

Progoblig 01 individuell del

Oppgave Ind01

a) Finn operatører og operander i uttrykket $3 + 5$?

- Operatøren er pluss tegnet, det sier noe om handlingen som skal utføres
- Operanden er tallene

b) Nevn 4 grunnleggende aritmetiske operatører.

- Multiplikasjon (gange)
- Divisjon (dele)
- Addisjon (pluss)
- Subtraksjon (minus)

c) Hvordan kan uttrykket $3 + 4 * 5$ tolkes av en kompilator (på en datamaskin)?

Den kan tolkes på forskjellige måter avhengig av hvilken datamaskin man bruker fordi JavaScript og Pyret har to ulike måter å regne det ut på. Man kan få forskjellige resultater.

d) Forklar resultatet av evalueringen av uttrykket fra punkt c), som

<https://code.pyret.org/editor> (Pyret fortolker, videre forkortet til CPO som forkortelse for code.pyret.org) returnerer.

Dette er hva som skjer når man skriver inn uttrykket i CPO:

Reading this expression errored:

definitions://:1:0-1:9

2 3 + 4 * 5

The **+** and ***** operations are at the same grouping level. Add parentheses to group the operations, and make the order of operations clear.

Resultatet av denne evalueringen er en feilmelding. Forklaring på dette kan være at at Pyret ikke følger den tradisjonelt hierarkiske rekkefølgen som kan huskes som «PEMDAS»:

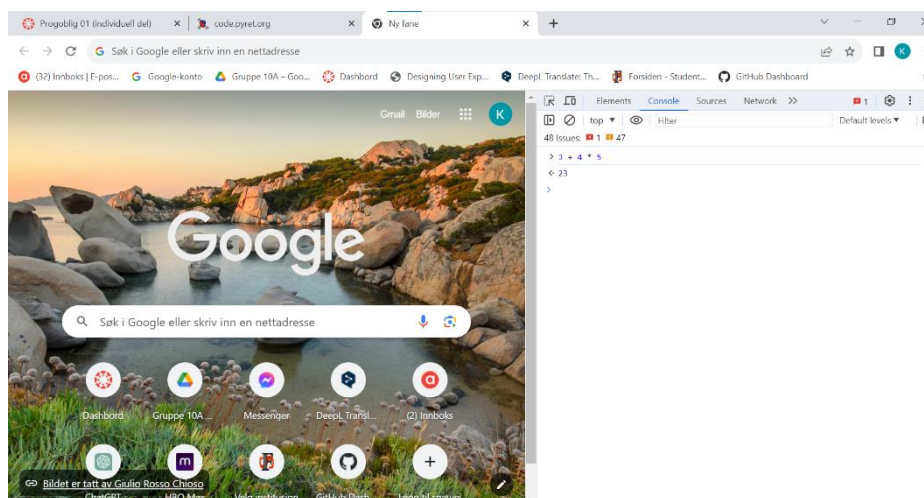
Rekkefølgen på hva som skal regnes ut/løses først:

1. Parentes
2. Exponent
3. Multiplikasjon
4. Divisjon
5. Addisjon
6. Substraksjon

Den klarer tilsynelatende ikke å skille mellom addisjon og multiplikasjon i hierarkisk rekkefølge.

e) Forklar resultatet av evalueringen av uttrykket fra punkt c), som JavaScript-konsollen (JavaScript fortolker, videre forkortet til JSF) i en nettleser returnerer.

