

```
>>> C-Programmering for begyndere  
>>> Del 2 - Operatorer, conditionals, loops og forgreninger
```

Name: Jacob B. Pedersen<sup>†</sup> og Jakob S. Nielsen<sup>‡</sup>

Date: 16. april 2018

---

<sup>†</sup>jacob.bp@mvb.net

<sup>‡</sup>jakob990@gmail.com

```
>>> Indhold
```

## 1. Repetition

Hvad lavede vi sidste gang?

Hello World

Datatyper

Input og output

## 2. Kreative Opgaver

```
>>> Hvad lavede vi sidste gang?
```

- \* Vi fik sat vores første program op
  - \* Hello World
- \* Stiftede bekendtskab med et par datatyper
  - \* `int`, `float`, `char`, `bool`
- \* Vi tog input, og skrev ud på konsollen
  - \* `scanf()` og `printf()` fra `stdio.h`

```
>>> Hello World
```

- \* Begynderprogrammet

- \* Indeholdt `main()`, `stdio.h`, `printf()` og en `return 0`

---

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("Hello world\n");
6      return 0;
7  }
```

---

>>> Datatyper

\* Et lille **cheatsheet** over datatyperne i C:

---

```
1  int i; // Integer/heltal - *mindst* 16 bits
2      // Fra [-32767 til +32767]
3  float f; // Floating point decimal - 32 bits
4      // *mindst* 6 betydende cifre
5  char ch; // Character - *mindst* 8 bits
6      // ASCII dækker fra 0-128 forskellige tegn
7  bool b; // Boolean - true/false - *mindst* 8 bits
8      // Kun 0 anses som Boolsk "false" ved tilskrivning
9      // Skal hentes med #include <stdbool.h>
```

---

>>> Datatyper

\* Lidt flere udvidede datatyper:

---

```
1 unsigned int ui; // Integer/heltal - *mindst* 16 bits
2 // [0 til +65535]
3 short int f; // Short integer - *mindst* 16 bits
4 // [ 32767 til +32767]
5 long int li; // Long integer - *mindst* 32 bits
6 // [ 2147483647 til +2147483647]
7 double db; // Double precision float - *mindst* 64 bits
8 // *mindst* 10 betydende cifre
9 long double ldb; // Double precision float - *mindst* 128
  bits - *mindst* 10 betydende cifre
```

---

## >>> Input og output

\* Til input og output bruges `#include <stdio.h>`

\* `scanf()` tager input, eks:

---

```
1 scanf("%d %d", &variable1, &variable2);
```

---

\* `printf()` skriver output, eks:

---

```
1 printf("Variable 1: %d Variable 2: %d", variable1,  
    variable2);
```

---

\* Vi bruger **format specifiers** til at fortælle `printf()` og `scanf()` hvilken datatype det skal fortolkes som

---

```
1 "%d" // Integer format specifier - printer/scanner heltal  
2 "%f" // Float format specifier - printer/scanner kommatal  
3 "%.2f" // Float med 2 decimaltal  
4 "%c" // Char format specifier - printer/scanner skrifttegn  
5 "%s" // String format specifier - printer/scanner  
    skriftstreng
```

---

## >>> Kreative Opgaver

- \* Der ligger kreative opgaver tilgængelige:
  - \* [../Del\\_2/Exercises/C\\_exercises\\_2\\_dansk.pdf](#)
- \* Der er hjælp at hente her på workshopen
- \* God arbejdslyst! - Happy Hacking!