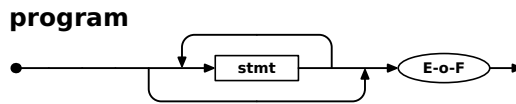


## 2.2 Asp-program

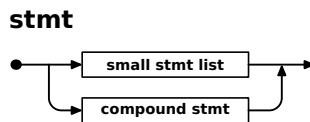
Som vist i figur 2.2, består et Asp-program av en sekvens av setninger (`<stmt>`). Tomme linjer er også lov. Symbolet E-o-f angir slutt på filen («end of file»).



Figur 2.2: Jernbanediagram for `<program>`

### 2.2.1 Setninger

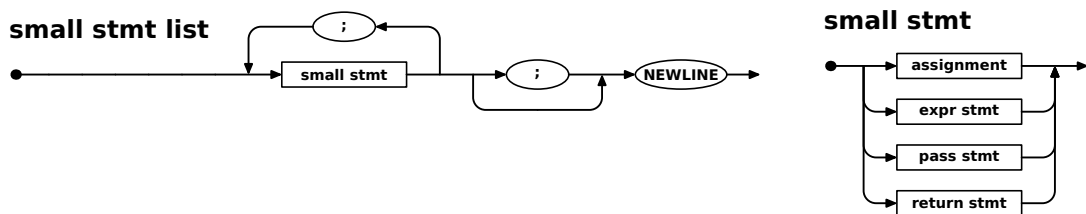
Figur 2.3 viser hva slags setninger man kan bruke i Asp.



Figur 2.3: Jernbanediagram for `<stmt>`

#### 2.2.1.1 Enkle setninger

Enkle setninger er gjerne korte og enkle, og da er det lov å ha flere på samme linje med semikolon mellom.



Figur 2.4: Jernbanediagram for `<small stmt list>` og `<small stmt>`

**Tilordning** Som i de fleste andre språk, brukes en tilordningssetning til å gi variabler en verdi. Siden Asp har *dynamisk typing*, skal ikke variablene deklarerer på forhånd. Les mer om dette i avsnitt 2.3.3 på side 27.

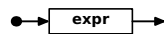


Figur 2.5: Jernbanediagram for `<assignment>`

**Uttrykk som setning** Et løserevet uttrykk er også en lovlig setning; dette er spesielt aktuelt når uttrykket er et funksjonskall.

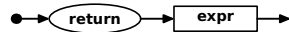
**Return-setninger** Return-setninger brukes til å avslutte utførelsen av en funksjon og angi en resultatverdi.

### expr stmt



**Figur 2.6:** Jernbanediagram for {expr stmt}

### return stmt



**Figur 2.7:** Jernbanediagram for {return stmt}

### pass stmt



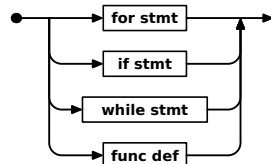
**Figur 2.8:** Jernbanediagram for {pass stmt}

**Pass-setninger** Pass-setninger gjør ingenting; de eksisterer bare for kunne settes der det kreves en setning uten at noe skal gjøres.

## 2.2.1.2 Sammensatte setninger

Sammensatte setninger inneholder andre setninger.

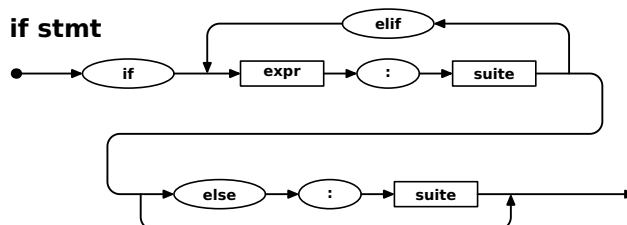
### compound stmt



**Figur 2.9:** Jernbanediagram for {compound stmt}

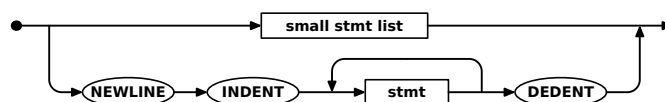
**If-setninger** If-setninger brukes til å velge om setninger skal utføres eller ikke. Se forøvrig avsnitt [2.3.1.1 på side 25](#) for hva som er lovlige testverdier.