Dette er hva som skjer når man skriver inn uttrykket i JSF:

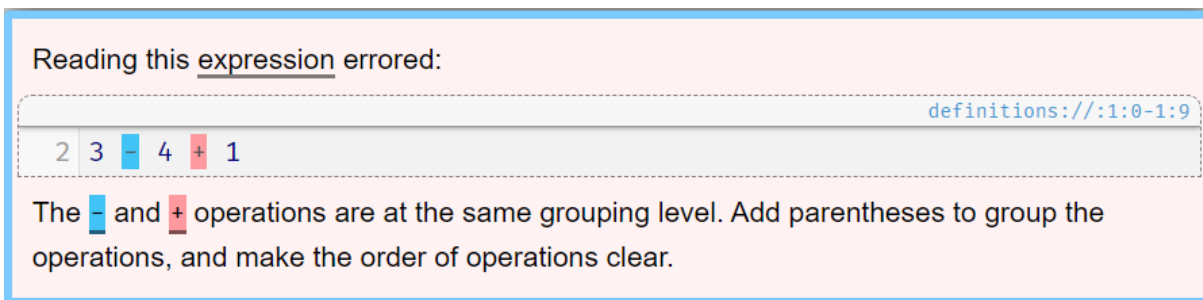


Resultatet her viser til at multiplikasjon (*) har høyere «prioritering» i utregningsrekkefølgen enn addisjon (+). Derfor blir det riktig resultat på uttrykket.

1. gruppering	1.Parentes
2. gruppering	2.Eksponent
3. gruppering	3. Multiplikasjon
	4.Divisjon
4. gruppering	5.Addisjon
	6.Substraksjon

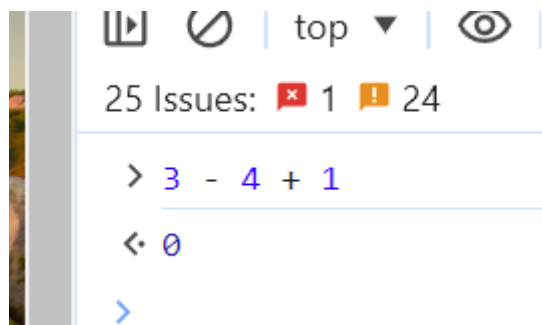
f) Hva er forskjeller på hvordan $3 - 4 + 1$ blir fortolket i CPO og JSF? Hvorfor?

Dette er hva som skjer når man skriver inn uttrykket i CPO:



Det kommer en feilmelding i CPO. Tilsynelatende klarer den ikke «vite» at den skal starte utregningen fra venstre til høyre fordi addisjon og subtraksjon er i samme type gruppering.

Dette er hva som skjer når man skriver inn uttrykket i nettleser, JSF:



Resultatet blir her riktig, da det følger en rekkefølge for utregningen fra venstre til høyre.

g) Hvordan må man endre uttrykket fra punkt f) for at resultatet skal være -2 i både CPO og JSF? Forklar.

Det man må endre på i uttrykket fra punkt f) for at resultatet skulle bli -2 var å legge til parentes slik at det ser slik ut: $3 - (4 + 1)$. Man må gruppere eller spesifisere det som skal regnes sammen med hverandre for å få den til å regne ut og komme frem til riktig svar. Da ble resultatet -2 i både CPO og JSF.

h) Hva kan man si om hvordan operatørene “+” og “-” blir behandlet i JSF?

JSF regner tilsynelatende en til to operatører i samme gruppering, slik som skrevet om tidligere virker det som addisjon og subtraksjon er i samme gruppe. Når man skriver inn uttrykk regner også JSF ut + og – som samme type gruppe, og prioriterer derfor ikke for eksempel å alltid subtrahere før den adderer, den bare løser uttrykket fra rekkefølgen venstre til høyre. For å oppnå et annet resultat skulle man brukt parentes for å presisere for JSF hva som skulle regnes ut sammen, eller hva som skulle regnes ut først.

i) Hva kan man si om hvordan “*” og “/” operatørene blir behandlet i JSF?

JSF regner som sagt tilsynelatende ut med grupperinger og deretter fra venstre til høyre. Det man legger merke til med grupperingene når man skriver inn uttrykk som $3 + 4 * 2$, er at JSF prioriterer multiplikasjonen før addisjon. Det vil si at disse operatørene regnes som av høyere «verdi»/prioritering enn addisjon og subtraksjon. Det vil si at den uansett regner ut multiplikasjon og divisjon før addisjon og subtraksjon. Hvis det kun er multiplikasjon og divisjon så regner den ut fra venstre til høyre. Den tar den som har høyest prioritering og innad i de grupperingene vil den regne ut fra venstre til høyre.

j) Lage et uttrykk som inneholder alle 4 aritmetiske operatørene og forklar hvordan JSF takler det.

