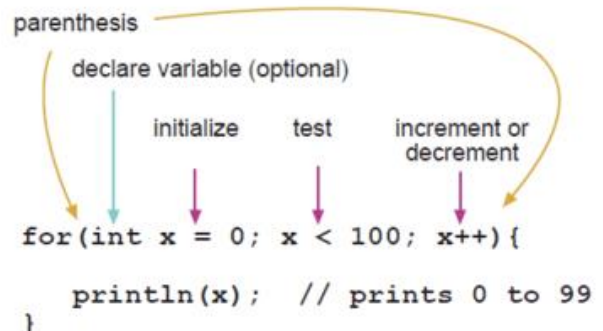


For-Loop.

En kodestump udføres et bestemt antal gange, eller indtil:



for Loop Program Flow



```
// Eksempel, analogWrite
```

```
// Værdi fra 0 til 255 !!!
```

```
void loop()
{
    int x = 1;
    for (int i = 0; i > -1; i = i + x) {
        analogWrite(PWMPin, i);
        if (i == 255) x = -1;           // switch direction at peak
        delay(10);
    }
}
```

Pseudokode for for-loop::



Initialiser:

Loop:

Udfør test, Hvis sand

```
{ udfør program;  
  udfør Increment / Decrement.  
}
```

Indtil Test er falsk, så hop til efter loop-en.

Endloop:

```
for ( variable initialization; condition; variable update ) {  
  
    //Code to execute while the condition is true  
}
```

Eksempel:

```
int PWMpin = 10;           // Dim an LED using a PWM pin  
                           // LED in series with 470 ohm resistor on pin 10  
  
void setup() {  
    // no setup needed  
}  
  
void loop() {  
    for (int i = 0; i <= 255; i++) {  
        analogWrite(PWMpin, i);  
        delay(10);  
    } // Endfor  
} // Endloop
```

Eksempel:

```
// analogWrite eksempel  
  
// Værdi fra 0 til 255 !!!  
  
void loop()  
{  
    int x = 1;  
    for (int i = 0; i > -1; i = i + x) {  
        analogWrite(PWMpin, i);  
        if (i == 255) x = -1;           // switch direction at peak  
        delay(10);  
    } // Endfor  
}
```



Increment kan godt være en anden beregning!!

```
for (int x = 2; x < 100; x = x * 1.5) {  
    // Do something  
}
```

For-Loop med break

```
for (int x = 0; x < 10; x = x++ ) {  
    if (i == 2) {  
        break; // Hop ud af løkken hvis i = 2, Hop til Hertil  
    }  
    // Do something  
}  
  
// Hertil:
```

En anden mulighed end break; er jo at gøre x = 10 !!!

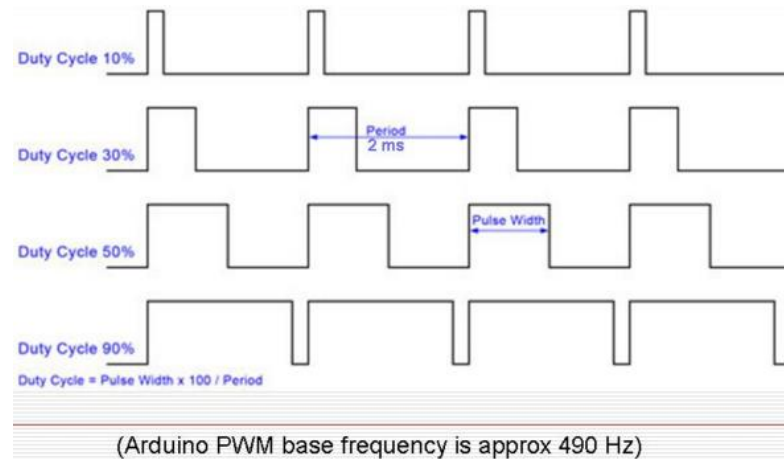
For-Loop med continue

```
for (int x = 0; x < 10; x = x++ ) {  
    if (i == 2) {  
        continue; // Undlad "Do something" hvis i = 2  
    }  
    // Do something  
}
```

