Mentimeter

Uke 5 - konsept

Gruppe 14

Kristin - kristlgj@ifi.uio.no

Mentimeter



Mentimeter

Plan for gruppetimen

- Recap av pensum + oppgaver
  - Prosedyrer med parametre
  - Funksjoner og returverdi
  - Innlesning fra fil
- Oppgaver (om vi rekker)

Mentimeter

Hva ble gjennomgått på forelesningen? Tenk selv i 2 min, så diskuter

- Hva lærte du, hva skjønte du ikke?
- Se gjerne over slides fra forelesningen om dere ikke husker!

Mentimeter

Læringsmål denne uken

- Ha kjennskap til parametre i prosedyrer og parameteroverføring
- Kjenne til funksjoner og kunne bruke de for å få unngå redundans og få mer strukturerte programmer
- Kjenne til innlesing fra og utskrift til fil, inkludert organisering av informasjon innad på linjer
- Kunne bruke filer, løkker, samlinger og funksjoner for å løse mer sammensatte problemer

Mentimeter

Hva mangler her?

```python
def hilse(navn, ukedag):
    print(f"Hei {navn}, ha en {ukedag}")

hilse("Kristin")
```

0	0	0	0	0
Første argument	Første parameter	Andre argument	Andre parameter	returverdi

Mentimeter

Prosedyrer med parametre

- til nå har prosedyrer kun gjort akkurat det samme hver gang
- mange ganger vil man gjøre samme funksjonalitet (f.eks. doble et tall), men med variasjon (forskjellige tall)
- det som varierer -> parametre

```
def prosedyre(parameter1, parameter2, ...):
    kodeblokk/kodelinjer

prosedyre(argument1, argument2, ...)
```

Mentimeter

Parametre er noe man tar inn til prosedyren så man kan bruke dem inni prosedyren (variabler)

Parameter og argument er to sider av det samme!

- Parameter: variabel i prosedyre som tar imot verdi
- Argument: verdi sendt inn når prosedyren kalles

```
def print_alder(alder):
    print(f"Du er {alder} år")

print_alder(5)
```

Mentimeter

Del 1 - Prosedyre med parameter

Lag en funksjon skriv_ut_tallkarakter() som tar inn en bokstav-karakter (A til F) og skriver ut tall-karakteren til terminalen. Bokstavkarakterer konverteres slik:

- A: 5.0
- B: 4.0
- C: 3.0
- D: 2.0
- E: 1.0
- F: 0.0

Mentimeter

Hva skrives ut?

```
tall1 = 5
tall2 = 10

def pluss(tall1, tall2):
    sum = tall1 + tall2
    return sum

print(pluss(3, 16))
```

0 ✗	0 ✗	0 ✗	0 ✗	0 ✓
15	16	3, 16	18	19

Mentimeter

Funksjoner

- funksjoner er prosedyrer med returverdi!
- kaller på funksjonen (ofte med argumenter) og får en verdi i retur/tilbake
- funksjonskallet får verdien til returverdien/det som returneres (så husk å bruke/lagre det)
- funksjonen er ferdig med å kjøre etter at den returnerer!

```
def funksjon(parameter1, parameter2, ...):
    kodeblokk/kodelinjer
    return returverdi

resultat = funksjon(argument1, argument2, ...)
```

Mentimeter

Diskuter

Hva må en funksjon returnere dersom vi vil bruke den i en if-sjekk eller som løkkebetingelse?

Mentimeter

Del 2 - Funksjon med parameter

Oppdater forrige prosedyre til å returnere flyttallet.

Lag en funksjon beregn_vekttall() som tar inn en bokstav-karakter og antall studiepoeng for et fag. Funksjonen skal returnere tall-karakteren ganget med antall studiepoeng.

Mentimeter

Prosedyrer vs. funksjoner?

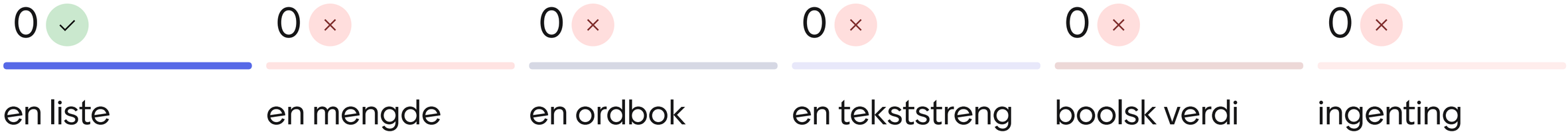
- helt lik syntaks
- begge kan være både med/uten parametre

- **prosedyrer: uten returverdi**
- **funksjon: med returverdi**

Prosedyrer er som servitører som ikke serverer (gir ingenting tilbake), mens funksjoner er servitører som faktisk serverer (gir noe tilbake)

Mentimeter

Hva får man når man bruker .split()?