Dette er ett uttrykket som inneholder alle de aritmetiske operatørene: $5 * 7 + 4 / 2 - 3$

Måten JSF hadde regnet ut dette uttrykket på er å regne ut multiplikasjon og divisjon først og deretter addisjon og subtraksjon. Det ville skjedd på følgende utregningsmåte:

$$5 * 7 + 4 / 2 - 3$$

$$35 + 2 - 3$$

$$37 - 3$$

$$= 24$$

Det stemte også når jeg testet det og skrev det inn i JSF i nettleseren.

k) Hva er fordelene og ulempene med måten CPO fortolker aritmetiske uttrykk med alle de 4 grunnleggende aritmetiske operatørene?

Fordeler: Blir mer nøyaktig resultat, hvis man legger inn arbeidet. Gruppere det riktig med parenteser ovs.

Ulemper: Det kan være med på å forvirre. Det oppstår feilmelding når man prøver å teste koden fordi den ikke klarer å skille de ulike operatørene. Må også skrive mye mer, og det kan gjøre kodingen ineffektivt, for jo mindre man behøver å skrive/presisere, jo mer effektivt blir det.

l) Hvordan kan man få regnemaskinen til å gjøre andre beregninger enn de grunnleggende aritmetiske operatørene? Hva gjør operatørene (som også ofte kalles funksjoner) num-max og num-min i CPO? Gi eksempler og forklar.

En operatør er et program. Det definerer noe.

num-max () – er en funksjon, alt man putter inn i parentes skiller man med komma. Det som skjer når man putter inn tall, er at den finner det høyeste/største tallet av de innsatte tallene.

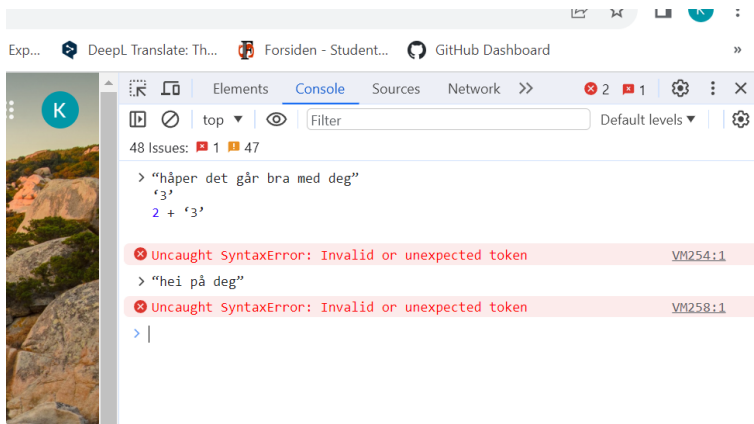
num-min () – Det som skjer når man putter inn tall, er at den finner det laveste/minste tallet av de innsatte tallene.

m) Sorter følgende eksempler i forhold om de er et uttrykk, en verdi eller et program:

3 + 4 - num-min(7, 4)	Uttrykk
3 + 4 3 - 4 num-max(-4, 4) “hei”	Program, over flere linjer
“IS-114”	Verdi
3.141592653	Verdi
“hei på deg”	Verdi
“håper det går bra med deg” ‘3’ 2 + ‘3’	Program

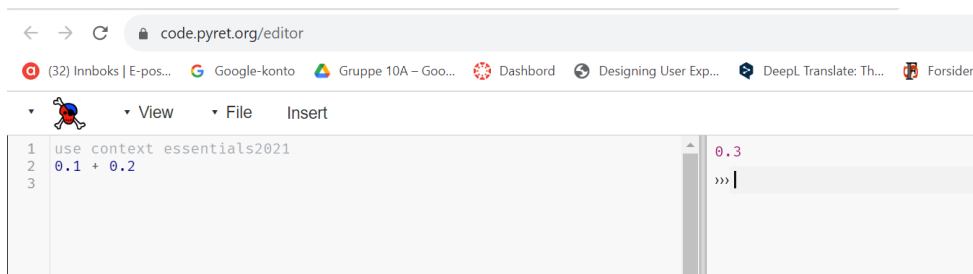
n) Kan JSF fortolke et uttrykk? En verdi? Et program? Forklar og vis med eksempler.

