

Noter til video 2

<u>Operator</u>	<u>Symbol</u>
Addition	+
Subtraktion	-
Multiplikation	*
Division	/
Integer division	//
Modulo (rest)	%

Operator hierarki:

**

- (når det bruges til at beskrive om et tal er negativt)

* / // %

+ -

Numeriske typer:

Int: *hele tal (integer)*

Float: *tal med decimaler (floating point decimal)*

Fejlmeldinger:

Syntaks: *når man har brugt en kombination af operatører, der ikke er valid*

Semantisk: *Når ens udtryk ikke giver forståelsesmæssig mening*

<u>Sammenlignings operator</u>	<u>Symbol</u>
Større end	>
Mindre end	<
Lig med	==
Større end eller lig med	>=
Mindre end eller lig med	<=
Forskellig fra/ikke lig med	!=

Noter til video 3

Når man vil lave sine egne funktioner, skriver man dem sådan her op:

<pre>def area(base, height): '''(number, number) -> number Return the area of a triangle with dimensions base and height. >>> area(10, 5) 25.0 >>> area(2.5, 3) 3.75 ... return base * height / 2</pre>	Header Type Contract Description Examples Body
---	--

Regler for navngivning af variabler/funktioner i

Python:

- 1) Navne skal starte med et bogstav eller _
- 2) Navne må kun indeholde bogstaver, tal og _
- 3) Der er forskel mellem store og små bogstaver
Variablen "omkreds_af_trekant" er forskellig fra "Omkreds_af_Trekant"

Vigtige begreber:

Argument: *en værdi man indsætter i sin funktion*

Call: *Når man beder Python om at evaluere ens funktion*

Return statement: *Det vi beder vores funktion om at gøre når vi evaluere den*

Return: *når vi vil have at funktion skal returnere et svar når vi evaluere den*

Eksempel:

```
function_name(argument)
```

```
>>> def omkreds_trekant(side1, side2, side3):
```

```
    return side1 + side2 + side3
```

Den her funktion har 3 argumenter og returnere summen af dem

Noter til video 4

Logiske operator og deres hieraki

not: giver den modsatte boolean værdi.

and: giver True hvis begge udsagn er sande, ellers False

Or: giver True hvis én af udsagnene er sande og False hvis begge er falske

<u>expr</u>	<u>not expr</u>
True	False
False	True

<u>expr1</u>	<u>expr2</u>	<u>expr1 and expr2</u>
True	True	True
True	False	False
False	True	False
False	False	False

<u>expr1</u>	<u>expr2</u>	<u>expr1 or expr2</u>
True	True	True
True	False	True
False	True	True
False	False	False

En IF statement kan se sådan her ud:

```
File Edit Format Run Options Window Help
def elev_status(time_start, elev_ankomst):
    ''' (number, number) -> str

    Return the status (i god tid, til tiden, forsent) of the
    arrival time of a student who arrived at elev_ankomst to
    a class that starts at time_start

    Pre-condition: 0.0 <= time_start < 24 and 0.0 <= elev_ankomst < 24

    >>> elev_status(9.0, 9.0)
    'til tiden'
    >>> elev_status(9.0, 8.5)
    'i god tid'
    >>> elev_status(9.0, 9.75)
    'forsent'
    '''
    if elev_ankomst < time_start:
        return 'i god tid'
    elif elev_ankomst == time_start:
        return 'til tiden'
    else:
        return 'forsent'
```

Et WHILE loop kan se sådan her ud:

```
>>> num = 2
>>> while num < 100:
    num = num *2
    print(num)

4
8
16
32
64
128
>>> num = 10
>>> while num < 100:
    num = num *2
    print(num)

20
40
80
160
>>>
```

Typer I Python:

Integer (int): *hele tal*

Float (float): *tal med decimaler*

String (str): *tekst strenge, markeres med ' eller " omkring. Pythong behandler dem som værdier og man kan derfor bruge operanter på dem*

Boolean (bool): *True/False udsagn*