



Översikt

- **Mål:**
 - Få kunskap om variabler och data typer.
 - Få kunskap om arrays.
 - Kunna kontrollera programflödet.
 - Arbeta med loopar.

.eerec

Variabler och data typer

- **Variabler**
 - Lagringsplats.
 - Innehåller information.
 - Vara del i ett C# uttryck.
- **Data typer**
 - Boolean.
 - Numeriskt: integrals, floating point och decimal.
 - String.

.eerec

Arrays

- Area där information kan lagras.
- Exempel: `int[] myInts = (5, 10, 15);`

.eeec

Kontrollera flöde

- I varje programmeringsspråk är det viktigt att kunna kontrollera flödet.
- `if` statement.
- `switch` statement.

.eeec

if statement

- Vanligaste sättet att kontrollera flödet.
- Ger möjlighet att arbeta med olika vägar för logiken, beroende på givet villkor.
- När villkoret undersökts till sant (`true`), kommer ett block av kod att exekveras.
- Kan användas ensamt eller tillsammans med `else if` och `else`.

.eeec

if statement exempel

```
bool myTest;
myTest = false;
if (myTest == false)
    Console.WriteLine("Hello1");
Console.WriteLine();

bool myTest;
myTest = false;
if (myTest == false)
{
    Console.WriteLine("Hello1");
    Console.WriteLine("Hello2");
}
Console.WriteLine();
```

.cs/ce

if statement exempel (forts.)

```
{
    bool condition = true;
    if (condition)
    {
        Console.WriteLine("Variabel är satt till true.");
    }
    else
    {
        Console.WriteLine("Variabel är satt till false.");
    }
    Console.WriteLine();
}
```

.cs/ce

if statement exempel (forts.)

```
int myTest;
myTest = 2;
if (myTest == 1)
{
    Console.WriteLine("Hello1");
}
else if (myTest == 2)
{
    Console.WriteLine("Hello2");
}
else
{
    Console.WriteLine("Hello3");
}
Console.WriteLine();
```

.cs/ce

switch statement

- Ett annat sätt att kontrollera flödet, är med `switch statement`.
- `switch` exekverar ett set av logik, beroende på värde för given parameter.
- Parameter kan vara `boolean`, `enums`, `integral types` och `string`.

.eeec

switch statement exempel

```
switch myTest;
{
    case 1:
        MessageBox.Show("Hello1");
        break;

    case 2:
        MessageBox.Show("Hello2");
        break;

    case 3:
        MessageBox.Show("Hello3");
        break;
}
```

.eeec

Övning if och switch



.eeec

Loopar

- I C# kan följande typer av loopar användas:
 - while loop.
 - do loop.
 - for loop.
 - foreach loop.

.eerec

while

- while loop kommer att undersöka ett villkor, så länge detta villkor uppfylls (true) kommer block av kod att exekveras.

- Exempel:

```
int myInt = 0;
while (myInt < 10)
{
    MessageBox.Show("Inside Loop: " + myInt.
        ToString());
    myInt++;
}
MessageBox.Show("Outside Loop: " + myInt.
    ToString());
```

.eerec

do loop

- do loop liknar while loop med undantaget att kontroll av villkoret sker i slutet av loop.
- Garanterar att do loop exekveras minst en gång.

.eerec

do loop exempel

```
int myInt = 0;
do
{
    MessageBox.Show("Inside Loop: " + myInt.
        ToString());
    myInt++;
} while (myInt < 10);
MessageBox.Show("Outside Loop: " + myInt.
    ToString());
```

.eerec

for loop

- for loop fungerar som while loop med undantaget att syntax inkluderar initialisering och modifikation.

Exempel: `for i = 0; i < 10; i++`)

- for loop används när du vet exakt hur många gånger du vill utföra ditt uttryck inom loop.

.eerec

for loop exempel

```
for (int i =0; i < 10; i++)
{
    MessageBox.Show("Inside Loop: " + myInt.
        ToString());
    myInt++;
}
MessageBox.Show("Outside Loop: " + myInt.
    ToString());
```

.eerec

foreach loop

- `foreach` loop används för att läsa igenom lista över poster.
- Fungerar på array och collections.

.eeec

foreach loop exempel

```
string[] names = { "Kalle", "Ture", "Sven" };  
  
foreach (string person in names)  
{  
    MessageBox.Show(person);  
}
```

.eeec

Övning loopar



.eeec

Repetitionsfrågor

.course
