

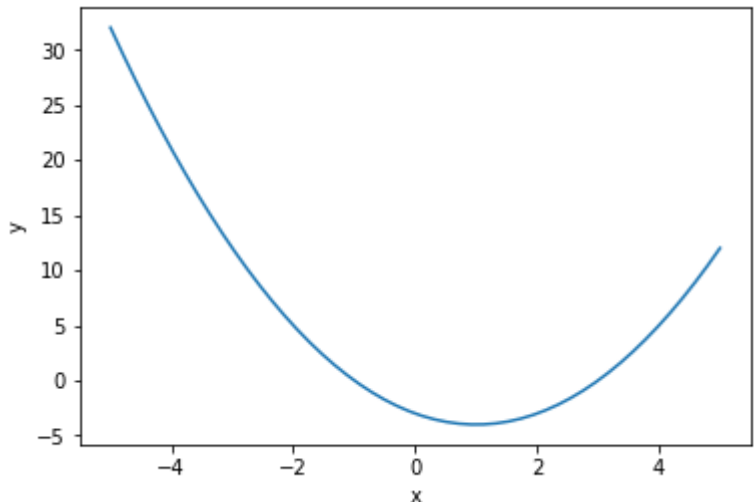
Fem små kodeeksempler i Python

| Kodeeksempel | Trinnvis forklaring av kode |
|---|---|
| <p>Eksempel 1 Hente data, levere data</p> <pre>1 #Program som henter input og gir output 2 3 navn = input("Hva heter du? ") 4 print("Hei " , navn) 5 print("nå skal vi summere to tall:") 6 t1=int(input("Velg det første tallet: ")) 7 t2=int(input("Velg det andre tallet: ")) 8 print("Summen av tallene dine er: " , t1+t2)</pre> | <p>Linje 1: Når linjen starter med # er det ingen kode, men en forklarende tekst</p> <p>Linje 3: Her er «navn» en variabel som tilordnes den verdien brukeren taster inn. «input» er syntaks som forteller at programmet skal vente på en verdi og motta denne. Input må kobles til variabel for å gi mening.</p> <p>Linje 4: «print» er kommandoen for å skrive verdi til skjerm. Verdi kan være tekst omgitt av «», verdien til en variabel eller en verdi som regnes ut på bakgrunn av flere variabler.</p> <p>Linje 5: skriver beskjed til bruker.</p> <p>Linje 6: ber om å få et tall. «int» forteller at det skal rundes til heltall.</p> <p>Linje 7: ber om neste tall. Bruker ny variabel, ellers overskrives forrige tall.</p> <p>Linje 8: Skriver ut tekst og summen av tallene</p> |
| Resultat | Kommentarer |
| <pre>Hva heter du? Grete Hei Grete nå skal vi summere to tall: Velg det første tallet: 63 Velg det andre tallet: 124 Summen av tallene dine er: 187</pre> | <p>En typisk nybegynnerfeil kan være å skrive syntaks med stor bokstav eller tastefeil, glemme parenteser eller anførselstegn i par. Slikt gjør at programmet feiler.</p> <p>Her brukes datatypene tekst og heltall. Andre vanlige datatyper er:</p> <ul style="list-style-type: none">• float – desimaltall (Bruk <i>punktum</i> som desimalskilletegn)• string – omdefinerer tall til tekst. Eks: a = string (17). Ser fortsatt ut som 17, men er tekst og kan derfor ikke brukes i beregninger. <p>Hvis vi har bruk for summen senere, kan vi legge inn linja sum = t1 + t2 før linje 8, og printet sum i stedet for t1+t2 i stedet.</p> |