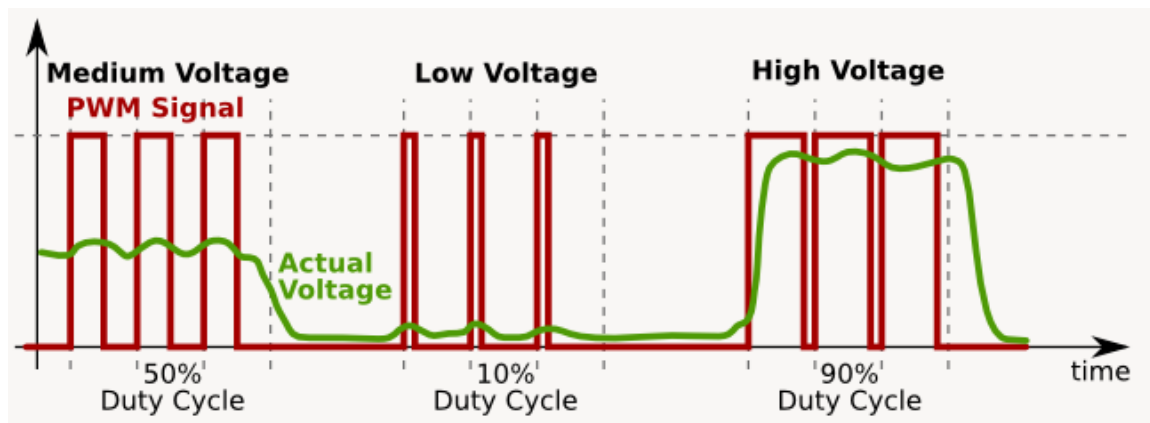


PWM.

Værdier til pins fra 0 til 255



Bare det går hurtigt nok, ses det ikke på LED.



```
// Dim an LED using a PWM pin
int PWMpin = 10; // LED in series with 470 ohm resistor on pin 10

void setup()
{
  // no setup needed
}

void loop()
{
  for (int i=0; i <= 255; i++){
    analogWrite(PWMpin, i);
    delay(10);
  } // Endfor
}
```



```
/*
  Fade

  This example shows how to fade an LED on pin 9
  using the analogWrite() function.

  The analogWrite() function uses PWM, so if
  you want to change the pin you're using, be
  sure to use another PWM capable pin. On most
  Arduino, the PWM pins are identified with
  a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.

  This example code is in the public domain.
  */

int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }  // Endif
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

Serial Read

```
char rx_byte;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT); // LED on pin 13 of UNO
}
```



```
void loop() {  
  
    if (Serial.available() > 0) { // is a character available?  
        rx_byte = Serial.read(); //  
        if (rx_byte == 'a') {  
            digitalWrite(13, HIGH);  
        } // Endif  
  
        if (rx_byte == 'b') {  
            digitalWrite(13, LOW);  
        } // Endif  
    } // Endif  
}
```

// Fra: <https://startingelectronics.org/software/arduino/learn-to-program-course/09-if-statement/>

Eksempel på, at der modtages 2 bytes, fx B1, fra Debugvinduet:

```
int pass[2];  
  
void setup() {  
    //  
    Serial.begin(9600);  
}  
  
void loop() {  
    if (Serial.available() >= 2) // når der er ankommet 2 bytes,  
    {  
        for (int i = 0; i < 2; i++)  
        {  
            pass[i] = Serial.read();  
        }  
  
        Serial.print(pass[0]);  
        Serial.println(pass[1]);  
    }  
}
```

// i ovenstående skal man slå fra, at debugvinduet efter to byte sender et linefeed, 0Ah. Ellers sendes der 3 byte

To eksempler på, hvordan man kan tømme den serielle modtage-buffer:

De er udlagt som funktioner:

```
// tøm Serial modtage-Buffer:  
  
void serialFlush() {  
    while (Serial.available() > 0) {  
        char t = Serial.read();  
    }  
}
```



```
void serial_flush(void) {  
    while (Serial.available()) Serial.read();  
}
```

Serial.flush(); // vent til Sendebuffer er tømt !!

Se speciel dokument for mere info!!

Brug af Funktioner eller subrutiner:

Her er vist et par eksempler på brug af Funktioner.

