

Objekter tilbyr  
tjenester

# Læringsmål

# Læringsmål

- Etter modulen:

# Læringsmål

- Etter modulen:
  - Kjenner du begrepet objekter og forstår hva det innebærer at objekter tilbyr tjenester

# Læringsmål

- Etter modulen:
  - Kjenner du begrepet objekter og forstår hva det innebærer at objekter tilbyr tjenester
  - Vet du at variabler holder objekter, kjenner syntaksen for å bruke tjenester for et objekt og vet hvordan uttrykk som inkluderer bruk av tjenester evaluerer

# Forkunnskaper

# Forkunnskaper

- Modulen bygger direkte på:

# Forkunnskaper

- Modulen bygger direkte på:
  - Variabler



# Forkunnskaper

- Modulen bygger direkte på:
  - Variabler
  - Evaluering av uttrykk og utførelse av kodelinjer

# Forkunnskaper

- Modulen bygger direkte på:
  - Variabler
  - Evaluering av uttrykk og utførelse av kodelinjer
- Nyttig for å forstå alle aspekter og eksempler:

# Forkunnskaper

- Modulen bygger direkte på:
  - Variabler
  - Evaluering av uttrykk og utførelse av kodelinjer
- Nyttig for å forstå alle aspekter og eksempler:
  -

# Objekter

# Objekter

- IN1000 er et kurs i objekt-orientert programmering

# Objekter

- IN1000 er et kurs i objekt-orientert programmering
  - Det å lage, bruke og forstå objekter vil være sentralt senere i faget

# Objekter

- IN1000 er et kurs i objekt-orientert programmering
  - Det å lage, bruke og forstå objekter vil være sentralt senere i faget
  - For nå nøyer vi oss med hvordan vi kan bruke tjenester som objekter tilbyr

# Objekter

- IN1000 er et kurs i objekt-orientert programmering
  - Det å lage, bruke og forstå objekter vil være sentralt senere i faget
  - For nå nøyer vi oss med hvordan vi kan bruke tjenester som objekter tilbyr
- Verdier er objekter og tilbyr ofte nyttige tjenester



# Objekter

- IN1000 er et kurs i objekt-orientert programmering
  - Det å lage, bruke og forstå objekter vil være sentralt senere i faget
  - For nå nøyer vi oss med hvordan vi kan bruke tjenester som objekter tilbyr
- Verdier er objekter og tilbyr ofte nyttige tjenester
  - Noen objekter er konkrete og rett frem, slik som en tekstverdi

# Objekter

- IN1000 er et kurs i objekt-orientert programmering
  - Det å lage, bruke og forstå objekter vil være sentralt senere i faget
  - For nå nøyer vi oss med hvordan vi kan bruke tjenester som objekter tilbyr
- Verdier er objekter og tilbyr ofte nyttige tjenester
  - Noen objekter er konkrete og rett frem, slik som en tekstverdi
  - Andre objekter er mer abstrakte og sammensatte, slik som et programvindu med menyer osv.

# Objekter

- IN1000 er et kurs i objekt-orientert programmering
  - Det å lage, bruke og forstå objekter vil være sentralt senere i faget
  - For nå nøyer vi oss med hvordan vi kan bruke tjenester som objekter tilbyr
- Verdier er objekter og tilbyr ofte nyttige tjenester
  - Noen objekter er konkrete og rett frem, slik som en tekstverdi
  - Andre objekter er mer abstrakte og sammensatte, slik som et programvindu med menyer osv.
- Punktum benyttes for å aksessere tjenestene (metodene) til et objekt

Eksempel på tjenester  
som tekstverdier tilbyr

# Eksempel på tjenester som tekstverdier tilbyr

- **upper** gir teksten som store bokstaver

# Eksempel på tjenester som tekstverdier tilbyr

- **upper** gir teksten som store bokstaver
  - `setning = "hallo"`  
`print( setning.upper() )` #HALLO

# Eksempel på tjenester som tekstverdier tilbyr

- **upper** gir teksten som store bokstaver
  - `setning = "hallo"`  
`print( setning.upper() )` #HALLO
- **count** teller hvor mange ganger man finner et bestemt tegn

# Eksempel på tjenester som tekstverdier tilbyr

- **upper** gir teksten som store bokstaver
  - `setning = "hallo"`  
`print( setning.upper() )` #HALLO
- **count** teller hvor mange ganger man finner et bestemt tegn
  - `setning = "kykkeli ky"`  
`print( setning.count("k") )` #4



