

# Technische handleiding SuperPy

Zie ook de Gebruikershandleiding SuperPy

---

## Inhoud

Technische handleiding SuperPy.....	1
Technische opmerkingen.....	2
Beginsituatie .....	2
Datumformaat .....	2
Gegevensbestanden in csv-formaat.....	2
Rich .....	2
Taal .....	2
Bevindingen.....	3
Functies .....	4

---

**In geel:** nieuw of aangepast in deze versie.

Onderstaand, in alfabetische volgorde, een aantal [technische opmerkingen](#).

En aan het eind een opsomming van bekende [bevindingen](#) die nog verwerkt moeten worden.

## Technische opmerkingen

### Beginsituatie

- Alle csv-bestanden bevatten 1 rij met kolomnamen, van elkaar gescheiden door een komma.
- Bestand SuperPy.csv bevat onder de kolomrij een tweede rij met de in SuperPy te hanteren “dag van vandaag”.
- De beginsituatie voor het eigen concreet uitgewerkt voorbeeld is in te stellen via de Command line interface (CLI). Type in **python super.py initiate own**  
Zie de Gebruikershandleiding voor dit voorbeeld.

### Datumformaat

- In de csv-bestanden en in de CLI worden datums getoond in ISO-formaat yyyy-dd-mm.

### Gegevensbestanden in csv-formaat

- Het bestand Bought.csv en Sold.csv bevat in de eerste kolom resp. het gegeven bought\_id en sold\_id, een integer die begint bij 1 en door het programma steeds met 1 wordt opgehoogd.
- Om mogelijk te maken dat de SuperPy-gegevens eenvoudig met andere programma's of interfaces gedeeld kunnen worden zijn die altijd opgeslagen in csv-bestanden. Na elk commando door de gebruiker in command line interface ingevoerd of na elke bewerking door SuperPy zelf worden de cs-bestanden daarom bijgewerkt op nieuw opgebouwd.
- 

### Rich

- Aan SuperPy toegevoegd en ingezet op belangrijke boodschappen aan de gebruiker te benadrukken door die in rood en vet te presenteren.
- Dit wordt bijvoorbeeld gebruikt wanneer getracht wordt een niet-bestaande beginsituatie te initialiseren:  

```
Sorry, 'startsituation = owns' is not a correct value. Correct value is: own.
```

  
En ook als de opgevraagde *revenu* of *profit* negatief is.

### Taal

- De beide handleidingen zijn in het Nederlands.
- De producten, tabelnamen, CLI-commando's en python-code (de toelichting in de code, de boodschappen aan de gebruiker) zijn in het Engels.

## Bevindingen

Nog te doen / mogelijke verbeteringen.

- Import van modules via `*` haalt veel meer op dan nodig is.
- Opmaak van de output van de csv-bestanden in functie `def show_csv(csv_name:str):` verbeteren.
- Voor bijna alle functies 1 of meerdere test-functies maken, incl. `assert` en gebruik `pytest`.

Functies

Onderdeel	Functienaam	Beschrijving	Input	Output	Onderdeel	Test
my_csv_handler.py	read_mytoday()->datetime:	read my_today from Today.csv	n.a.	return my_today_datetime	test_csv_handler.py	
my_csv_handler.py	modify_mytoday(shift_in_days:int)->datetime:	writing a adjusted my_today in SuperPy.csv	shift_in_days:int	my_todate changed	test_csv_handler.py	
my_csv_handler.py	next_id(csv_name:str)->int:	counting lines in {csv_name}	csv_name:str	return next_id	test_csv_handler.py	def test1_next_id():
my_csv_handler.py	show_csv(csv_name:str):	showing table content {csv_name} to user.	csv_name:str	table content showed	test_csv_handler.py	
my_csv_handler.py	append_csv(csv_name:str, new_row:list):	row {new_row} will be appended to {csv_name}.	csv_name:str, new_row:list	row added	test_csv_handler.py	def test1_appends_new_line():
my_csv_handler.py	make_csv_inventory_and_expired():	making a new Inventory.csv and Expired.csv	n.a.	tables Inventory.csv and expired.csv new filled	test_csv_handler.py	
my_initiate.py	initiate():	all csv-files are reset to startvalue	startsituation:str	print('Finished: SuperPy has been initiated.')	test_initiate.py	
my_functions.py	bought_id_sold(find_bought_id:int)->bool:	Checks if bought_id {find_bought_id} is sold.	find_bought_id:int	return message	test_my_functions.py	
my_functions.py	bought_id_expired(find_bought_id:int)->bool:	Checks if bought_id = {find_bought_id} is expired.	find_bought_id:int	return message	test_my_functions.py	
my_functions.py	product_id_in_csv_inventory(find_product_name:str):	Searches the product_id of {product_name} in Inventory.csv.	find_product_name:str	return found_product_id	test_my_functions.py	
my_functions.py	find_bought_id(find_product_name:str):	Searches bought_id for {find_product_name} in Bought.csv	find_product_name:str	return found_product_id	test_my_functions.py	
my_functions.py	total_sold_price(date:datetime):	Calculates the total_sold_price up to and including the specified date	date:datetime	return total_sold_price	test_my_functions.py	
my_functions.py	total_bought_price(date:datetime):	Calculates the total_bought_price up to and including the specified date	date:datetime	return total_bought_price	test_my_functions.py	
my_functions.py	total_expired_price(date:datetime):	Calculates the total_expired_price up to and including the specified date	date:datetime	return total_expired_price	test_my_functions.py	
my_functions.py	revenue(date:datetime):	In SuperPy 'revenue' is defined as: the total price of all (and only) SOLD products	date:datetime	return revenue	test_my_functions.py	
my_functions.py	profit(date:datetime):	In SuperPy 'profit' is defined as: the total price of (all sold products - bought products - expired products).	date:datetime	return profit	test_my_functions.py	
my_functions.py	this_date_total_sold_price(date:datetime):	Calculates total_sold_price on the specified date	date:datetime	return this_date_total_bought_price		
my_functions.py	this_date_total_bought_price(date:datetime):	Calculates the total_bought_price on the specified date	date:datetime	return this_date_total_bought_price		
my_functions.py	this_date_total_expired_price(date:datetime):	Calculates the total_expired_price on the specified date	date:datetime	return this_date_total_expired_price		
my_functions.py	this_date_revenue(date:datetime):	In SuperPy 'revenue' is defined as: the total price of all (and only) SOLD products	date:datetime	return this_date_revenue	test_my_functions.py	
my_functions.py	this_date_profit(date:datetime):	In SuperPy 'profit' is defined as: the total price of (all sold products - bought products - expired products).	date:datetime	return this_date_profit	test_my_functions.py	