

# Kick off



**BIN-OWE2**

# Studiewijzer

Les	Onderwerp	
	Algemeen	Python
1	Debuggen en testen	Exceptions
2		Matplotlib
3		Datastructuren
4		Regular Expressions
5		Object Oriëntatie
6		Recurisie
7		GUI programmeren

# Doelstellingen

- Aan het eind van deze course kan je
  - Gebruik maken van complexere datastructuren
  - Python gebruiken om grafieken te maken
  - Een GUI maken met Python
- Aan het eind van deze course begrijp je
  - Geavanceerdere concepten binnen Python
  - De beginselen van Object Oriëntatie.

# Studiemateriaal

- Boek: “*Starting Out with Python, third edition*”
- Matplotlib tutorial
- Dive into Python
- Onderwijs Online
- Kom je er niet uit? → Google



# Onderwijs Online

- **Praktijk**
  - Oefen- en afvinkopdrachten
- **Theorie**
  - Studiewijzer
  - Slides
  - Digitaal studiemateriaal

## Terugblik

- Variabelen
- Conditionele controle
- Iteratieve controle
- Beslisstructuren
- Commentaar
- Flowcharts
- Git

# Wat was het leukste onderdeel van Python I?

Conditionele controle

Iteratieve controle

Variabelen

Commentaar

Flowcharts

2D lijsten

Anders

**Start the presentation to activate live content**

If you see this message in presentation mode, install the add-in or get help at [PollEv.com/app](https://PollEv.com/app)

# Wat was het lastigste onderdeel van Python I?

2D lijsten

Conditionele controle

Iteratieve controle

Variabelen

Commentaar

Anders...

Flowcharts

**Start the presentation to activate live content**

If you see this message in presentation mode, install the add-in or get help at [PollEv.com/app](https://PollEv.com/app)





# nen we verbeteren aan informatica?



**Start the presentation to activate live content**

If you see this message in presentation mode, install the add-in or get help at [PollEv.com/app](https://PollEv.com/app)

# Informatica

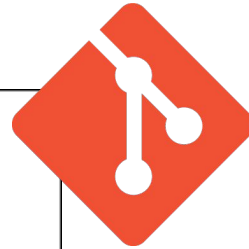
... lets met computers?



# Informatica

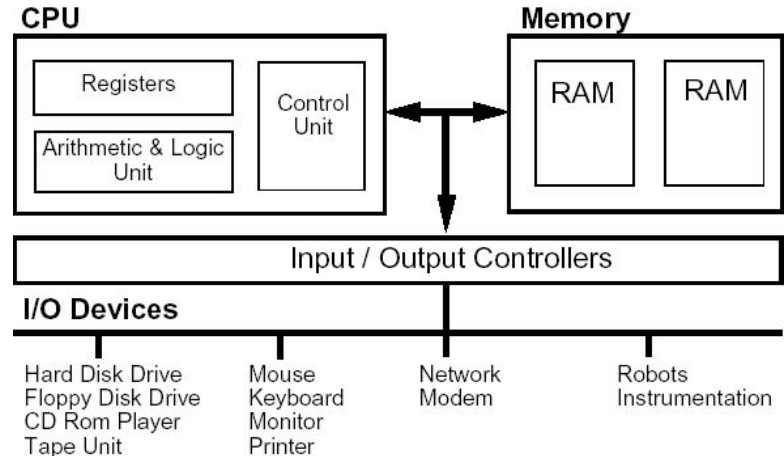
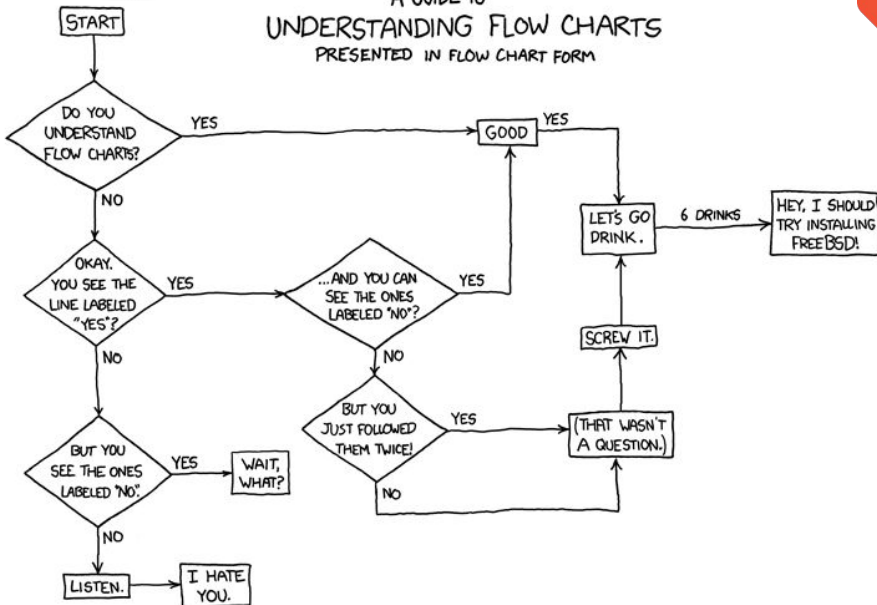
# python™

# Linux™



# git

## A GUIDE TO UNDERSTANDING FLOW CHARTS PRESENTED IN FLOW CHART FORM



## Tentamen Onderdelen

- **Bi2a-Af: Afvinkopdrachten (NV/V)**
  - Moeten voldoende zijn
- **Bi2a-T: Praktische Opdracht (Cijfer)**
  - Programmeeropdracht
  - Dit keer een week de tijd
- **Bi2a-Stat: Statistiek toets (Cijfer)**
  - Meer hierover in de Statistiek lessen