| Kodeeksempel | Trinnvis forklaring av kode |
|---|--|
| <p>Eksempel 2 Gjøre valg med if-setning</p> <pre data-bbox="203 347 1106 608"> 1 #Eksempel på if-setning - valg av morgenhilsen 2 3 svar=input("Hei, har du en god morgen? Svar J eller N: ") 4 if svar == "J": 5 print("Bra! Gi et smil, og du får 1000 tilbake") 6 elif svar == "N": 7 print("Så leit, håper det snart blir bedre!") 8 else: 9 print("Du svarte noe jeg ikke forstår, ønsker deg en god dag!") </pre> | <p>Linje 3: Ber om et tegn</p> <p>Linje 4, 6 og 8: Her splittes handlingen i tre alternativer, avhengig av input fra bruker. Slik lat vi programmet gi ulik respons, avhengig av svar.</p> <p>Vi kan bruke så mange elif som vi ønsker. For hver if og elif oppgis et premiss. Se bruk av ==. Dette betyr at vi ikke gir variabelen en verdi, men at vi sjekker om et premiss er oppfylt.</p> |
| Resultat | Kommentarer |
| <pre data-bbox="203 810 913 879"> Hei, har du en god morgen? Svar J eller N: J Bra! Gi et smil, og du får 1000 tilbake. </pre> | <p>Hvis bruker vil svar J, men f.eks. svarer j , Ja, eller ja, kjenner ikke dette programmet igjen svaret i linje 4 eller 6, og vil derfor velge alternativet i linje 8. En forbedring av programmet vil være å gjøre det mer brukervennlig med tanke på mulige synonyme svar.</p> <p>Til slik bruk kan det være nyttig med logiske operatorer: or betyr eller, and betyr og, og i tillegg til == som betyr er lik, kan vi bruke: != for ikke lik, < for mindre enn, > for større enn, <= for mindre eller lik og >= for større eller lik.</p> <p>Legg merke til innrykket som forteller editoren at printlinjene hører inn under if-linjene. Å unnlate innrykket gir syntaksfeil. Like galt går det hvis vi glemmer kolon på slutten av if-linjene.</p> |

| Kodeeksempel | Trinnvis forklaring av kode |
|---|--|
| <p>Eksempel 3 Gjøre beregninger i while-løkke</p> <pre data-bbox="203 347 1122 550"> 1 #Programmet teller nedover med 2 fra et oppgitt heltall i et ønsket antall trinn 2 3 verdi=int(input("Hviket heltall skal vi starte på? Skriv tall og tast enter. ")) 4 antall=int(input("Hvor mange tall vil du ha i nedtellingen? ")) 5 while antall>0: 6 print(verdi) 7 verdi -= 2 8 antall -= 1 </pre> | <p>Programmet skal telle nedover med 2 fra et utgangspunkt som brukeren oppgir. Brukeren skal også oppgi hvor mange tall som skal komme i rekka.</p> <p>En while-løkke gjentas så lenge en gitt betingelse er sann.</p> <p>Linje 3: Henter inn starttallet fra bruker, legger i variabelen «verdi».</p> <p>Linje 4: Henter opplysning om antallet tall i nedtellingen.</p> <p>Linje 5: while <premiss>: Så lenge premisset er sant, gjennomføres linjene 6-8. De som er innrykket.</p> <p>Linje 6: Skriver ut innholdet i «verdi».</p> <p>Linje 7: Reduserer verdien av «verdi» med 2. Les: fra verdi trekkes et tall lik 2.</p> <p>Linje 8: Reduserer verdien av «antall» med 1 før premiss i linje 5 sjekkes på nytt.</p> |
| Resultat | Kommentarer |
| <pre data-bbox="203 869 1093 1085"> Hviket heltall skal vi starte på? Skriv tall og tast enter. 215 Hvor mange tall vil du ha i nedtellingen? 4 215 213 211 209 </pre> | <p>Hvis du glemmer å legge inn i løkken noe som endrer premisset for løkken, kan du ende med å gå inn i en evig løkke.</p> <p>Når bruker her velger 4 tall i nedtellingen, vil løkken gjennomføres når antall ==4, 3, 2 og 1, men når antall ==0, er ikke lenger premisset oppfylt og programmet avsluttes (eller hadde fortsatt på linje 9, om det var noen). Merk at innrykkene definerer hva som er innhold i løkka. En linje 9 ved marg, viser at programmet fortsetter der etter fullføring av løkka.</p> <p>Linje 7: her kan vi også bruke syntaksen <code>verdi = verdi - 2</code> . Det vil ha samme betydning</p> |

