

Seminarium 04

Experimentering

Gustav Sörnäs

28 september 2021

Seminarieformen

Ibland läsa kod och diskutera frågor, ibland skriva kod i mindre grupper. Sedan diskussion i helklass.

Innan seminariet: läs förberedelsematerialet och försök er på uppgifterna.

Skicka in lösningar så vi kan diskutera i helklass:
`seminarium.sörnäs.se`. Anonymt, sålänge du inte skriver ditt namn i koden :>

Dagens seminarium

- ▶ Uppgift: listbyggare
- ▶ git
- ▶ Hjälp
- ▶ Dokumentation
- ▶ Uppgift: programutvecklingsprocessen

Listbyggare

Övning 404: Konstruera en funktion `distribute` som tar ett element och en lista av listor, och lägger in elementet i varje dellista.

```
def distribute(elem, lists):  
    distributed = []  
    for lst in lists:  
        distributed.append(lst + [elem])  
    return distributed  
  
def distribute(elem, lists):  
    return [lst + [elem] for lst in lists]
```

Listbyggare, men med färger

Övning 404: Konstruera en funktion `distribute` som tar ett element och en lista av listor, och lägger in elementet i varje dellista.

```
def distribute(elem, lists):  
    distributed = []  
    for lst in lists:  
        distributed.append(lst + [elem])  
    return distributed  
  
def distribute(elem, lists):  
    return [lst + [elem] for lst in lists]
```

Listbyggare, men med färger

Övning 404: Konstruera en funktion `distribute` som tar ett element och en lista av listor, och lägger in elementet i varje dellista.

```
def distribute(elem, lists):  
    distributed = []  
    for lst in lists:  
        distributed.append(lst + [elem])  
    return distributed  
  
def distribute(elem, lists):  
    return [lst + [elem] for lst in lists]
```

Listbyggare som ett filter

```
def filter_strings(values):  
    only_strings = []  
    for val in values:  
        if isinstance(val, str):  
            only_strings.append(val)  
    return only_strings
```

```
def filter_strings(values):  
    return [val for val in values if isinstance(val, str)]
```

Listbyggare som ett filter

```
def filter_strings(values):  
    only_strings = []  
    for val in values:  
        if isinstance(val, str):  
            only_strings.append(val)  
    return only_strings
```

```
def filter_strings(values):  
    return [val for val in values if isinstance(val, str)]
```


Listbyggare med två loopar

Övning 402: Konstruera en funktion `all_pairs` som utifrån en lista ger en lista med alla par, d.v.s. alla möjliga sätt att kombinera två element ur originallistan.

```
def all_pairs(values):  
    pairs = []  
    for val1 in values:  
        for val2 in values:  
            pairs.append((val1, val2))  
    return pairs
```

Listbyggare med två loopar

Övning 402: Konstruera en funktion `all_pairs` som utifrån en lista ger en lista med alla par, d.v.s. alla möjliga sätt att kombinera två element ur originallistan.

```
def all_pairs(values):  
    pairs = []  
    for val1 in values:  
        for val2 in values:  
            pairs.append((val1, val2))  
    return pairs
```

```
def all_pairs(values):  
    return [(val1, val2) for val2 in values for val1 in values]
```

Listbyggare – uppgift

1. Alla tal större än eller lika med noll. seminarium.sörnäs.se
[1, 4, -2, 3.3] => [1, 4, 3.3]
2. Näst minsta talet i varje lista. Input är en lista av listor (på 1 nivå, så ingen rekursion). Anta att det alltid finns minst två element.
[[1, 2, 3], [4, 6, 5, 7]] => [2, 5]
Tips: kolla på sorted.
3. Alla ord som innehåller minst ett a, men alla a:n är utbytta mot *.
["apa", "citron", "apelsin"] => ["*p*", "*pelsin"]
4. Alla tal 0–100 som är jämnt delbara med 3 eller 5, men *inte* båda.
=> [3, 5, 6, 9, 10, 12, 18, 20, ...]
5. En 5x5-matris fylld med nollor.
=> [[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], ... (x5 totalt)]
6. En 5x5-identitetsmatris, alltså en 5x5-matris med ettor på *diagonalen* och nollor överallt annars. (Gör sig bättre uppritad.)
=> [[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], ...]

git

git är i grunden ett verktyg för att spara olika versioner av hur filer såg ut vid olika tillfällen.

Hittills: add - commit - push.

- ▶ add: markera filer som "dom här vill jag spara".
- ▶ commit: spara dom markerade filerna i en *commit*.
- ▶ push: synkronisera mina commits med Gitlabs.

git - varför ni använder det

- ▶ Det blir tröttsamt att maila kod till sig själv.
- ▶ Arbeta parallellt på egen dator / LiU:s datorer.
- ▶ Arbeta två+ personer parallellt. (TDDE25)

git - några vanliga saker ni kan göra

Hur gör ni dom här sakerna med git? Ta reda på valfritt sätt!

- ▶ Visa en lista av alla commits.
- ▶ Kolla vilka ändringar som skedde för 3 commits sedan.
- ▶ Återställ en fil till hur den såg ut i en tidigare commit.
- ▶ Kasta bort ändringar sedan senaste commiten.
- ▶ Skapa en commit med ändringar från bara vissa filer, inte alla.

git - hur ni tar er vidare

Den officiella git-boken: <https://git-scm.com/book/en/v2>.
Från vanlig användning till hur det fungerar på insidan.

En git-föreläsning jag var med och höll i vintras:
<https://youtu.be/Db1XV8UTM1M>. Andra halvan inte lika
relevant (än).

Labbassistenten!

Hjälp

Vad gör ni om ni fastnar med något i kursen?

I vilken ordning?

Vad har ni tillgång till på tentan?

Dokumentationen

<https://docs.python.org>

<https://docs.opencv.org/2.4/modules/refman.html>

Programutvecklingsprocessen

1. Analys
2. Specifikation
3. Design
4. Implementation
5. Testning
6. Underhåll

1-3: Ni

4: Jag

5-6: Någon annan

Programutvecklingsprocessen – uppgiften

Tentauppgift från förra året.

Skriv en funktion som exekverar ett program i det påhittade språket PyASM.

Ett program består av en lista av instruktioner. En instruktion består av en tuple som innehåller en sträng och 1 eller fler argument.

Programmet kan under körningen använda ett antal *register* som är bokstäverna A–Z. Register innehåller heltal. Oanvända register har värdet 0.

Programutvecklingsprocessen – instruktionerna

- ▶ LOG <register> skriv ut värdet i registret.
- ▶ SET <register> <value> spara ett värde i ett register.
- ▶ CPY <register> <source> kopiera värdet från ett register till ett annat.
- ▶ ADD <register> <value> lägg till ett värde till ett register.
- ▶ MUL <register> <value> multiplicera ett register med ett värde och spara det i samma register.

PDF:er

<https://github.com/sornas/tdde23-seminars.git>