

Tupler

En tuple består *element* och varje element har en plats som anges av *index*. Det första elementet har index 0. Elementen kan vara av olika typ. Ej ändringsbar.

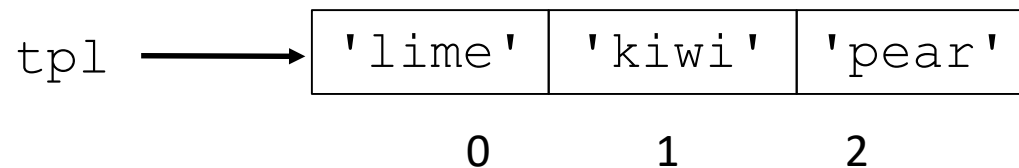
Ex.

```
tpl = ('lime', 'kiwi', 'pear') # Notera parenteser
print(tpl)                    # Skriv ut variabeln tpl
print(type(tpl))              # Skriv ut typen för variabeln tpl
```

Utskrift:

```
('lime', 'kiwi', 'pear')
<class 'tuple'>
```

Kan illustreras på följande sätt:



Variabeln `tpl` har tre element, `tpl[0]`, `tpl[1]`, `tpl[2]`, alla är av typen `str`, dvs strängar

Hantering av enskilda element:

```
tpl = ('lime', 'kiwi', 'pear')  
print(tpl[1])    # Skriv ut andra elementet i tpl  
print(tpl[-1])   # Skriv ut sista elementet i tpl  
tpl[0]= 'lemon'  # Uppdatera första elementet, men ...
```

Utskrift:

kiwi

pear

```
tpl[0]= 'lemon'
```

```
TypeError: 'tuple' object does not support item assignment
```

Element kan vara variabler:

```
f1 = 'lime'
```

```
f2 = 'kiwi'
```

```
f3 = 'pear'
```

```
tpl2 = (f1, f2, f3) # eller tpl2 = f1, f2, f3
```

Exempel: En lista med element som består av tuplar

```
pers = [('Kim',21), ('Robin',19), ('Agda',22)] # tre element  
print(pers)
```

Utskrift:

```
[('Kim', 21), ('Robin', 19), ('Agda', 22)]
```

```
for el in pers:           # Loopa igenom elementen (tuplarna)  
    print(el[0], el[1]) # Skriv ut elementen i tupeln
```

Utskrift:

```
Kim 21  
Robin 19  
Agda 22
```

Alt.

`print(el)` →

```
('Kim', 21)  
('Robin', 19)  
('Agda', 22)
```

```
pers = [('Kim',21), ('Robin',19), ('Agda',22)] # tre element
```

Kan även skriva:

```
for i, (name,age) in enumerate(pers):  
    print(i, name, age) # Skriv ut index, och tupeln
```

i är index i listan pers. (name,age) är en tupel

```
for tpl in enumerate(lst): # För varje tuple (index,element)  
    print(tpl) # Skriv ut tuple (parvärdet)
```

Utskrift:

```
(0, 4)  
(1, 16)  
(2, -7)
```

Exempel: Byta värde på två variabler mha tuple

```
a = 1    # En int variabel
b = 2
(b,a) = (a,b) # kan även skrivas: b, a = a, b
print(a,b)
```

Utskrift:

```
2 1
```

Exempel: Fibonacci talföljden mha tuple

0, 1, 1, 2, 5, 8, 13...

```
(a, b) = (0, 1)          # De två första talen: 0, 1
# Ovan kan även skrivas: a, b = 0, 1
while a < 13:
    print(a, end=' ')    # skriv ut, utan radbyte
    (a, b) = (b, a+b)    # a=1, b=0+1, etc
    # Ovan kan även skrivas: a, b = b, a+b
```

Utskrift:

```
0 1 1 2 3 5 8 13
```