```
void setup() {  
    Serial.begin(9600);  
    DashedLine(); // Kald Funktion, ( Subroutine )  
    Serial.println("Print something");  
    DashedLine();  
}
```

```
void loop() {          // Do nothing  
}
```

```
void DashedLine()  
{  
    Serial.println("-----");  
}
```

```
void setup() {  
    Serial.begin(9600);  
  
    // draw the menu box  
    DashedLine(24);  
    Serial.println("| Program Options Menu |");  
    DashedLine(22);  
}
```

```
void loop() {  
}
```

```
void DashedLine(int len)  
{  
    int i;  
  
    // draw the line  
    for (i = 0; i < len; i++) {  
        Serial.print("-");  
    }  
    // move the cursor to the next line
```



```
Serial.println("");  
}
```

Switch case:

En Switch Case struktur er smart hvis der skal udføres noget fx afhængig af en variabels værdi.

Kan bruges i stedet for en række if-statements.

```
switch (switch_var) {  
  case '1':  
    // statements placed here run if switch_var == '1'  
    break;  
  case '2':  
    // statements placed here run if switch_var == '2'  
    break;  
  default:  
    // statements placed here run if if no case found  
    break;  
}
```

Annotations:

- switch keyword
- Switch variable - can be char or int
- Opening brace of switch body
- int or char constant to check for
- If case matches, statements are run followed by break which breaks out of the switch body
- Closing brace of switch body

Structure of an Arduino switch Statement

Eksempel:

Switch Case sammenlignet med flere if-statements:

```
switch ( variable)  
{  
  case value1:  
    // do something  
    break;  
  
  case value2:  
    // do something else  
    break;  
  
  default:  
    // do something by default  
    break;  
} // end switch
```

// this is equivalent to:

```
if ( variable == value1 )  
{  
  // do something  
}  
else if ( variable == value2 )  
{  
  // do something else  
}  
else  
{  
  // do something by default  
}
```



Eksempel:

```
// der testes på variablen state
switch (state)
{
    case 1:
        // Kode hvis state er 1
        break;
    case 2:
        // Kode hvis state er 2
        break;
    default:
        // Kode hvis state hverken er 1 eller 2
} // end switch
```

Man kan også bruge en switch case, så samme funktion udføres ved forskellige værdier:

```
char c = Serial.read();

switch(c) {
    case 'a' ... 'z':
    case 'A' ... 'Z':
        Serial.print ("result = alpha");
        break;
    case '0' ... '9':
        Serial.print ("result = number");
        break;
    default:
        Serial.print ("result = other");
        break;
} // end switch
}
```

Eller hvis værdierne er i et område. Fx hvis der læses på en analog port.

```
//switch case statement

int number = 0;

number = analogRead( A0 );
switch(number)
```



```
{
  //case values within a range
  case 1 ... 50:
    Serial.print("Number is in between 1 to 50");
    break;
  //case values within a range
  case 51 ... 100:
    Serial.print("Number is in between 51 to 100");
    break;
  //default case
  default:
    Serial.print("Number is out of range!!! ");
    break;
} // end switch
}
```

Her er switch-variablen fundet efter et tabelopslag i et Array:

```
void fun() {
  int data[10] = { 5, 4, 10, 25, 60, 47, 23, 80, 14, 11};
  int i;
  for (i = 0; i < 10; i++) {
    switch (data[i]) {
      case 1 ... 10:
        Serial.print("in range 1 to 10 ");
        Serial.print(data[i]);
        break;
      case 11 ... 20:
        Serial.print("in range 11 to 20 ");
        Serial.print(data[i]);
        break;
      case 21 ... 30:
        Serial.print("in range 21 to 30 ");
        Serial.print(data[i]);
        break;
      case 31 ... 40:
        Serial.print("in range 31 to 40 ");
        Serial.print(data[i]);
        break;
      default:
        Serial.print("Exceeds the range ");
        Serial.print(data[i]);
        break;
    } // end switch
  }
}
```

```
pins = PINB & 0x7;           // read pins 8/9/10
switch (pins) {
  case 0:
```



```
                                // no switches set
    break;
    case 1:
        switch1_func();
        break;
    case 2:
        switch2_func();
        break;
    case 3:                        // switches one and two.
        switch1_func();
        switch2_func();
    case 4:                        // switch 3 makes 1&2 into "don't care"
    case 5:
    case 6:
    case 7:
        switch3_func();
        break;
} // end switch case
}
```

Random Tilfældig nummer:

Syntax:

```
random(max)
random(min, max)
random(2, 7)
```

Bemærk !Min-grænsen er inklusiv, max er eksklusiv !