Mentimeter

Hvilket modus er default når du åpner en fil?



Mentimeter

Innlesning fra fil

- tenk på en fil som en samling av linjer
  - for å lese inn en fil kan man derfor bruke en for-løkke og iterere gjennom linjene:)
- open() og navnet på filen inni parentesene for å åpne filen
- husk å lukke filen med .close() til slutt

```
fil = open("filnavn.txt")

for linje in fil:
    print(linje)

fil.close()
```

Mentimeter

- linjene har linjeskift (\n) etter seg - man kan bruke .strip() for å fjerne dem

```
fil = open("fil.txt")
for linje in fil:
    print(linje)
```

```
Linje 1

Linje 2

Linje 3
```

```
fil = open("fil.txt")
for linje in fil:
    print(linje.strip())
```

```
Linje 1
Linje 2
Linje 3
```

Mentimeter

Ofte har vi forskjellig data på hver linje og vil dele dem

- **.split()** -- for å splitte en linje på blanke tegn
- **.split(“,”)** -- for å splitte en linje på komma
  - (man splitter på det man skriver inn i parentesen)

Man får en liste med de forskjellige delene når man bruker split()

- disse delene kan man da hente ut ved vanlig liste-aksessering med indeks

Mentimeter

```
fil = open("fil.txt")

for linje in fil:
    biter = linje.split()
    linjenr = biter[0]
    ord = biter[1]

    print(f"På {linjenr} står det: {ord}")
```

```
1    linje1 hei
2    linje2 paa
3    linje3 deg
```

```
På linje1 står det: hei
På linje2 står det: paa
På linje3 står det: deg
```

Mentimeter

Del 3 - Lesing av fil

Ta utgangspunkt i en csv-fil der hver rad inneholder en fagkode, antall studiepoeng, og en bokstavkarakter.

Lag en prosedyre som tar inn navnet til filen, og som skriver ut informasjon om alle fagene, inkludert fagkode, antall studiepoeng, karakter, og vekttall på en ryddig og forståelig måte.

Mentimeter

Kan åpne filer i forskjellige moduser

- **open(“filnavn.txt”, “r”)**
  Read (default) - leser filen, feilmelding om filen ikke eksisterer
- **open(“filnavn.txt”, “w”)**
  Write - skriver til filen, oppretter filen om den ikke eksisterer
- **open(“filnavn.txt”, “a”)**
  Append - legger til på slutten av filen, oppretter filen om den ikke eksisterer
- **open(“filnavn.txt”, “x”)**
  Create - oppretter filen, feilmelding om filen eksisterer

(bruker .write(“tekst”) for å skrive til filen i Write og Append modus)

Mentimeter

Del 4 - Skriving til fil

Endre forrige prosedyre til å returnere en liste med den oversiktlige utskriften for hvert fag.

Lag en prosedyre *karakterutskrift()* som tar inn et filnavn å lese av, samt et filnavn å skrive til som parameter. Skriv én linje per fag til filen, som inkluderer fagkode, studiepoeng, bokstav-karakter, og vekttall.

Nederst i filen, skriv ut hvor mange studiepoeng studenten har totalt, og det vektede karaktersnittet.

Mentimeter

Oppgave 1

Skriv en funksjon som tar i mot to tall og returnerer det største av de to tallene.

Lagre tallet funksjonen returnerer i en variabel. Skriv ut variabelen.

Hva hadde stått i terminalen hvis du i a) hadde glemt return og så printet variabelen? Begrunn resultatet.

Mentimeter

Oppgave 2

Diskuter følgende spørsmål:

Hva skiller prosedyrer med parametre fra funksjoner med parametre?

Kan det fortsatt være en funksjon hvis den ikke har parametre?

Begrunn svaret i b). Hvorfor eller hvorfor ikke? Tenk tilbake på forelesningen og hva som skiller prosedyrer fra funksjoner.

Mentimeter

Oppgave 3

Skriv en funksjon som tar i mot to sannhetsverdier (True/False) og returnerer 1 hvis begge verdier er sanne, 0 hvis begge verdier ikke er sanne. Hva tar funksjonen ikke hensyn til (hva kan gå “galt”)?

Mentimeter

Skriv ned en ting som ble klarere og en ting du fortsatt sliter med:) Og tilbakemeldinger på gruppetimene!