SAMMEN FRA FØRSTE LINJE

Bygg din egen brøkkalkulator

Et arbeidshefte for deg som vil lære å hente inn svar, bruke variabler og la Python velge mellom pluss og minus.

Slik jobber dere

1. Skriv koden i ett eksempel om gangen. Erstatt den gamle koden når dere starter et nytt steg.

2. Gjett hva programmet vil gjøre. Trykk så Run og prøv.

3. Når programmet spør, skriver du svaret i konsollen og trykker Enter. Svar på ett spørsmål om gangen.

4. Gjør den lille oppgaven. Forklar med egne ord før dere går videre.

Python i nettleseren

Åpne Programiz sin Python-editor. Adressen er klikkbar:

programiz.com/python-programming/online-compiler/

Skriv koden i kodefeltet. Trykk Run. Klikk i feltet for utdata/konsoll når programmet venter på svar. Nettjenestens utseende kan endre seg.

Ta det i små økter

Begynn med brøkkapitlet på de neste sidene. Regn og tegn på papir. Gå deretter til Python: steg 1-3, så steg 4-6, og til slutt steg 7-8. Ta korte økter på 15-25 minutter, eller stopp tidligere.

Vi lager først et program med tekst. Et vindu med knapper kan komme senere. Ingen av eksemplene trenger ekstra Python-pakker.

Hva forteller en brøk?

Før vi programmerer, skal vi forstå regningen. Finn frem papir og blyant. Bruk like store rektangler som hele når du tegner brøkene.

En hel, delt i like store deler

I 3/4 er 4 nevneren: den hele er delt i fire like store deler. 3 er telleren: vi har tre av delene. Vi leser brøken som tre firedeler.

3/4: tre av fire like store deler



1/4: én av fire like store deler



Nevneren sier hvilken størrelse delene har. Telleren teller delene. Flere deler i samme hele betyr at hver del blir mindre: en åttedel er mindre enn en firedel.

Samme hele er viktig

Når vi sammenligner tegningene, må de hele være like store. En halv liten pizza og en halv stor pizza er ikke like mye mat.

Prøv selv

Tegn to like store rektangler. Del det ene i fire like deler og fargelegg én. Del det andre i åtte like deler og fargelegg én. Hvilken fargelagt del er størst?

Stopp og forklar

Pek på teller og nevner i 2/5. Hva betyr hvert av tallene? Hvorfor kan nevneren ikke være 0?

En nevner på 0 ville bety deling på null. Det er ikke definert i vanlig brøkregning.

Legg sammen like store deler

Tenk at du har 2/7 av en sjokolade og får 3/7 til. Alle bitene er sjudeler. Vi teller hvor mange sjudeler vi har til sammen.

Du har 2/7



Du får 3/7 til



Til sammen: 5/7



Steg for steg

1. Sjekk nevnerne: begge er 7, så delene er like store.
2. Legg sammen tellerne: 2 + 3 = 5.
3. Behold nevneren: svaret er 5/7.

```
2/7 + 3/7 = (2 + 3)/7 = 5/7
```

Vi legger ikke sammen nevnerne. 5/14 ville betydd fem fjortendeler, som er mindre biter enn dem vi startet med.

Prøv selv

A: 1/6 + 3/6. B: 2/5 + 3/5. Tegn minst ett av regnestykkene. Hva betyr det når telleren blir like stor som nevneren?

Stopp og forklar

Forklar hvorfor 1/4 + 1/4 blir 2/4, og ikke 2/8.

Trekk fra like store deler

Du har 5/8 av en sjokolade og gir bort 2/8. Bitene er fortsatt åttedeler. Vi trekker fra antall biter vi gir bort.

Før: 5/8



Du gir bort 2/8



Igjen: 3/8



Steg for steg

1. Sjekk nevnerne: begge er 8.
2. Trekk den andre telleren fra den første: 5 - 2 = 3.
3. Behold nevneren: svaret er 3/8.

```
5/8 − 2/8 = (5 − 2)/8 = 3/8
```

Rekkefølgen betyr noe

5/8 - 2/8 er ikke det samme som 2/8 - 5/8. Det siste blir -3/8. Det kan vi tenke på som at vi mangler tre åttedeler. På en tallinje ligger svaret til venstre for 0.

