



Kalkulator

Kalkulator

I denne oppgaven skal du lage en kalkulator helt på egenhånd. Det er koden selv, ved hjelp av noen hint.

Vi ønsker at kalkulatoren skal kunne addere (`+`), subtrahere (`-`), multiplisere (`*`) og dividere (`/`), og i denne oppgaven skal du lage en funksjon for hver av `subtract`, `multiply` og `divide`. Hver funksjon skal ha to tall som parameter og returnere ut svaret.

Brukeren skal selv skrive inn hva slags regneoperasjon som skal utføres.

Eksempel på bruk av programmet:

```
>>>
Operator: *
First number: 3
Second number: 5
3 * 5 = 15
>>>
```

Klar, ferdig, programmer!

Da er det bare å sette i gang!

Her er noen ting å tenke på:

- ☐ Hvordan avgjør du hvilken operasjon som skal utføres?
- ☐ Har rekkefølgen på tallene noen betydning? (Er `4-2` lik `2-4`?)



Hvis du sitter fast kan det være lurt å lese tipsene i de gule bok

int()

Når man får input fra brukeren får man en *tekststreng*, selv om brukeren skriver et tall. For å konvertere teksten til et tall, ved hjelp av `int()`.

Hva er forskjellen på disse kodesnuttene? (Kjør koden og test selv!)

```
tall = input("Skriv et tall: ")
svar = 3 + tall
print(svar)
```

```
tall = int(input("Skriv et tall: "))
svar = 3 + tall
print(svar)
```

Funksjoner med parametre

En funksjon deklarerer ved hjelp av `def`-nøkkelordet. Den kan brukes på mange måter.

Eksempel:

```
def hello_word():
    print("Hello World!")

hello_word()
```

En funksjon som har *parametre* deklarerer med parametre på innsiden.

Eksempel:

```
def greet(firstName, lastName):  
    print("Hello, " + firstName + " " + lastName)
```

Når vi senere kaller funksjonen, så gir vi den *argumenter*.

Eksempel:

```
greet("Ola", "Nordmann")
```

Du la kanskje merke til at vi skiller mellom *parametre* og *argumenter* i funksjonendefinisjonen, slik som `firstName` og `lastName`. Argumentene, slik som `"Ola"` og `"Nordmann"`.

Test programmet

Fungerer programmet som det skal? Hvis ikke må du rette på det.

Delt på null

Hva skjer når du deler på null? Prøv for eksempel `4 / 0`.

Hvis programmet ditt feiler nå, så har du trolig fått en delt-på-null-feil. Programmet ditt slik at programmet skriver ut `"Division by zero"`. Slik:

```
>>>  
Operator: /  
First number: 4  
Second number: 0  
Division by zero is not  
>>>
```

Flere utregninger

Endre programmet ditt slik at brukeren kan skrive inn hvor mange vil da fungere slik:

```
>>>
How many calculations? 4
Operator: +
First number: 0
Second number: 246
0 + 246 = 246

Operator: *
First number: 3
Second number: 255
3 * 255 = 765

Operator: /
First number: 4
Second number: 0
Division by zero is not

Operator: /
First number: 0
Second number: 4
0 / 4 = 0.0

>>>
```

Flere operasjoner

Prøv å legge til flere operatorer. Du kan for eksempel legge til `**` -
annet. F.eks. er `2**3` lik `8` fordi `2*2*2` er lik `8`.

Kommer du på andre operatorer som kan legges til i kalkulatoren?

