Variabler

En variabel är ett namn på ett värde.
Man skapar variabler med =

```
a = 5
```

När man skriver `a` senare i programmet ersätts den med värdet 5.

Typer

tal	Ex: 0, 10, -45, 2.44
strängar	Text, skrivs inom "". Ex: "Nautilus"
listor	sekvens av värden, t ex tal eller strängar. Ex. [1,2,3,4]

Funktioner

```
def add (a,b):
    return a + b
```

Definierar funktionen *add* som tar två argument: a och b.

`return`används för att returnera värde tillbaka från funktionen.

Strängar

En sträng är en lista av bokstäver.
Strängar skrivs genom att sätta " runt texten som ska ingå i strängen.

```
s="Alla vägar bär till Rom"
```

Index

```
Varje bokstav i en sträng har ett index. Index böjar
på 0 och går till strängens längd - 1.
Exempel: här är sträng "Alfabetet" med varje bokstavs
index skrivet under bokstaven
A l f a b e t e t
0 1 2 3 4 5 6 7 8
```

Hitta index for en bokstav

För att hitta index för en bokstav i en sträng kan man använda *find* på strängen.
Exempel: om `s="Alfabet"` så kan vi få ut vilket index "b" har med

```
s.find("b")
```

I det här fallet blir svaret 4.
Om bokstaven in finns i strängen så returnerar`find` -1

Delsträngar

```
s = "Alfabet"
s[0:3] blir då "Alfa"
```

Genom att ange start- och slut-index inom hakar ([]) kan man få ut en del av en sträng.

Längden pa en sträng

Längden på sträng, dvs antalet bokstäver som ingår, kan man få fram med funktionen `len(s)`
Exempel: om s="Alfabet" så blir

```
len(s)
```

7

if / else

```
if <villkor>:
    <vad som ska göras om villkor är sant>
else:
    <vad som ska göras om villkor är falskt>
```

Används för att göra val i programmet. Villkor är t ex jämförelse mellan två tal (a < b).

Koden ska utföras *indenteras* dvs man flyttar in från kanten med mellanslag eller tab.

Villkor

==	Lika med, kan användas både till strängar och tal
!=	Inte lika med, kan användas både till strängar och tal
<	Mindre än
>	Större än
s in *sträng*	Finns *s* i *sträng*?

for

```
for c in msg:
   l = l + c
```

for kan användas för att gå igenom alla tecken i en sträng. För varje bokstav i *msg* så går programmet över innehållet i loopen och variabeln*c* tilldelas bokstäverna i *msg* i tur och ordning.

Dela upp en sträng i lista av ord

För att få en lista av ord från en sträng kan man använda `split` på strängen.
Exempel:

```
"All vägar bär till Rom".split()
```

blir då:

```
['All', 'vägar', 'bär', 'till', 'Rom']
```

Slå ihop en lista av ord till en sträng

För att slå ihop en lista av ord till en sträng så kan man använda `join` på listan.
Om:

```
l = [ "Alea", "jacta", "est" ]
```

och gör:

```
s = " ".join(l)
```

Så blir:

```
s = "Alea jacta est"
```

By **Filip Körling** (fkorling)
cheatography.com/fkorling/
patwic.com

Published 16th October, 2014.
Last updated 25th February, 2016.