Prøv selv

C: 6/7 - 2/7. D: 3/5 - 3/5. Tegn og stryk ut delene du tar bort. Sjekk C ved å legge 2/7 tilbake til svaret.

Stopp og forklar

Hva skjer med nevneren når du trekker fra? Hvorfor?

Gi delene samme størrelse

Vi kan ikke telle halvdeler og firedeler som om de var like store biter. Først må vi skrive brøkene med en felles nevner.

1/2 av den hele



2/4 av den samme hele



Det fargelagte området er like stort. 1/2 og 2/4 er to navn på samme verdi. Vi har delt hver halvdel i to mindre deler.

Utvid en brøk

```
1/2 = (1 * 2)/(2 * 2) = 2/4
```

Vi ganger teller og nevner med det samme tallet. Da beholder brøken verdien. Her bruker vi * som gangetegn, akkurat som i Python.

Eksempel: 1/2 + 1/4

1. Velg felles nevner 4. Den passer for begge brøkene.
2. Gjør om 1/2 til 2/4. Brøken 1/4 er allerede klar.
3. Legg sammen tellerne og behold nevneren.

```
1/2 + 1/4 = 2/4 + 1/4 = 3/4
```

Prøv selv

E: Gjør om 1/3 til sjettedeler. F: Regn ut 1/3 + 1/6. Tegn hvis du er usikker.

Stopp og forklar

Hvorfor blir det feil å gange bare nevneren når vi utvider?

Pluss og minus med ulike nevnere

Pluss: 1/2 + 1/3

Se på gangetabellene: 2, 4, 6 ... og 3, 6 ... Begge treffer 6. Vi bruker 6 som felles nevner.

```
1/2 = (1 * 3)/(2 * 3) = 3/6
1/3 = (1 * 2)/(3 * 2) = 2/6

1/2 + 1/3 = 3/6 + 2/6 = 5/6
```

Minus: 3/4 - 1/6

4, 8, 12 ... og 6, 12 ... Begge treffer 12. Først utvider vi begge brøkene. Så trekker vi fra tellerne i samme rekkefølge som før.

```
3/4 = (3 * 3)/(4 * 3) = 9/12
1/6 = (1 * 2)/(6 * 2) = 2/12

3/4 - 1/6 = 9/12 - 2/12 = 7/12
```

En felles nevner som alltid fungerer

Du kan alltid gange de to positive nevnerne. For 4 og 6 får du 24. Det fungerer også: 18/24 - 4/24 = 14/24 = 7/12. Du trenger ikke den minste felles nevneren for å få riktig svar.

Huskeregel for begge regnearter

Like nevnere først. Deretter pluss eller minus mellom tellerne. Behold felles nevner. Se til slutt om svaret kan forkortes.

Prøv selv

G: 1/2 + 1/5. H: 5/6 - 1/3. Vis hva du ganger med når du utvider.

Samme verdi, enklere svar

Å forkorte er å dele teller og nevner på det samme tallet, større enn 1. Verdien endrer seg ikke. Du samler små deler til større deler.

```
2/8 = (2 : 2)/(8 : 2) = 1/4
6/9 = (6 : 3)/(9 : 3) = 2/3
```

Her betyr : deling på papir. I Python bruker vi andre tegn for deling. Vi venter med å programmere forkorting til vi er trygge på pluss og minus.

Eksempel med forkorting

```
3/4 − 1/2 = 3/4 − 2/4 = 1/4

Med nevnerproduktet 4 ∗ 2 får vi også:
6/8 − 4/8 = 2/8 = 1/4
```

Begge fremgangsmåtene gir samme verdi. Et svar som 2/8 er riktig selv om 1/4 er enklere.