| Kodeeksempel | Trinnvis forklaring av kode |
|--|--|
| <p>Eksempel 4 Gjøre beregninger i for-løkke</p> <pre data-bbox="203 347 1099 475"> 1 #Program som finner et valgt antall trekantall 2 antall=int(input("Hvor mange trekantall vil du ha? ")) 3 for i in range(1,antall+1,1): 4 print(i*(i+1)/2) </pre> | <p>Her bruker vi en for-løkke til å regne ut et antall trekantall. En for-løkke gjentas et forhåndsbestemt antall ganger.</p> <p>Linje 2: Innhenter opplysning fra bruker om hvor mange trekantall som skal beregnes.</p> <p>Linje 3: Definerer for-løkke med følgende syntaks:</p> <ul style="list-style-type: none"> • for <variabel> in range(<fra og med>, <til>, <steglengde>) <ul style="list-style-type: none"> ○ Linje 4: rykkes inn fordi den tilhører løkka. Beregner og skriver ut trekantall. |
| Resultat | Kommentarer |
| <pre data-bbox="203 833 770 1082"> Hvor mange trekantall vil du ha? 6 1.0 3.0 6.0 10.0 15.0 21.0 </pre> | <p>Jeg velger å starte på 1, fordi vi vanlig vil ikke definerer 0 som et trekantall. Siden variabelverdi går stegvis med 1 fra og med 1 og til et tall, må til være antall + 1 for å få med et riktig antall runder i løkka. Beregningen starter altså med i =1, bruker formelen og returnerer 1, øker i med 1 til 2, bruker formelen og returnerer 3, øker i med 1 til 3 ... osv til dette er gjort 6 ganger, slik brukeren ba om i linje 2.</p> <p>Når løkka er føllført, avsluttes programmet, eller det kunne fortsatt i linje 5, dersom denne hadde vært der.</p> <p>Linje 3: Strengt tatt trenger vi ikke oppgi <fra og med> eller <strelengde> i dette tilfellet, etter som disse parameterne har defaultverdi 1.</p> |

| Kodeeksempel | Trinnvis forklaring av kode |
|---|---|
| <p>Eksempel 5 Tegne graf til funksjon</p> <pre data-bbox="212 347 672 622"> 1 # Tegne grafen til en funksjon 2 3 from pylab import * 4 x=linspace(5 , -5) 5 y= x**2 -2*x - 3 6 plot(x,y) 7 xlabel("x") 8 ylabel("y") 9 show()</pre> | <p>Merk! Du må ha python3 for at dette skal virke.</p> <p>Linje3: Henter funksjonalitet fra et bibliotek av forhåndsprogrammerte prosedyrer som gjør graftegning mulig. Bruk av * betyr at all funksjonalitet hentes.</p> <p>Linje 4: oppgir det intervallet av x-verdier der grafen skal tegnes. Kan ofte være definisjonsmengden til funksjonen.</p> <p>Linje 5: Definerer funksjonsuttrykket, altså hvordan y beregnes ut fra x-verdier. Her en andregradsfunksjon.</p> <p>Linje 6: Ber om å få tegnet alle tallpar (x,y) i det aktuelle intervallet.</p> <p>Linje 7: Setter merkelapp på x-aksen.</p> <p>Linje 8: Setter merkelapp på y-aksen.</p> |
| Resultat | Kommentarer |
|  | <p>Legg merke til aritmetiske operatorer:</p> <ul style="list-style-type: none"> + Addisjon - Subtraksjon * multiplikasjon / divisjon ** opphøyet i % modulus (rest ved divisjon) // heltallsdivisjon <p>Merk; her vil det gå galt å skrive 2x i førstegradsleddet. Python vil lete etter variabelen 2x uten å finne denne, vi må bruke 2*x. Python forholder seg ellers til standarder for regnerekkefølge og parentesbruk i regneuttrykk.</p> |