



In deze les maken we kennis met de voorwaardelijke opdrachten.

Doelen :

- Het gebruik van de opdracht if ... else
- Wat is de syntax voor deze opdracht in Python.

In elke programmeertaal is er wel een voorwaardelijke opdracht.

Zo'n opdracht heeft de vorm: als ... dan

Als een bepaalde voorwaarde geldt **dan** moet er iets gebeuren.

Vaak kan er nog een extra regel aan toegevoegd worden wat er anders moet gebeuren.

```

If conditie :
    Instructie A
else :
    Instructie B
    
```

Tip voor de docent: In Python bestaat geen opdracht om het eind van een reeks instructies aan te geven.

In plaats daarvan moeten alle regels die voor deze voorwaarde gelden dezelfde inspringing hebben.

Om te testen of twee variabelen gelijk zijn gebruik je in Python een dubbel = teken (dus: a==b)

Voorbeeld:

Een autoverhuurbedrijf heeft de volgende aanbieding:

De huurprijs van een auto is € 66,- voor de eerste 70 kilometer plus nog € 0,25 voor elke kilometer die extra gereden wordt.

We maken in de Python editor een functie die de prijs berekent voor het aantal gereden kilometers.

Open een new Python programma en geef het een naam.

Kies nu F1(Fns...) en **Func** voor het functiesjabloon.

Noem de functie bijvoorbeeld bedrag(k)

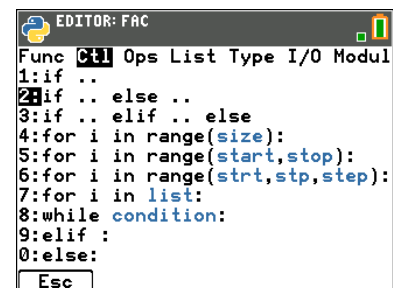


Zet de cursor in de tweede regel achter de twee stippen.

Kies nu F1(Fns...) en **Ctrl**

Je ziet hier een menu met allerlei sjablonen voor programmastructuren.

We kiezen 2:if .. else ..





Je krijgt op het scherm de juiste structuur te zien met de nodige inspringingen. In Python geldt de regel dat alle opdrachten met dezelfde inspringing bij elkaar horen. Dit hoort bij de syntax van de taal. Het geeft tegelijkertijd een overzichtelijk beeld van de structuur van het programma. Let weer op de dubbele punt aan het eind van de regel met **if** en ook die met **else**. Ook dat is een syntaxeigenschap van Python (Als je die weglaat krijg je een foutmelding.)

```
EDITOR: AUTO
PROGRAM LINE 0002
def bedrag(k):
  if :
  else:
```

Je ziet hiernaast hoe de functie er bijvoorbeeld uit kan zien. Als $k \leq 70$ dan is de uitkomst 66. In het andere geval berekent de functie eerst het aantal kilometers boven de 70 en daarna de kosten. In dit voorbeeld zie je dat een blok (onder else in dit geval) uit meerdere regels kan bestaan. Het \leq teken kun je vinden met **F1(Fns...)** en dan **Ops**. Maar het kan ook met $y \gg$

```
EDITOR: TEST
PROGRAM LINE 0008
def bedrag(k):
  if k<=70:
    a=66
  else:
    k1=k-70
    a=66+0.25*k1
  return a
```

Voer het programma uit met Run. Met de **vars** toets kun je de functie opvragen en er een waarde invullen. Om een nieuwe waarde in te vullen kun je weer op de **vars** toets drukken. Je kan ook met de pijl omhoog de vorige regel weer opnieuw in beeld krijgen en dan de waarde wijzigen.

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running TEST
>>> from TEST import *
>>> bedrag(50)
66
>>> bedrag(150)
86.0
>>> |
```

Tip voor de docent: De spaties aan het begin van de regel heten indent. Je kunt deze invoegen met de spatietoets $f+0$, maar je kunt deze ook met **F3(Tools)** invoegen of verwijderen.