# Studiewijzer

Les	Onderwerp	
	Algemeen	Python
1	Debuggen en testen	Exceptions
2		Matplotlib
3		Datastructuren
4		Regular Expressions
5		Object Oriëntatie
6		Recurisie
7		GUI programmeren



# Doelstellingen

- Aan het eind van deze les kan je
  - Een eenvoudig testscript opstellen
  - De Python debugger vinden
- Aan het eind van deze les begrijp je
  - Waarom het handig is om gebruik te maken van debuggen en testen

## Debuggen vs testen

- Debugging is het zoeken naar fouten in een programma dat wordt ontwikkeld. Deze fouten zijn bekend.
- Testen is het zoeken naar onbekende fouten. Verondersteld wordt dat het programma juist werkt.



# Testing

- Er zijn professionele testers
- Deze testen software op specificatie
- Bijvoorbeeld in de farmaceutische industrie is dit extreem belangrijk
- Testers maken ook gebruik van testtools

## Niveau van correctheid

- Geen syntax errors
- Geen semantische errors
- Er bestaat test data waar het programma met het juiste antwoord komt
- Het programma geeft een juist antwoord op willekeurige data
- Programma geeft correcte antwoorden met complexe test data
- Alle legale input leidt tot een juist antwoord
- Alle legale en alle foute input leidt tot een juist antwoord

# Voorbeeld test script

## Test Case: Modify Item

**Description:** This test case simulates one of the actions a stock adjuster would perform each day. The user will search for item by Item ID, and then modify the item description.

### Data Requirements:

{Username} – User must have update privileges. User name must be unique (as application does not allow simultaneous logins).

{Password} – must be valid for given {Username}

{ItemID} – any item that is currently in stock may be used. Items should be selected at random. Use the following SQL query: "select item\_id from items where quantity > 0".

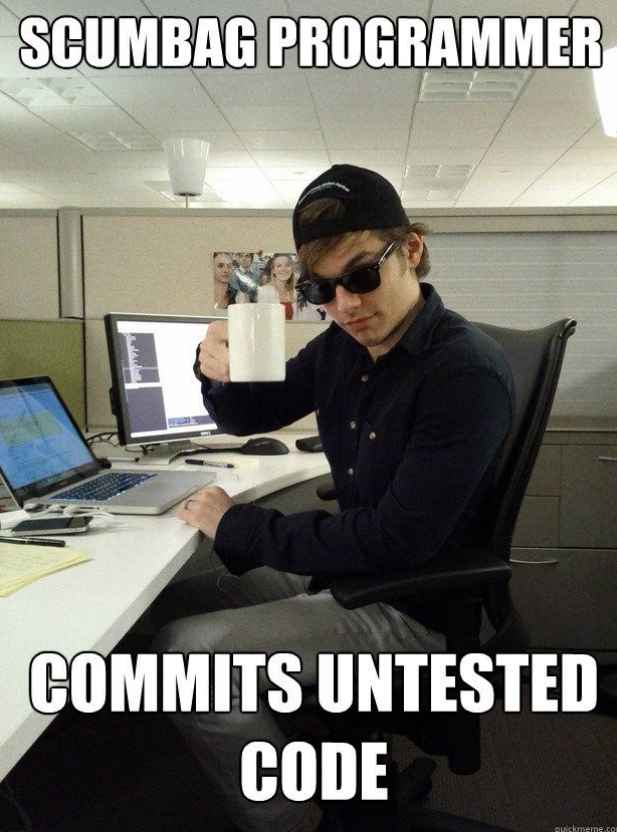
Note: if two users open the same item for modification and one saves the item. The second user will not be able to save their changes.

{ItemDescription} – The new item description should be the same as the old item description, except with "modified" added to the end.

Step Number	Step Description	Expected Result	Transaction Name	User Think Time
01	Invoke application from desktop icon. Log in with {Username} and {Password}	Main menu screen is displayed	user_login	5
02	Select item search from menu.	Search screen is displayed	select_search	2
03	Enter {ItemID} in Exact Find field. Press Search button.	Item properties screen is displayed.	search_by_item_id	10
04	Press Edit button	Item for specified {ItemID} is displayed in edit mode.	press_edit	1
05	Modify the {ItemDescription} in the Description field. Press Save button.	Item properties screen is displayed	modify_description	20
06	Press Main Menu button	Main menu screen is displayed.	return_to_main_menu	5
	Return to step 02 and repeat.			

# Eenvoudiger voorbeeld

<b>Test:</b> Afvinkopdracht 6 test 1	
<b><i>Omschrijving</i></b>	Het programma starten zonder dat het bestand aanwezig is
<b><i>Verwacht resultaat</i></b>	Het programma geeft een nette afhandeling van de error, geeft aan dat het bestand afwezig is.
<b><i>Pass/fail</i></b>	Pass
<b><i>Opmerkingen</i></b>	Gegeven error bericht: "Het bestand kan niet gevonden worden. Zet een geldig fasta bestand in dezelfde map als waar u het programma vanuit draait en probeer het dan opnieuw."



# Python debugger

<https://docs.python.org/3.5/library/pdb.html>

## Voorbeeld code

```
# experiment with Python debugger
import pdb
a = "aaa"
pdb.set_trace()
b = "bbb"
c = "ccc"
final = a + b + c
print (final)
```

## Samenvatting

**Debuggen en testen is best wel  
belangrijk voor software ontwikkeling**



