1.1b Övriga elementära datatyper: sekvenser m.m.

Sekvenser: strängar, fält och tipplar

- En sträng är ett fält av tecken
- Ett fält (eller en lista) är en datamängd med ett antal element i en viss ordning
- En tippel (t.ex. ett par eller en trippel) liknar ett fält, men kan inte ändras när det väl har skapats
- Syntax i Python:
  - strängar med citattecken
  - fält med hakparenteser [,]
  - tipplar med parentestecken (,)

Exempel: sekvenser i Python

```
s = "Hejsan på dejsan!"
print(len(s))               # 17
print(s[0])                 # "H"
print(s[10])                # "d"
print(s.index("dejsan"))    # 10
print(s.upper())            # "HEJSAN PÅ DEJSAN"
tuple = s.partition("på")
print(tuple)                # ('Hejsan' ,'på', 'dejsan!')
print(tuple[2])             # "dejsan"
tuple[2] = "mejsan!"        # fel - kan inte ändra en tuppel!
```

Övriga datatyper: associativt fält (dictionary)

Ett associativt fält förknippar (associerar) ett värde till vart och ett av en uppsättning nycklar

```
dic = {"Sverige": "Stockholm", "Norge": "Oslo", "Finland": "Helsingfors"}
print(dic["Sverige"])               # "Stockholm"
print(dic["Norge"])                 # "Oslo"
print(dic["Finland"])               # "Helsingfors"
print(dic["Danmark"])               # KeyError
```

Oftast är nyckeln en textsträng, som t.ex. "Sverige", "Norge" och "Finland" ovan, men det behöver inte nödvändigtvis vara så.

```
dic = {1: "Foo", "Bar": "Baz", True: False}
print(dic[1])                       # "Foo"
print(dic["Bar"])                   # "Baz"
print(dic[True])                    # False
```

Användbara metoder:

```
dic.update({"Estland": "Tallinn"})  # lägga till ett nyckel-värdepar
dic.pop("Sverige")                  # ta bort nyckel-värdeparet med nyckeln "Sverige"
```

Övriga datatyper: Boolean

Antingen sann eller falsk

I likhet med i t.ex. JavaScript så är vissa värden (t.ex. tom sträng eller siffran noll) ”falsy”, dvs. de värderas falska i logiska sammanhang, medan andra är ”truthy”

```
bool = True or 0
print(bool)         # True
bool = True and 1
print(bool)         # 1
bool = not False or ""
print(bool)         # True
bool = False and ""
print(bool)         # False
bool = True and ""
print(bool)         # tom sträng
```

Övriga datatyper: mängd (set)

Endast ett av varje element, ingen särskild ordning

Lägg till ett element med metoden `add`

Kan vara användbar när man vill försäkra sig om att varje element i en datasamling är unikt, dvs. inga dubletter

```
s = set()
s.add("Äpple")
s.add("Banan")
s.add("Päron")
s.add("Äpple")      # Vi har redan ett äpple
print(s)            # {"Banan", "Päron", "Äpple"}
```

Idag: Övningar sekvenser, dictionary, boolean och set

Studera fler exempel från w3schools enligt länkar i lektionsanteckningarna.

1. Skriv ett program som översätter från svenska till rövarspråket.
2. Skriv ett program som skriver en inmatad text baklänges.
3. Utgå från det associativa fältet med länder och huvudstäder ovan. Använd metoden update för att lägga till nyckel-värdeparet {"Danmark": "Köpenhamn"}. Använd sedan metoden pop för att ta bort nyckel-värdeparet {"Finland": "Helsingfors"}.
4. Använd metoden union för att lägga ihop de två mängderna {"Banan", "Päron", "Äpple"} och {"Kiwi", "Ananas", "Päron"}. Hur många element finns i den resulterande mängden?

Jobba i övrigt med valfria uppgifter på Kattis idag

Försök pusha ditt arbete i slutet av lektionen till GitHub (fråga läraren om hjälp)