Velkommen til gruppetime i IN1000 ☺

11. september 2020

Jessie Yue Guan

Planen for i dag

- Parametere og argumenter
- Returverdier
- For-løkker
- While-løkker

Prosedyrer og inndata/utdata

- Tenk på variabler som bokser som INNEHOLDER noe
  - «Noe» er altså en verdi, f.eks. en tekst, et heltall, et desimaltall, eller en boolsk verdi

---

- Tenk på prosedyrer som bokser som GJØR noe
  - «Noe» kan (men må ikke) være if-setninger, løkker, tilordning eller endringer av variabler, osv.

Prosedyrer og inndata/utdata

- Prosedyrer kan ta imot inndata (input) og/eller gi utdata (output)
  - Inndata = Du forteller boksen hvilke ting den skal utføre handlingene på
  - Utdata = Du forteller boksen hvilke ting handlingene skal resultere i

- Inndata for prosedyrer kalles for argumenter og parametere
- Utdata for prosedyrer kalles for returverdier

Prosedyrer uten parametere og argumenter

- Hittil har vi bare sett på prosedyrer som ikke tar imot inndata
- **def skriv_hilsen():**
  - **print("Hei!")**
- **skriv_hilsen()**

Prosedyrer med parametere og argumenter

- Men nå skal vi se litt på prosedyrer som faktisk gjør det

- **def skriv_hilsen(navn):**
  - **print("Hei, " + navn + "!")**

- **skriv_hilsen("Kong Harald")**

Når skal man bruke parametere og argumenter?

- Når man skal gjøre den samme tingene flere ganger med forskjellige variabler eller verdier

- **skriv_hilsen("Kong Harald")**
- **skriv_hilsen("Dronning Sonja")**
- **skriv_hilsen("Prins Haakon")**

Hva er forskjellen på parametere og argumenter?

- Parametere er det du skriver mellom parentesene når du definerer prosedyren
- **def skriv_hilsen(navn):** ← Parameter: navn
  - **print("Hei, " + navn + "!")**
- Argumenter er det du skriver mellom parentesene når du kaller på prosedyren
- **skriv_hilsen("Kong Harald")** ← Argument: "Kong Harald"

Prosedyrer uten returverdi

- Hittil har vi bare sett på prosedyrer uten utdata

- **def gange(a, b):**
  - **print(a*b)**

Prosedyrer med returverdi

- Men nå skal vi se litt på prosedyrer med utdata

- **def gange(a, b):**
  - **return a*b**

Når skal man bruke returverdier?

- Når du vil at en prosedyre skal resultere i en verdi
- Når du ikke vil printe ut denne verdien
- Når du ikke vil at brukeren skal se denne verdien
- Når du vil bruke denne verdien til noe senere

Eksempel

- def gange(a, b):
  - return a*b
- def halver(a):
  - return a/2
- def finn_areal_trekant(a, b):
  - print("Arealet av trekanten er:", halver(gange(a, b)))

Ulike kombinasjoner

- Ingen parametere eller returverdier
- **def velkommen():**
  - **print("Velkommen til programmet mitt!")**

- Parametere og returverdier
- **def summer(a, b)**
  - **return a + b**

- Parametere men ingen returverdier
- **def kvadrer(tall):**
  - **print(tall, "kvadrert er lik", tall**2)**

- Returverdier men ingen parametere
- **def kan_kjøre():**
  - **return hoy_nok and gammel_nok;**

Terminologi

- Prosedyre: Ingen returverdi
- Funksjon: Returverdi
- Metode: Tilhører en klasse
- Subrutiner: Samlebetegnelse på alle de tre ovenfor
- Alle disse kan ha ingen/en/flere parameter(e)
- Med andre ord, inndata påvirker ikke terminologien, men utdata gjør det

Oppgave 1

- Lag en prosedyre som tar imot to tall og skriver ut det største tallet

Oppgave 2

- Lag en funksjon som tar imot to tall og returnerer det største tallet

Oppgave 3

- Utfordringsoppgave
- Lag en prosedyre som lar brukeren skrive inn noen ord og skriver en historie ut av det

Når skal man bruke løkker?

- Print ut de 100 første kvadrattallene

Når skal man bruke løkker?

- Vi kan gjøre det manuelt…

- **print(1**2)**
- **print(2**2)**
- **print(3**3)**
- osv.

- Tar veldig lang tid og bruker veldig mye minne

Når skal man bruke løkker?

- Vi kan gjøre det ved hjelp av en for-løkke…
- **for i in range(0, 100):**
  - **print(i**2)**
- Tar veldig kort tid og bruker veldig lite minne, og er lett å kontrollere

Når skal man bruke løkker?

- Vi kan gjøre det ved hjelp av en while-løkke…

- **i = 0**
- **while (i < 100):**
  - **print(i**2)**
  - **i += 1**

- Tar veldig kort tid og bruker veldig lite minne, men er vanskelig å kontrollere

Når skal man bruke for-løkker?

- Når man skal gjøre den samme tingen om og om igjen *et visst antall ganger*
- For-løkker brukes ofte med strenger, lister, ordbøker, osv.

Når skal man bruke while-løkker?

- Når man skal gjøre den samme tingen om og om igjen *så lenge en betingelse er sann*
- While-løkker bruker ofte med conditions, if/elif/else-setninger, brukerinput, osv.

Oppskriften på en for-løkke

- for i in range(start, stopp, inkrement):
  - # gjør dette…

- For eksempel…
- **for x in range(0, 100, 5):**
  - **print(x, "poeng")**
- **for a in range(0, 100, 2):**
  - **print(a, "er et partall")**
- **for tall in range(0, 100, 7):**
  - **print(tall, "kan deles på 7")**

Oppskriften på en while-løkke

- while (betingelse):
  - # gjør dette...

- For eksempel…
- **while (i < 100):**
  - **i += 10**
- **while (bruker_input.isdigit() == False):**
  - **brukerinput = input("Skriv inn et tall: ")**
- **while (tips < 0):**
  - **tips = 0**

Oppgave 4

- **i = 1**
- **while (i <= 10):**
  - **i += 2**
- **print(i)**
- Hva printes ut her?

- 21

Oppgave 5

- **tekst = ["hadet", "på", "badet", "din", "gamle", "sjokolade"]**
- **indeks = 0**

- **for indeks in range(0, 6, 2):**
  - **print(tekst[indeks])**

- Hva printes ut her?

- hadet
- badet
- gamle

Oppgave 6

- **a = 10**
- **b = 1**

- **while a > 0:**
  - **b = b * 2**
  - **a = a - b**

- **print("a =", a)**
- **print("b =", b)**

- Hva printes ut her?

- a = -4
- b = 8

Oppgave 7

- Lag et program som spør brukeren hvem som er best
- Fortsett å spørre helt til brukeren skriver inn navnet ditt

Oppgave 8

- Lag en liste med tilfeldige tall
- Print ut alle oddetallene i denne lista

På tide å sjekke om dere har fulgt med…

- Kahoot! :D