IN1000 UKE 2 - KONSEPTFORSTÅELSE

Gruppe 8

Plan for dagen



Kort gjennomgang av ukas pensum



Jobbe med oppgaver

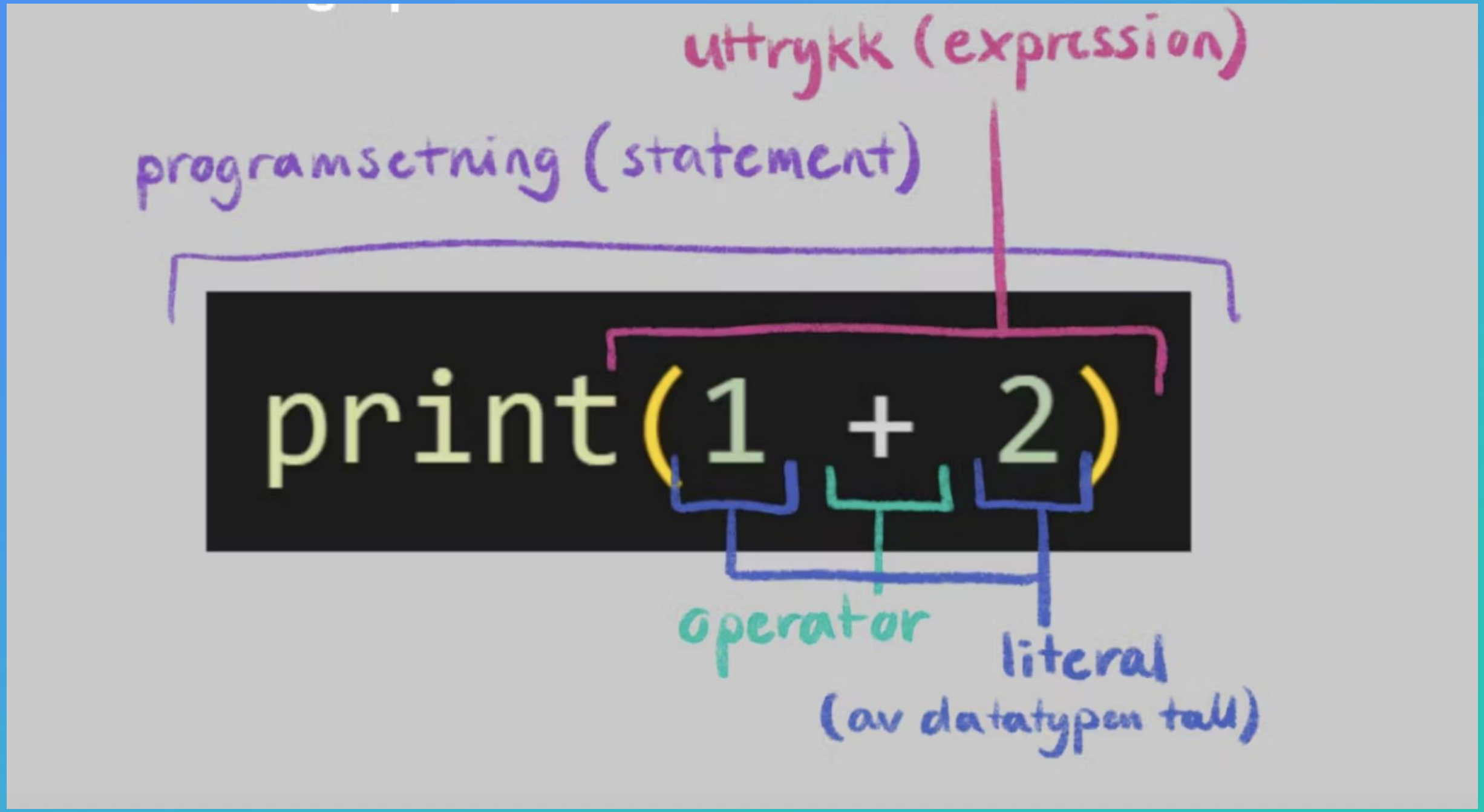


Eventuelt: Gjennomgang av quiz



Læringsmål for uken

- Forstå hvordan én enkelt linje utføres:
  - Datatyper, evaluering av uttrykk og funksjoner (input, int, float, str, etc.)
- Ha god forståelse av variabler
- Forstå og kunne bruke enkle prosedyrer uten parametre
- Forstå hvordan et helt program utføres - kodeflyt fra linje til linje


uttrykk (expression)
programsetning (statement)
print(1 + 2)
operator
literal
(av datatypen tall)