Regn selv før dere blar videre

I: 1/4 + 1/4. J: 3/4 - 1/2. K: 2/3 + 1/6. L: 5/6 - 1/4.

For hvert regnestykke: velg felles nevner, skriv om brøkene, regn med tellerne og forkort hvis du kan. Sjekk om svaret virker rimelig.

Finn feilen

M: En elev sier at 1/2 + 1/3 = 2/5. Forklar feilen. Hint: Når du legger noe positivt til 1/2, må svaret bli større enn 1/2. Er 2/5 det?

Stopp og forklar

Kan du forklare hele fremgangsmåten uten å se på eksemplene? Bruk 1/2 + 1/3 som eksempel.

Fra brøkregning til Python

Fasit til oppgavene

A: 4/6 = 2/3. B: 5/5 = 1 hel. C: 4/7. D: 0/5 = 0.

E: 1/3 = 2/6. F: 2/6 + 1/6 = 3/6 = 1/2.

G: 5/10 + 2/10 = 7/10. H: 5/6 - 2/6 = 3/6 = 1/2.

I: 2/4 = 1/2. J: 3/4 - 2/4 = 1/4.

K: 4/6 + 1/6 = 5/6. L: 10/12 - 3/12 = 7/12.

M: Eleven har lagt sammen både tellerne og nevnerne. Først må brøkene få lik nevner: 3/6 + 2/6 = 5/6. 2/5 er mindre enn 1/2, og kan derfor ikke være riktig svar.

Dette skal vi lære Python å gjøre

Vi kan selv følge en oppskrift. Nå skal vi skrive den samme oppskriften slik at datamaskinen kan følge den.

1. Spør hvilke brøker og hvilken regneart vi vil bruke: input().

2. Ta vare på tallene og valget: variabler.

3. Finn like store deler: gang nevnerne og utvid tellerne.

4. Velg pluss eller minus: if og elif.

5. Vis regnestykket og resultatet: print().

Vi begynner med det enkleste

På neste side lager vi bare en inputhenter og skriver ut valget. Så bygger vi på litt etter litt. Brøkregningen kommer inn når vi kan hente tall og velge regneart.

Til den voksne: La barnet forklare regnemåten først. Når koden senere gir for eksempel 2/8, bruk dette kapitlet til å vise at svaret har samme verdi som 1/4.

Hva valgte du?

Først skal programmet bare hente et svar og skrive det ut. Det skal ikke regne ennå.

```
valg = input("Vil du bruke pluss eller minus? ")
print("Du valgte:", valg)
```

Dette skjer

input() viser spørsmålet og venter på svaret ditt. Variabelen valg tar vare på svaret. print() skriver det på skjermen.

Test sammen

Skriv pluss og trykk Enter. Da skal du få:

```
Du valgte: pluss
```

Prøv selv

Kjør igjen og svar minus. Prøv deretter pizza. Endre teksten i spørsmålet og kjør på nytt.

Stopp og forklar

Hvor lagres svaret ditt? Vet programmet hva pluss betyr ennå?

Tips til den voksne: La barnet peke på input, valg og print mens han forklarer.

Husk flere svar

Nå vil vi hente to tall og en regneart. Foreløpig behandler Python alle svarene som tekst.

```
forste = input("Første tall: ")
andre = input("Andre tall: ")
valg = input("Velg + eller –: ")

print("Første tall:", forste)
print("Andre tall:", andre)
print("Regneart:", valg)
print("Du skrev:", forste, valg, andre)
```

Test sammen

Svar 8, så 3, så -. Den siste linjen skal bli:

```
Du skrev: 8 – 3
```

Et variabelnavn skrives uten anførselstegn. Teksten "Du skrev:" står i anførselstegn fordi den skal vises akkurat slik.

Prøv selv

Bytt navn fra forste til tall1 begge steder der variabelen brukes. Virker programmet fortsatt?

Stopp og forklar

Hva er forskjellen på print("valg") og print(valg)? Prøv begge.

Hvorfor blir 2 + 3 til 23?