Ja DevTools-konsollen i nettleser kan tolke flere linjer.

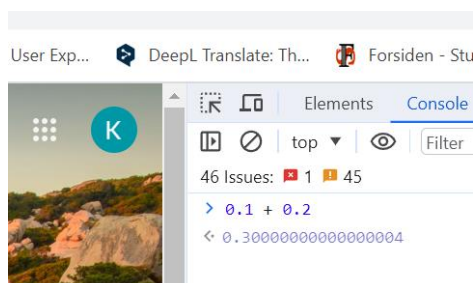


o) OBS! VANSKELIG SPØRSMÅL: Sammenligne hvordan JSF og CPO tolker uttrykket $0.1 + 0.2$. Forklar og demonstrer med eksempler i begge fortolkere.

For å svare på denne oppgaven er viktig å kunne noe om tallenes egenskaper og ulike metoder til å regne med tall. Dette er hvordan CPO tolket uttrykket:



Dette er hvordan JSF tolket uttrykket:



Problemet oppstår når programmeringsspråkene behandler desimaltall internt i det binære tallsystemet. I JSF gav $0.1 + 0.2$ et resultat på 0.30000000000000004 , mens i CPO ble svaret det man gjerne ville forventet, som er 0.3 . Dette skyldes avrundingsfeil i binær representasjon. Datamaskiner bruker kun 0 og 1 i det binære systemet, og noen desimaltall kan ikke uttrykkes

nøyaktig. Rundingsnivået avhenger av datamaskinens presisjon. Hvis eller når man arbeider med slike tall, kan man prøve å unngå å sammenligne for likhet eller begrense antall desimalplasser i resultatene som vises for å unngå forvirring.

Oppgave Ind02

a) Gitt følgende verdier: 64, “A”, “2023-10-12 12:00”

Navngi variabel for å representere hjerterytme eller puls, en eksamenskarakter, tidspunkt for møte (navn er antydnet her, men du må foreslå navn, som tilfredsstiler Pyret kodestil)

- hjerterYtme = 64
- eksamensKarakter = «A»
- moteTidspunkt = “2023-10-12 12:00”

b) Hvilke av disse er definisjonene og hvilke er uttrykk (tall foran hver linje er ikke en del av koden) i CPO. Begrunn.

1. $5 + 8$

2. $x = 14 + 16$

3. `triangle(20, "solid", "purple")`

4. `blue-circ = circle(x, "solid", "blue")`

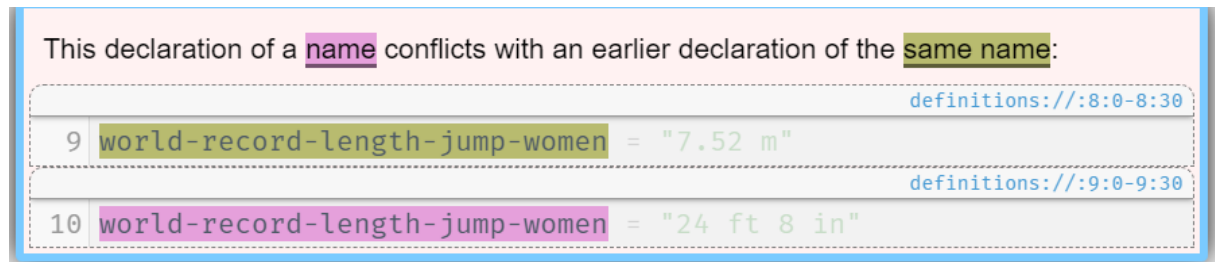


1. Dette er et uttrykk. Det er et enkelt matematisk regnestykke, som kan regnes ut.
2. Dette er en definisjon. X er en variabel.
3. Denne kan tolkes som et uttrykk. Det er en kode for å lage en lilla trekant.
4. Dette er en definisjon. Ved å skrive «blue-circ» først blir de spesifikke egenskapene som blir skrevet etter, tildelt variabelen «blue-circ».