Datatyper

```
2
3    tekst = "hei"    # Tekst (String)
4    tall = 5         # Heltall (Integer)
5    flyt = 5.1       # Flyttall (Float)
6    sant = True      # Sannhetsverdi (Boolean)
7
8    print("5")       # Tekst - fordi det er hermetegn rundt
9    print('tekst')   # Tekst - kan også bruke enkle hermetegn
```

Typekonvertering

- str()
- int()
- float()

```
a = "7"
b = "1"
print(a + b)
```

Hva skrives ut her?

```
alder = input("Hvor gammel er du?\n> ")


alder = int(input("Hvor gammel er du?\n> "))
```

Tilordning av variabler/evaluering av uttrykk

Variabler tilordnes verdien av summen av høyresiden av likhetstegnet (ergo: gjør høyresiden først!)

Vi kan bruke paranteser for å vise hvilken rekkefølge ting skal skje i! (Man begynner innerst og ”jobber seg utover”)

```
tall = 5 + 1 - 4

tall = ((5 + 2) * 2) + (4 - 1)
```

CONDITIONS

	BETYDNING
<	mindre enn
>	større enn
==	er lik
!=	er ikke lik
<=	mindre eller lik
>=	større eller lik
and	True hvis begge betingelser er sanne
or	True hvis (minst) en av betingelsene er sanne
not	reverserer resultatet

```
True and True
True and False

True or True
True or False
False or False

not True
not False

# Ekstra
True and not True
True and not False
not True or not False
```

Hva evalueres disse uttrykkene til?

Diskuter i små grupper!

```
True and True                   # True
True and False                  # False

True or True                    # True
True or False                   # True
False or False                  # False

not True                        # False
not False                       # True

# Ekstra
True and not True               # False
True and not False              # True
not True or not False           # True
```

Fasit

Ulike typer feil

Syntaksfeil (syntax error)

- Noe er galt med skrivemåten til noe

Kjøretidsfeil (runtime error)

- Noe er galt når programmet kjøres

Logisk feil (logical error)

- Noe er galt med måten vi har tenkt på
- (Får ingen feilmelding!)

Prosedyrer – uten parametre

- Why? Gjenbruk av kode! Mer oversiktelig
- Syntaks:

```
def si_hallo():
    print("Hallo!")
    print("Ha en fin dag :D")

si_hallo()
si_hallo()
si_hallo()
si_hallo()
```

Oppgave 1

a) Lag fire variabler, *heltall1*, *heltall2, flyttall1 og flyttall2*. Tilordne variablene tall som passer med variabelnavnene ved bruk av input().

b) Skriv ut summen og produktet av de to heltallene, deretter av de to flyttallene.

c) Undersøk hva som skrives ut i terminalen når dere summerer og multipliserer et heltall med et flyttall. Hva ser dere? Finn ut hvorfor oppførselen er slik.

d) Hva skjer hvis dere summerer et heltall eller et flyttall med en tekststreng?

Oppgave 2

a) Kjør koden nedenfor og gi en detaljert forklaring av innholdet i feilmeldingen. Hva slags type feil er det snakk om? Diskuter hvordan feilen kan rettes.

b) Endre koden, slik at programmet kjører feilfritt. Løs feilen på to forskjellige måter.

c) Utvid programmet slik dere ønsker, men legg til en logisk feil i utvidelsen. Hva er mulige konsekvenser av den logiske feilen i deres utvidede program?

```
flyselskap = "Norwegian"
rutenummer = 8537
print(flyselskap + "s", "rute " + rutenummer + ".")
```

Oppgave 3

Diskuter hvordan boolske uttrykk (og boolske operatorer) henger sammen med evaluering av uttrykk i programmering. Bruk begrepet sannhetsverdier i diskusjonen, og finn eksempler med og uten bruk av tall.

Oppgave 4

Beskriv flyten i programmet nedenfor i detalj (når hvilke programlinjer kjøres og hva som nøyaktig skjer i hver programlinje). Skriv deretter koden inn i PythonTutor og sammenlign Python Tutors beskrivelse av programflyten med deres egen.

```
def opprett_bankkonto():
    print("Velkommen til opprettelse av bankkonto!")
    saldo = float(input("Skriv inn inngående saldo: "))
    if(saldo == 0):
        print("Du kan ikke opprette konto med 0 kroner.")
    elif(saldo < 0):
        print("Du kan ikke opprette konto med et negativt beløp.")
    else:
        print("Konto er opprettet med " + str(saldo) + " kroner.")

opprett_bankkonto()
```