Skriv og kjør denne koden. Svar 2 og 3. Gjett resultatet først!

```
forste = input("Første tall: ")
andre = input("Andre tall: ")
print(forste + andre)
```

Resultatet blir 23. input() gir tekst. Pluss mellom to tekster setter dem sammen.

Gjør teksten om til heltall

```
forste = int(input("Første tall: "))
andre = int(input("Andre tall: "))
svar = forste + andre
print("Svaret er:", svar)
```

int() gjør teksten om til et heltall. Nå blir svaret 5. Variabelen svar lagrer resultatet av regningen.

Prøv selv

Prøv 8 og 4. Endre + til - og kjør igjen. Du skal få 4.

Stopp og forklar

Hvorfor trenger vi int()? Hva lagres i svar?

Skriv bare heltall her, for eksempel 4 eller -2. Bokstaver eller 1/2 gir en feilmelding fra int(). Det er noe vi kan lære å håndtere senere.

La Python velge

Nå skal programmet bruke regnearten du skriver inn.

```
forste = int(input("Første tall: "))
andre = int(input("Andre tall: "))
valg = input("Velg + eller –: ")

if valg == "+":
    svar = forste + andre
    print("Svaret er:", svar)
elif valg == "–":
    svar = forste – andre
    print("Svaret er:", svar)
else:
    print("Du må velge + eller –.")
```

if betyr hvis. elif betyr ellers, hvis. else betyr ellers. Python tester ovenfra og kjører den første delen som passer.

= gir en variabel en verdi. == spør om to verdier er like. Kolonet og innrykket viser hvilke linjer som hører til valget. Bruk fire mellomrom i innrykket.

Prøv selv

Kjør med 8, 3 og +. Kjør så med 8, 3 og -. Prøv til slutt 8, 3 og pizza. Du skal få 11, 5 og en beskjed om ugyldig valg.

Stopp og forklar

Hvilke linjer kjører når du velger minus? Hva gjør else?

Teller og nevner

I brøken 2/7 er 2 telleren og 7 nevneren. Nevneren forteller hvor mange like deler en hel er delt i. Telleren forteller hvor mange slike deler vi har.

Med samme nevner

```
2/7 + 3/7 = 5/7
5/8 − 2/8 = 3/8
```

Vi legger sammen eller trekker fra tellerne. Nevneren beholdes fordi delene fortsatt er like store.

Hent brøkene før vi regner

```
nevner = int(input("Felles nevner: "))
teller1 = int(input("Første teller: "))
teller2 = int(input("Andre teller: "))
valg = input("Velg + eller −: ")

print("Du valgte:", teller1, "/", nevner,
      valg, teller2, "/", nevner)
```

Svar 7, 2, 3 og +. Programmet skal vise 2 / 7 + 3 / 7. Vi bruker to linjer til print() for å gjøre koden lettere å lese.

Prøv selv

Vis regnestykket 5/8 - 2/8. Hvilke fire svar må du skrive, i hvilken rekkefølge?

Stopp og forklar

Hvorfor henter vi bare én nevner i dette programmet?

Regn med samme nevner

Nå setter vi sammen input, variabler og valg. Skriv hele programmet under.

```
nevner = int(input("Felles nevner: "))
teller1 = int(input("Første teller: "))
teller2 = int(input("Andre teller: "))
valg = input("Velg + eller –: ")

print("Du valgte:", teller1, "/", nevner,
      valg, teller2, "/", nevner)

if nevner == 0:
    print("Nevneren kan ikke være 0.")
elif valg == "+":
    svar = teller1 + teller2
    print("Svaret er:", svar, "/", nevner)
elif valg == "–":
    svar = teller1 – teller2
    print("Svaret er:", svar, "/", nevner)
else:
    print("Du må velge + eller –.")
```

Prøv selv

Test 2/7 + 3/7, 5/8 - 2/8 og 1/4 - 3/4. Du skal få 5/7, 3/8 og -2/4. Test også nevner 0 og et ugyldig valg.