c) Hva skjer når man skriver inn to påfølgende definisjoner med det samme navnet i CPO? Forklar.

world-record-length-jump-women = “7.52 m”

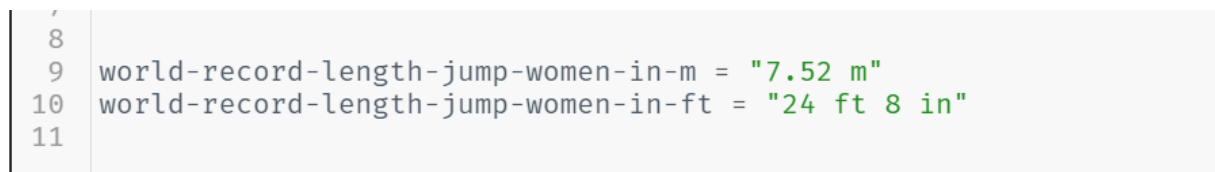
world-record-length-jump-women = “24 ft 8 in”



Dette er resultatet jeg fikk når jeg skrev inn to påfølgende definisjoner med det samme navnet i. Det kom opp en feilmelding hvor det stod at det oppstod en konflikt med en tidligere linje definisjon, fordi den hadde samme navn.

d) Kunne man skrive definisjonene om, slik at CPO responderte annerledes enn i punkt c)?

Ved å endre navnet på definisjonene fikk man ikke lengre feilmelding når man kjørte linjene.



e) Utføre definisjoner fra punkt c) i JSF. Sammenligne resultater med resultater i CPO. Forklar.

```
> world-record-length-jump-women = "7.52 m"
world-record-length-jump-women = "24 ft 8 in"
```

✖ Uncaught SyntaxError: Invalid left-hand side in assignment VM69:1

Dette var resultatet jeg fikk ved å utføre definisjoner fra punkt c) i JSF. Det kom opp en feilmelding her som i JSF. I CPO ble feilmeldingen spesifisert, mens her kom det opp feilmelding om det som ble skrevet etter bruken av bindestrek. Det er fordi at bindestrek har andre funksjoner i JSF, og funker ikke som en binder av navn.

f) Forklar hvordan definisjons- og interaksjons-vinduene fungerer i CPO (code.pyret.org)

Det finnes to vinduer i Pyret, definisjons- og interaksjons-vinduene.

- Definisjonsvinduet er til venstre. Der kan man opprette og definere sine egne funksjoner. Kodene man skriver inn her kan brukes for å bygge opp kodeelementer, fordi de ulike elementene blir lagret i prosessen.
- Interaksjonsvinduet er til høyre, her kan man prøve seg frem med ulike koder uten å måtte bygge et fullstendig program. Man kan altså teste og kjøre kodene man har skrevet i definisjonsvinduet. Her får man også opp resultatet av koden og kan samhandle med den i dette vinduet. Dette er et fint vindu for eksperimentering med koding.

g) Skriv og forklar koden som produserer Norges handels- og nasjonalflagg i interaksjonsvinduet til CPO når man trykker “Run” (laste koden opp i din egen Github-repository)

Dette er en eksempelkode som produserer Norges handels- og nasjonalflagg:

```
a = put-image(rectangle(330, 50, "solid", "white"), 165, 120, rectangle(330, 240, "solid", "red"))

b = put-image(rectangle(60, 240, "solid", "white"), 130, 120, a)

c = put-image(rectangle(330, 25, "solid", "dark blue"), 165, 120, b)

put-image(rectangle(25, 240, "solid", "dark blue"), 130, 120, c)
```