```
// print a random number from 10 to 19
randNumber = random(10, 20);
Serial.println(randNumber);
```

```
void loop(){
    int i=random(4);
    digitalWrite(ledPin[i], HIGH);
    while(digitalRead(buttonPin[i]) == LOW)
    { //do nothing
    } // EndWhile
    digitalWrite(ledPin[i], LOW);
}
```

Random fra 0 til 3

randNumber = random(300);

random number from 0 to 299

randNumber = random(10, 20);
from 10 to 19

Se: <https://www.arduino.cc/en/Reference/Random>



Et par eksempler mere:

```
long randNumber;

void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));    // Læs tilfældig værdi ( 0 til 1023 ) på
                                // svævende pin, og brug værdien som basis
                                // basis for en pseudo random number-
                                // generator.
}

void loop() {
  randNumber = random(300);      // generer værdi fra 0 til 299 !!
  Serial.println(randNumber);

  delay(50);
}
```

```
long randNumber;

void setup() {
  Serial.begin(9600);

  // if analog input pin 0 is unconnected, random analog
  // noise will cause the call to randomSeed() to generate
  // different seed numbers each time the sketch runs.
  // randomSeed() will then shuffle the random function.
  randomSeed(analogRead(0));
}

void loop() {

  randNumber = random(300);
  Serial.println(randNumber);    // print a random number from 0 to 299

  randNumber = random(10, 20);
  Serial.println(randNumber);    // print a random number from 10 to 19

  delay(50);
}
```

```
// if analog input pin 0 is unconnected, random analog
// noise will cause the call to randomSeed() to generate
// different seed numbers each time the sketch runs.
// randomSeed() will then shuffle the random function.

randomSeed(analogRead(0));
```

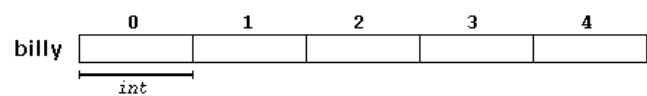


```
//generate five random bytes
for (int i=0; i<5; i++) {
    myBytes[i] = random(0, 256);
}
```

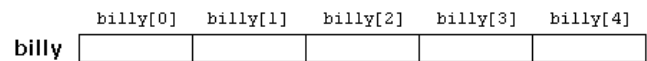
Array

Eksempler på at definere et array med værdier: Husk, at det første element i et array har nummer 0.

Her ses en grafisk præsentation af et array, der hedder Billy.

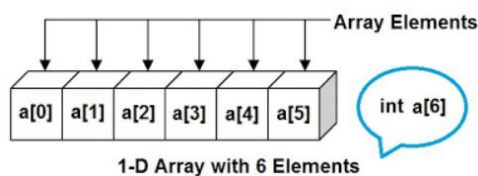


Det har 5 elementer, nummre fra 0 til 4.



Se evt. eksempler her: <http://www.cs.uregina.ca/Links/class-info/207/Lab7/>

Et Array er en samling data, der kan adresseres via et index.



Arrayet har 6 elementer, #0 til #5

Man kan forestille sig denne opbevaringskasse med inddelinger som et array. #0 er den forreste, #1 den næste osv.





I dette array er der lagt bynavne.

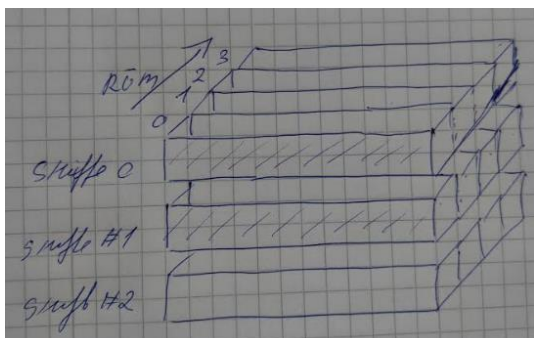
De to venstre er to forskellige array med hver deres navn. Cities og Temperatures.

Det højre er et to-dimensionel array.

Bør der **ikke være 8 elementer i den ???**

	Cities(7)	Temperatures(7)	Temperatures(7, 1)
0	San Francisco	78	San Francisco 78
1	Los Angeles	86	Los Angeles 86
2			
3			
4			
5			
6			
7	Seattle	65	Seattle 65

Two one-dimensional arrays A two-dimensional array



Her er der tre skuffer med hver 4 rum.

Reol [3] [4]

Matrix!!

	Column 0	Column 1	Column 2
Row 0	X[0][0]	X[0][1]	X[0][2]
Row 1	X[1][0]	X[1][1]	X[1][2]
Row 2	X[2][0]	X[2][1]	X[2][2]

2D-Array



Hedder det 2-dimensionel Array – eller et Matrix

Arrays have only one dimension, a row of elements while a **matrix** has more dimensions, mostly 2 but can have more than that, and its elements are either column or row **arrays** which are here called vectors for a **matrix** with two dimensions or **matrices** with dimension $n-1$ for a **matrix** of n dimensions.



Et to-dimensionel array !! Eller Matrix



Dette må være et 3-dimensionel Matrix

Eksempler:

```
int myInts[6];  
int myPins[] = {2, 4, 8, 3, 6};  
int mySensVals[6] = {2, 4, -8, 3, 2};  
char message[6] = "hello";
```

`mySensVals[0] == 2, mySensVals[1] == 4, /* and so forth.`

It also means that in an array with ten elements, index nine is the last element. Hence:
*/

```
int myArray[10]={9,3,2,4,3,2,7,8,9,11};  
  
    // myArray[9]    contains 11  
    // myArray[10]   is invalid and contains random information (other memory  
address)
```

Eksempel: Brug array til at definere pins.

```
int ledPins[] = {9, 3, 2, 12, 15, 11, 7, 6}; // LED pins  
int ledCnt = 8;
```



```
void setup()
{
    for(int p=0; p<ledCnt; p++)
    {
        pinMode(ledPins[p], OUTPUT); // Set the mode to OUTPUT
    }
}

void loop()
{
    for(int p=0; p<ledCnt; p++)
    {
        analogWrite(ledPins[p], HIGH); // Turn an LED pin on
        delay(100); // Twiddle thumbs for a while
        analogWrite(ledPins[p], LOW); // Turn an LED pin off
        delay(100); // Twiddle thumbs some more
    } // Endfor
}
```

Eksempel: Brug af Array til at definere pins.

```
// This example code is in the public domain.
// http://www.arduino.cc/en/Tutorial/Array

int timer = 100; // The higher the number, the slower the timing.
int ledPins[] = {
    2, 7, 4, 6, 5, 3 }; // an array of pin numbers to which LEDs
                        // are attached
int pinCount = 6; // the number of pins (i.e. the length of
                  // the array)

void setup() {
    // the array elements are numbered from 0 to (pinCount - 1).
    // use a for loop to initialize each pin as an output:

    for (int thisPin = 0; thisPin < pinCount; thisPin++) {
        pinMode(ledPins[thisPin], OUTPUT);
    } // Endfor
}

void loop() {
    // loop from the lowest pin to the highest:

    for (int thisPin = 0; thisPin < pinCount; thisPin++) {

        // turn the pin on:
        digitalWrite(ledPins[thisPin], HIGH);
        delay(timer);

        // turn the pin off:
        digitalWrite(ledPins[thisPin], LOW);

    } // Endfor
}
```



```
// loop from the highest pin to the lowest:

for (int thisPin = pinCount - 1; thisPin >= 0; thisPin--) {

    // turn the pin on:
    digitalWrite(ledPins[thisPin], HIGH);
    delay(timer);

    // turn the pin off:
    digitalWrite(ledPins[thisPin], LOW);
} // Endfor
}
```

Matrix Multi-dimensionel array

En Matrix er en multi dimensionel Array

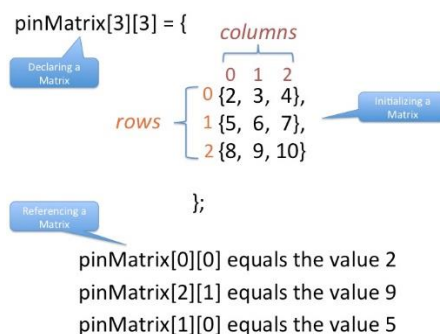
Eksempel:

```
// Pins kontrolleret af et matrix

int pinMatrix[3][3] = {
    {2, 3, 4},
    {5, 6, 7},
    {8, 9, 10}
};

void setup() {
    //use a nested for-loop to initialize all the pins
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            pinMode(pinMatrix[i][j], OUTPUT);
        } //close for i
    } //close for j
} //close setup()

void loop() {
    //this nested for-loop will turn each LED on and off in sequence
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            digitalWrite(pinMatrix[i][j], HIGH);
            delay(100);
            digitalWrite(pinMatrix[i][j], LOW);
        } //close for i
    } //close for j
} //close loop()
```



Man kan også definere matrix-indhold på følgende måde:

```
int pinMatrix[3][3] = { { 2, 3, 4}, {5, 6, 7}, {8, 9, 10} };
```

Multidimensionel array fx brugt til at styre 7-segmenter:

// Kan fx bruges til at definere bitmønsteret til et 7-segment.

