

KRASJ OG FEIL ASSERT

INF100

HØST 2026

Torstein Strømme

RECAP: KODESPORING OG FUNKSJONER

```
1  def incremented_thrice(x):
2      x = x + 1
3      x += 1
4      return x + 1
5
6  a = 5
7  b = incremented_thrice(a)
8  c = incremented_thrice(b)
9
10 print(f"a: {a}, b: {b}, c: {c}")
```

HVA MED DETTE?

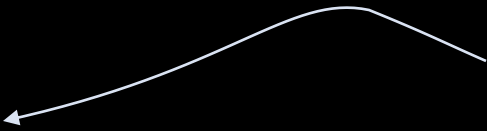
ikke
samme
variabel!

```
1  def incremented_thrice(x):  
2      x = x + 1  
3      x += 1  
4      return x + 1  
5  
6  x = 5  
7  y = incremented_thrice(x)  
8  y = incremented_thrice(y)  
9  
10 print(f"x: {x}, y: {y}")
```

HVA MED DETTE?

```
1  def incremented_thrice(x):  
2      x = x + 1  
3      x += 1  
4      print(x + 1)  
5  
6  x = 5  
7  y = incremented_thrice(x)  
8  y = incremented_thrice(y)  
9  
10 print(f"x: {x}, y: {y}")
```

uten retursetning vil None returneres



KRASJ OG FEIL

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre

```
def volume_of_box(x, y, z)  
    print(x * y + z)
```

```
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen
```

line 4

```
    print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen
```

SyntaxError: unterminated string literal (detected at line 4)

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre

```
def volume_of_box(x, y, z)  
    print(x * y + z)
```

```
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 1  
    def volume_of_box(x, y, z)  
                                ^
```

```
SyntaxError: expected ':'
```

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre

```
def volume_of_box(x, y, z):  
    print(x * y + z)
```

```
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4  
    print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen"  
      ^  
SyntaxError: '(' was never closed
```

TRE FORMER FOR FEIL

- Krasj (engelsk: runtime error)
 - Programmet krasjer når det kjører

```
def volume_of_box(x, y, z):  
    print(x * y + z)  
  
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4, in <module>  
    print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")  
NameError: name 'volum_of_box' is not defined. Did you mean:  
'volume_of_box'?
```

TRE FORMER FOR FEIL

- Krasj (engelsk: runtime error)
 - Programmet krasjer når det kjører

```
def volume_of_box(x, y, z):  
    print(x * y + z)
```

```
print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4, in <module>  
    print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")  
TypeError: can only concatenate str (not "NoneType") to str
```

TRE FORMER FOR FEIL

- Krasj (engelsk: runtime error)
 - Programmet krasjer når det kjører

```
def volume_of_box(x, y, z):  
    return x * y + z
```

```
print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4, in <module>  
    print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")  
TypeError: can only concatenate str (not "int") to str
```

TRE FORMER FOR FEIL

- Logisk feil
 - Programmet gir feil svar

```
def volume_of_box(x, y, z):  
    return x * y + z
```

```
print("Det er plass til " + str(volume_of_box(1, 2, 3)) + " m3 i boksen")
```

```
Det er plass til 5 m3 i boksen
```

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre
 - Feilmelding gir visuell indikasjon på hva som er feil
- Krasj
 - Programmet krasjer underveis i kjøring
- Logiske feil
 - Programmet gir galt svar

IndentationError
SyntaxError

AttributeError
IndexError
KeyError
NameError
TypeError
ZeroDivisionError
...

ASSERT

- Krasj programmet med vilje når noe ikke er som det skal
- Tester koden, og beskytter mot logiske feil

```
assert True # Gjør ingenting  
assert False # Krasjer
```

- Vi bruker prinsippet om assert når vi retter kode automatisk (CodeGrade)
- Det er mulig å slå av assert for å optimisere kjøretid (men: ikke gjør det)
- Sjekk at assert er aktivt: legg inn `assert False` og se at det krasjer

ASSERT

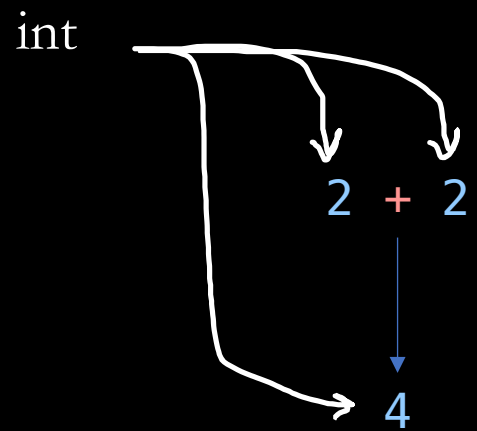
- Krasj programmet med vilje når noe ikke er som det skal
- Tester koden, og beskytter mot logiske feil

```
def volume_of_box(x, y, z):  
    return (x * y + z)
```

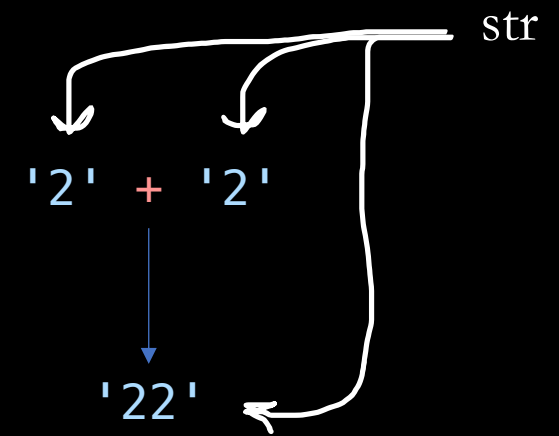
```
assert 6 == volume_of_box(1, 2, 3)  
print("Det er plass til " + str(volume_of_box(1, 2, 3)) + " m3 i boksen")
```

```
line 4, in <module>  
    assert 6 == volume_of_box(1, 2, 3)  
AssertionError
```

TYPED



addisjon



kontatinasjon



sammenligner tallverdi

sammenligner «alfabetisk»