Mer abstrakte objekter:  
tegne på skjermen

# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret

# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret
  - Med *print* kommer utskriften rett til terminalen

# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret
  - Med *print* kommer utskriften rett til terminalen
  - For å tegne figurer må man lage et eget vindu

# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret
  - Med *print* kommer utskriften rett til terminalen
  - For å tegne figurer må man lage et eget vindu
  - Man lager først et vindu-objekt som representerer et område på skjermen

# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret
  - Med *print* kommer utskriften rett til terminalen
  - For å tegne figurer må man lage et eget vindu
  - Man lager først et vindu-objekt som representerer et område på skjermen
  - Deretter får man fra vinduet et lerret-objekt som er det som holder selve tegningene (en av tjenestene et vindu tilbyr er å lage lerret)

# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret
  - Med *print* kommer utskriften rett til terminalen
  - For å tegne figurer må man lage et eget vindu
  - Man lager først et vindu-objekt som representerer et område på skjermen
  - Deretter får man fra vinduet et lerret-objekt som er det som holder selve tegningene (en av tjenestene et vindu tilbyr er å lage lerret)
- Hvordan dette ser ut i kode:

# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret
  - Med *print* kommer utskriften rett til terminalen
  - For å tegne figurer må man lage et eget vindu
  - Man lager først et vindu-objekt som representerer et område på skjermen
  - Deretter får man fra vinduet et lerret-objekt som er det som holder selve tegningene (en av tjenestene et vindu tilbyr er å lage lerret)
- Hvordan dette ser ut i kode:
  - `vindu = GraphicsWindow()`



# Mer abstrakte objekter: tegne på skjermen

- Mer abstrakte objekter: vindu og lerret
  - Med *print* kommer utskriften rett til terminalen
  - For å tegne figurer må man lage et eget vindu
  - Man lager først et vindu-objekt som representerer et område på skjermen
  - Deretter får man fra vinduet et lerret-objekt som er det som holder selve tegningene (en av tjenestene et vindu tilbyr er å lage lerret)
- Hvordan dette ser ut i kode:
  - `vindu = GraphicsWindow()`
  - `lerret = vindu.canvas()`

# Tegne på skjermen

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne
  - I programmet: `from ezgraphics import GraphicsWindow`



# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefild:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne
  - I programmet: `from ezgraphics import GraphicsWindow`
- Deretter lager man objekter og begynner å tegne:

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne
  - I programmet: `from ezgraphics import GraphicsWindow`
- Deretter lager man objekter og begynner å tegne:
  - Lag vindu og lerret (GraphicsWindow og canvas)

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne
  - I programmet: `from ezgraphics import GraphicsWindow`
- Deretter lager man objekter og begynner å tegne:
  - Lag vindu og lerret (GraphicsWindow og canvas)
  - Tegn rektangler o.l. på lerretet (drawRect)

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne
  - I programmet: `from ezgraphics import GraphicsWindow`
- Deretter lager man objekter og begynner å tegne:
  - Lag vindu og lerret (GraphicsWindow og canvas)
  - Tegn rektangler o.l. på lerretet (drawRect)
  - Sørg for at vinduet ikke bare lukkes med en gang (wait)

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne
  - I programmet: `from ezgraphics import GraphicsWindow`
- Deretter lager man objekter og begynner å tegne:
  - Lag vindu og lerret (GraphicsWindow og canvas)
  - Tegn rektangler o.l. på lerretet (drawRect)
  - Sørg for at vinduet ikke bare lukkes med en gang (wait)
- {grafikk.py}

# Tegne på skjermen

- For å kunne tegne i praksis må vi hente en kodefil:
  - Muligheten for å tegne på enkel måte følger ikke med Python
  - Vi trenger derfor en ekstra pakke for dette formålet - læreboka (og vi) bruker ezgraphics
- Hvordan få på plass pakken vi trenger:
  - Vi laster ned en pakke *ezgraphics* og installerer denne
  - I programmet: `from ezgraphics import GraphicsWindow`
- Deretter lager man objekter og begynner å tegne:
  - Lag vindu og lerret (GraphicsWindow og canvas)
  - Tegn rektangler o.l. på lerretet (drawRect)
  - Sørg for at vinduet ikke bare lukkes med en gang (wait)
- {grafikk.py}
- *(merk forøvrig at både utskrift og tegning ikke er blant det sentrale i faget - ikke fokuser mye på det)*