```
bool num_array[10][8] = {  
    { 1,1,1,1,0,1,1,0 }, // 0  
    { 1,0,0,1,0,0,0,0 }, // 1  
    { 1,1,0,1,0,1,1,0 }, // 2  
    { 1,1,0,1,1,1,0,0 }, // 3  
    { 1,0,1,1,1,0,0,0 }, // 4  
    { 0,1,1,1,1,1,0,0 }, // 5  
    { 0,1,1,1,1,1,1,0 }, // 6  
    { 1,1,0,1,0,0,0,0 }, // 7  
    { 1,1,1,1,1,1,1,0 }, // 8  
    { 1,1,1,1,1,1,0,0 };// 9
```

// Arrayet bruges fx som følgende:

// Digit_1

```
for (int pin = 5; pin < 12; pin++)  
{  
    digitalWrite(pin, num_array[digit1][pin - 5]); // definer 8 pins afhængig af  
    tallet i digit1.  
}  
digitalWrite(12, HIGH); // tænd 7-segmentet kortvarigt.  
delay(5);  
digitalWrite(12, LOW);
```

// Digit_2

```
for (int pin = 5; pin < 12; pin++)  
{
```



```
digitalWrite(pin, num_array[digit2][pin - 5]);  
} // Endfor  
  
digitalWrite(13, HIGH);  
delay(5);  
digitalWrite(13, LOW);  
}
```

Tæl ned på et 7-segment: Eksempel:

```
/*  
7 Segment Count Down  
This example code is in the public domain.  
*/  
  
boolean numArray[10][8] = {  
  {1,1,1,0,1,1,0,1}, //Zero  
  {1,0,0,0,1,0,0,0}, //One  
  {0,1,1,0,1,0,1,1}, //Two  
  {1,0,1,0,1,0,1,1}, //Three  
  {1,0,0,0,1,1,1,0}, //Four  
  {1,0,1,0,0,1,1,1}, //Five  
  {1,1,1,0,0,1,1,1}, //Six  
  {1,0,0,0,1,0,0,1}, //Seven  
  {1,1,1,0,1,1,1,1}, //Eight  
  {1,0,1,0,1,1,1,1}  //Nine  
};  
  
void setup() {  
  for(int i = 10; i <= 17; i++) {  
    pinMode(i, OUTPUT);  
    digitalWrite(i, HIGH);  
  } // Endfor  
}  
  
void loop() {  
  for(int a = 9; a >= 0; a--){ //Count down through the Numbers  
    for(int j = 0; j <= 8; j++){ //Loop through the digital pins  
      digitalWrite(j+10, !numArray[a][j]);  
    } // Endfor  
  
    delay(1000);  
    for(int j = 10; j <= 17; j++) { //Turn all the LEDs off  
      digitalWrite(j, HIGH);  
    } // Endfor  
  } // Endfor Ydre  
}
```

Debounce



```
// buttonPin er med intern pullup

const int debounceTime = 20; // debounce in milliseconds

void deBounce ()
{
  unsigned long now = millis ();
  do
  {
    // on bounce, reset time-out
    if (digitalRead (buttonPin) == LOW)
      now = millis ();
  }
  while (digitalRead (buttonPin) == LOW || (millis () - now) <= debounceTime);
  // || = or.
} // end of deBounce
```

Return

Using return within setup():

- exits setup and starts with loop()

Using return within loop():

- exits loop() and starts with loop() again

Fra <<https://forum.arduino.cc/index.php?topic=45322.0>>

Max, Min:

Example Code

The code ensures that sensVal is at least 20.

```
sensVal = max(sensVal, 20); // assigns sensVal to the larger of sensVal or 20
                        // (effectively ensuring that it is at least 20)
```

Her mangler mere !!

Mere til historien:

Define:



```
#define repeat(n) for (int i = 0; i < n; i++)
```

```
repeat(10) {  
    //do stuff  
}
```