

# Tilordningsoperatoren



Gitt to variabler  $a$  og  $b$ , bytt om innholdet.

Tilordningsoperatoren =

Original verdi blir overskrevet.

# Tilordningsoperatoren



```
a = "hei"  
b = 3.14  
a = b  
b = a
```

	1	2	3	4
a				
b				

# Tilordningsoperatoren



```
a = "hei"  
b = 3.14  
a = b  
b = a
```

	1	2	3	4
a	"hei"			
b	?			

# Tilordningsoperatoren



```
a = "hei"  
b = 3.14  
a = b  
b = a
```

	1	2	3	4
a	"hei"	"hei"		
b	?	3.14		

# Tilordningsoperatoren



```
a = "hei"  
b = 3.14  
a = b  
b = a
```

	1	2	3	4
a	"hei"	"hei"	3.14	
b	?	3.14	3.14	

# Tilordningsoperatoren



Problem: Gammel verdi blir overskrevet. Vi kan bare endre en ting av gangen.

```
a = "hei"  
b = 3.14  
a = b  
b = a
```

	1	2	3	4
a	"hei"	"hei"	3.14	3.14
b	?	3.14	3.14	3.14

# Tilordningsoperatoren



## To vanlige løsninger

```
a = "hei"  
b = 3.14  
old_a = a    # ofte kalt temporary / temp  
a = b  
b = old_a
```

	1	2	3	4	5
a	"hei"	"hei"	"hei"	3.14	3.14
b	?	3.14	3.14	3.14	"hei"
a_old	?	?	"hei"	"hei"	"hei"

# Tilordningsoperatoren



## To vanlige løsninger

```
a = "hei"  
b = 3.14  
(b, a) = (a, b)  
# bruker tupler  
# kommer tilbake til dette senere
```



# Tilordningsoperatoren



Flere tilordninger etter hverandre

```
a = b = c = 3.14
```

# Tilordningsoperatoren



Flere tilordninger etter hverandre

*Triks:* Parenteser.

```
a = (b = (c = 3.14))  
# ikke riktig syntaks, kun for illustrere
```

## Flere tilordningsoperatorer



Finnes flere tilordningsoperatorer.

**VIKTIG:** Bortsett fra standard tilordning, må variabelen som tilordnes har en verdi fra før av.

Operator	Eksempel	Det samme som
=	a = 1	a = 1
--	a -= 1	a = a - 1
/=	a /= 1	a = a / 1
//=	a //= 1	a = a // 1
+=	a += 1	a = a + 1
*=	a *= 1	a = a * 1
%=	a %= 1	a = a % 1
**=	a **= 1	a = a**1