### if stmt



**Figur 2.10:** Jernbanediagram for {if stmt}

### suite



**Figur 2.11:** Jernbanediagram for {suite}

**For-setninger** Denne formen for løkke går gjennom alle elementene i løkkekontrolluttrykket, som må være en liste.

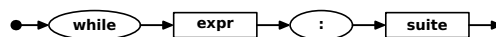
#### for stmt



Figur 2.12: Jernbanediagram for {for stmt}

**While-setninger** While-setninger er en annen form for løkkesetning i Asp. Se forøvrig avsnitt 2.3.1.1 på side 25 for hva som er lovlige testverdier.

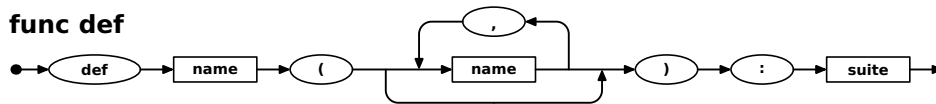
#### while stmt



Figur 2.13: Jernbanediagram for {while stmt}

**Funksjonsdeklarasjoner** I Asp regnes funksjonsdeklarasjoner som setninger.

#### func def

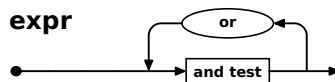


Figur 2.14: Jernbanediagram for {func def}

## 2.2.2 Uttrykk

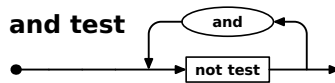
Et uttrykk beregner en verdi. Det er definert ved hjelp av ganske mange ikketerminaler for å sikre at presedensen<sup>2</sup> blir slik vi ønsker.

#### expr

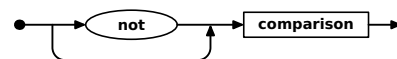


Figur 2.15: Jernbanediagram for {expr}

#### and test



#### not test

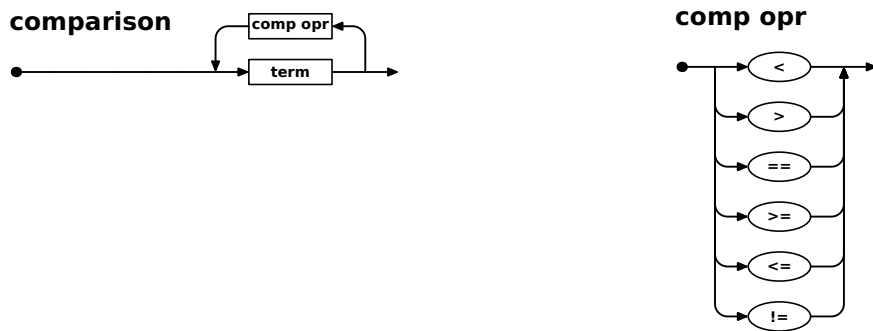


Figur 2.16: Jernbanediagram for {and test} og {not test}

<sup>2</sup> Operatører har ulik **presedens**, dvs at noen operatører binder sterkere enn andre. Når vi skriver for eksempel

$$a + b \times c$$

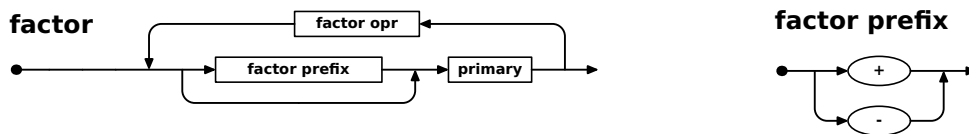
tolkes dette vanligvis som  $a + (b \times c)$  fordi  $\times$  normalt har høyere presedens enn  $+$ , dvs  $\times$  binder sterkere enn  $+$ .