Stopp og forklar

Hvorfor sjekker vi om nevneren er 0 før vi regner?

Bruk positive nevnere når dere øver. Vi lar svaret stå uten forkorting. -2/4 er riktig selv om det også kan skrives -1/2.

Forskjellige nevnere

Ta dette steget når dere er trygge på de første seks. Nå må brøkene få like store deler før vi legger sammen eller trekker fra.

```
1/2 + 1/3 = 3/6 + 2/6 = 5/6
```

For å gjøre 1/2 om til sjettedeler ganger vi både teller og nevner med 3. For 1/3 ganger vi begge med 2. Brøkene beholder verdien sin.

La Python gjøre om brøkene

```
teller1 = 1
nevner1 = 2
teller2 = 1
nevner2 = 3

felles_nevner = nevner1 * nevner2
ny_teller1 = teller1 * nevner2
ny_teller2 = teller2 * nevner1

print(ny_teller1, "/", felles_nevner)
print(ny_teller2, "/", felles_nevner)
```

* betyr gange i Python. Vi ganger nevnerne for å finne en felles nevner. Den fungerer selv om den ikke alltid er den minste.

Prøv selv

Endre brøkene til 1/4 og 1/2. Programmet skal vise 2/8 og 4/8. Regn deretter ut summen på papir.

Stopp og forklar

Hvorfor må vi endre telleren også når vi endrer nevneren?

Din brøkkalkulator

Her er hele programmet med to nevnere. Følg verdiene gjennom koden.

```
teller1 = int(input("Første teller: "))
nevner1 = int(input("Første nevner: "))
teller2 = int(input("Andre teller: "))
nevner2 = int(input("Andre nevner: "))
valg = input("Velg + eller –: ")

print("Du valgte:", teller1, "/", nevner1,
      valg, teller2, "/", nevner2)

if nevner1 == 0 or nevner2 == 0:
    print("Nevneren kan ikke være 0.")
elif valg == "+" or valg == "–":
    felles_nevner = nevner1 * nevner2
    ny_teller1 = teller1 * nevner2
    ny_teller2 = teller2 * nevner1

    if valg == "+":
        svar = ny_teller1 + ny_teller2
    else:
        svar = ny_teller1 – ny_teller2

    print("Svaret er:", svar, "/", felles_nevner)
else:
    print("Du må velge + eller –.")
```

or betyr eller. I regnedelen er det et nytt if inni elif. Linjene inni dette valget har åtte mellomrom. Der vet vi allerede at valget er + eller -.

Prøv selv

Test 1/2 + 1/3 = 5/6 og 3/4 - 1/2 = 2/8. Svaret 2/8 er riktig; forkorting til 1/4 kan vi legge til senere.

Når du står fast

Programmet venter

Ser du et spørsmål? Klikk i konsollen, skriv svaret og trykk Enter. Det er vanlig at programmet venter på input().

SyntaxError

Se etter manglende anførselstegn, parentes eller kolon. Se også på linjen rett før stedet Python peker på.

IndentationError

Sjekk innrykk. Etter if, elif og else skal linjene inni valget rykkes inn fire mellomrom. Ikke bland tabulator og mellomrom.

NameError

Sjekk at variabelnavnet er skrevet likt overalt, og at variabelen har fått en verdi før du bruker den.

ValueError

int() fikk noe som ikke kan bli et heltall. Kjør igjen og skriv for eksempel 3, ikke tre eller 1/2.

Svaret blir 23 i stedet for 5

Sjekk at tallene er gjort om med int(). Tekstene "2" og "3" blir satt sammen til "23".

Kan du forklare dette nå?

Hva gjør input()? Hva gjør print()? Hva er en variabel? Hva er forskjellen på = og ==? Når kjører elif? Hvorfor må brøkene ha felles nevner?

Til den voksne: Be om forklaring og en liten endring i koden, fremfor å spørre om han har forstått. Gå tilbake ett steg hvis det blir mye på en gang.