

# Plan for dagen

- Hvordan én enkelt linje utføres:
  - **Dat typer**
  - Evaluering av uttrykk og funksjoner
- Hvordan et helt program utføres:
  - Kodeflyt fra linje til linje
  - Prosedyrer
- Sjekke antagelser og tolke feilmeldinger

# Datatyper

```
print("Hei")
```

Tekst (String)

```
print(5)
```

Heltall (Integer)

```
print(5.1)
```

Flyttall (Float)

```
print("5")
```

Tekst (pga hermetegn)

```
print('hei')
```

Enkle eller doble  
hermetegn

# Man kan sjekke datatype vha funksjonen *type*

```
type("Hei")  
type(5)  
type(5.1)  
type("5")  
type('hei')
```

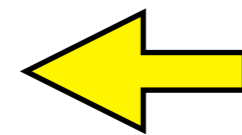
# Operasjoner avhenger av datatype

```
print(2+3)
```



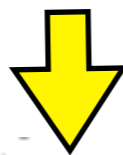
+ betyr addisjon

```
print("Hei" + "IN1000")
```



+ betyr  
konkatenerer  
tekst

```
print("Hei" + 1000)
```



```
guardian:kode_uke1 sandve$ python3 hei.py
```

```
Traceback (most recent call last):
```

```
File "hei.py", line 1, in <module>
```

```
print("Hei" + 1001)
```

```
TypeError: Can't convert 'int' object to str implicitly
```

# Hva skrives ut her?

```
a = "5"  
b = "5"  
print(a + b)
```

# Innhente tall fra brukeren

- Man ønsker ofte at bruker skal oppgi tall
  - **input** gir alltid en verdi av type tekst
- Biblioteksfunksjonen **int** konverterer en tekst til et tall (om mulig)
  - `tall = int("5")`
- Enkelt å teste ut:
  - `print( "5"+"5" )`
  - `print( int("5") + int("5") )`

# Innhente tall fra brukeren

- Innhente tall:
  - Be brukeren skrive et tall
  - Bruke **input** til å hente det i form av tekst
  - Bruke **int** for å konvertere til heltall (integer)
  - Tilsvarende bruke **float** for å få flyttall (float)
  - (Man kan også konvertere tall til tekst med **str**)
- Om teksten ikke er tall får man feilmelding
  - Traceback (most recent call last):  
File "<stdin>", line 1, in <module>  
ValueError: invalid literal for int() with base 10: 'hei'
- [innlesing\_tall.py]