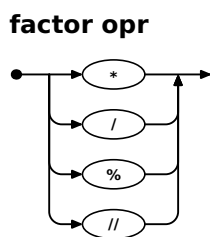
**Figur 2.17:** Jernbanediagram for  $\langle \text{comparison} \rangle$  og  $\langle \text{comp opr} \rangle$



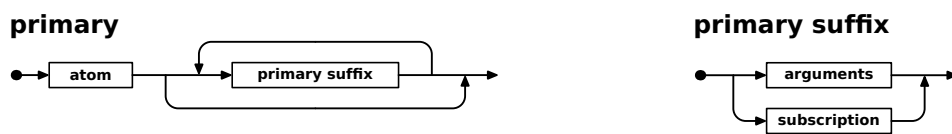
**Figur 2.18:** Jernbanediagram for  $\langle \text{term} \rangle$  og  $\langle \text{term opr} \rangle$



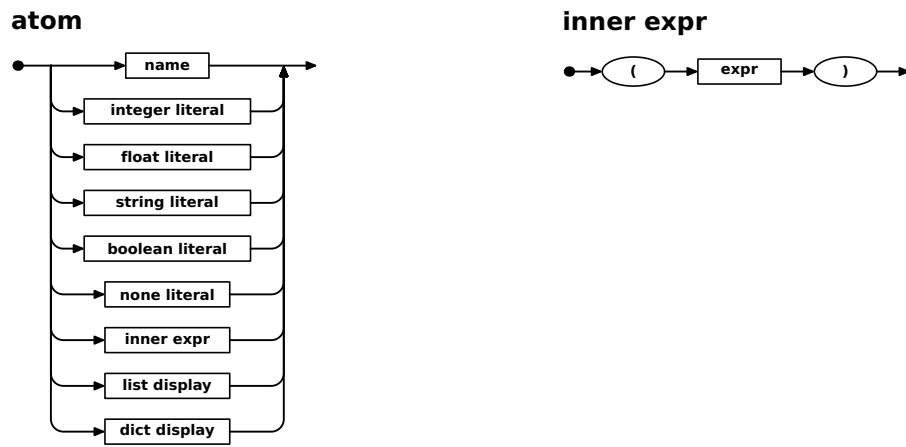
**Figur 2.19:** Jernbanediagram for  $\langle \text{factor} \rangle$  og  $\langle \text{factor prefix} \rangle$



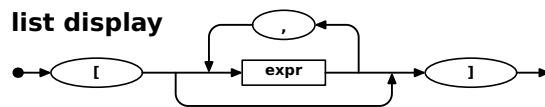
**Figur 2.20:** Jernbanediagram for  $\langle \text{factor opr} \rangle$



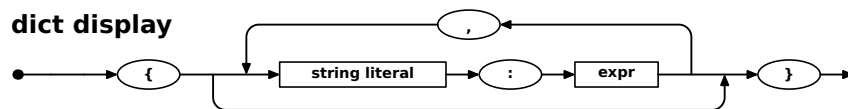
**Figur 2.21:** Jernbanediagram for  $\langle \text{primary} \rangle$  og  $\langle \text{primary suffix} \rangle$



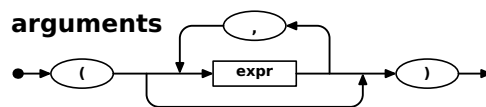
Figur 2.22: Jernbanediagram for {atom} og {inner expr}



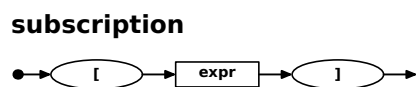
Figur 2.23: Jernbanediagram for {list display}



Figur 2.24: Jernbanediagram for {dict display}



Figur 2.25: Jernbanediagram for {arguments}

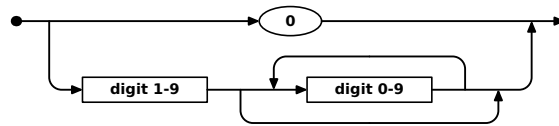


Figur 2.26: Jernbanediagram for {subscription}

### 2.2.2.1 Literaler

En **literal**<sup>3</sup> er et språkelement som angir en verdi; for eksempel angir «123» alltid heltallsverdien 123.

#### integer literal



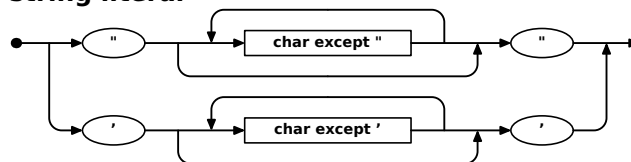
Figur 2.27: Jernbanediagram for {integer literal}

#### float literal



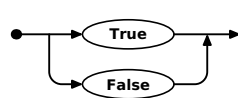
Figur 2.28: Jernbanediagram for {float literal}

#### string literal

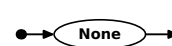


Figur 2.29: Jernbanediagram for {string literal}

#### boolean literal



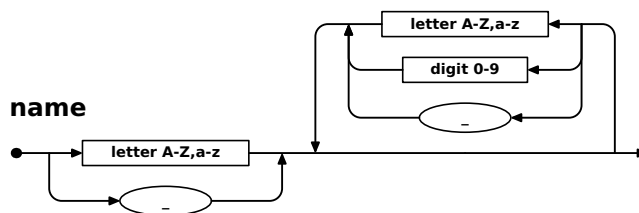
#### none literal



Figur 2.30: Jernbanediagram for {boolean literal} og {non literal}

### 2.2.2.2 Navn

I Asp benyttes navn til å identifisere variabler og funksjoner.



Figur 2.31: Jernbanediagram for {name}

<sup>3</sup> En **literal** er noe annet enn en **konstant**. En konstant er en navngitt verdi som ikke kan endres mens en literal angir verdien selv.

## 2.3 Spesielle ting i Asp

Noen konstruksjoner i Asp (og følgelig også i Python) kan virke uvante første gang man ser dem.

### 2.3.1 Typer

Tabell 2.1 gir en oversikt over hvilke typer data i Asp kan ha.

Type	Verdier	Eksempel
bool	Logiske verdier True og False	True
dict	Ordbok med verdier	{'Ja': 17, 'Nei': 0}
float	Flyt-tall	3.14159
func	Funksjoner	def f(): ...
int	Heltall	124
list	Liste av verdier	[1, 2, "Ja"]
none	«Ingenting»-verdien None	None
string	Tekster	"Abrakadabra"

**Tabell 2.1:** Typer i Asp

#### 2.3.1.1 Logiske verdier

Språket Asp har en logisk type med verdier True og False, men det er mye mer fleksibelt i hva det godtar som lovlige logiske verdier i if- og while-setninger eller i uttrykk. Tabell 2.2 angir hva som tillates som logiske verdier.

Type	False	True
bool	False	True
dict	{}	ikke-tomme ordbøker
float	0.0	alle andre verdier
int	0	alle andre verdier
list	[]	ikke-tomme lister
none	<b>None</b>	—
string	" "	alle andre tekststrenger

**Tabell 2.2:** Lovlige logiske verdier i Asp

Tabell 2.3 på side 28 viser at vi har de vanlige operatorene and, or og not for logiske verdier, men resultatet er litt uventet for and og or: de gir ikke svarene True eller False, men returnerer i stedet én av de to operandene, slik som dette:

```
"To be" or "not to be" ⇒ "To be"
"Yes" and 3.14          ⇒ 3.14
```