

BIO-INFORMATICA

OWE2A PYTHON II

PROGRAMMEREN IN PYTHON II

CURRICULUM OVERZICHT

	Periode 1	Periode 2	Periode 3	Periode 4
1e jaar	Programmeren in Python (1)	Programmeren in Python (2)	Informatiesystemen met Python	Geautomatiseerde identificatie van eiwitten via sequentievergelijking
	Metabolisme	Opsporen van genetische mutaties bij erfelijke ziektes	Vergelijkende genomanalyse: evolutie van virussen	Sequentie alignment (BLAST) en functionele eiwit analyse
2e jaar	Programmeren in Java	Datastructuren en algoritmen in Java	Analyse en ontwerptechnieken	Webtechnologie en textmining
	Proteomics: eiwitstructuren, eiwitfuncties, scheidingstechnieken en data analyse	Transcriptomics: data analyse mbt regulatie van metabole routes	Genomics: annoteren van genomisch DNA	Moleculaire fylogenie: evolutie, multiple sequence alignment mbt signaaltransductie
3e jaar	Stage of minor		Data Mining en grid computing	RNA seq, biostatistiek,
			Eiwitten: carcinogenese en eiwitstructuren	Workflows
4e jaar	Stage, afstudeerstage of minor		Minor of afstudeerstage	

JAAR 1

Basis van programmeren in Python

Complexe structuren

Databases voor opslag van data

	Periode 1	Periode 2	Periode 3	Periode 4
1e jaar	Programmeren in Python (1)	Programmeren in Python (2)	Informatiesystemen met Python	Geautomatiseerde identificatie van eiwitten via sequentievergelijking
	Metabolisme	Opsporen van genetische mutaties bij erfelijke ziektes	Vergelijkende genoomanalyse: evolutie van virussen	Sequentie alignment (BLAST) en functionele eiwit analyse

Ontwikkelen van applicaties voor biologisch analyse en visualisatie op het web

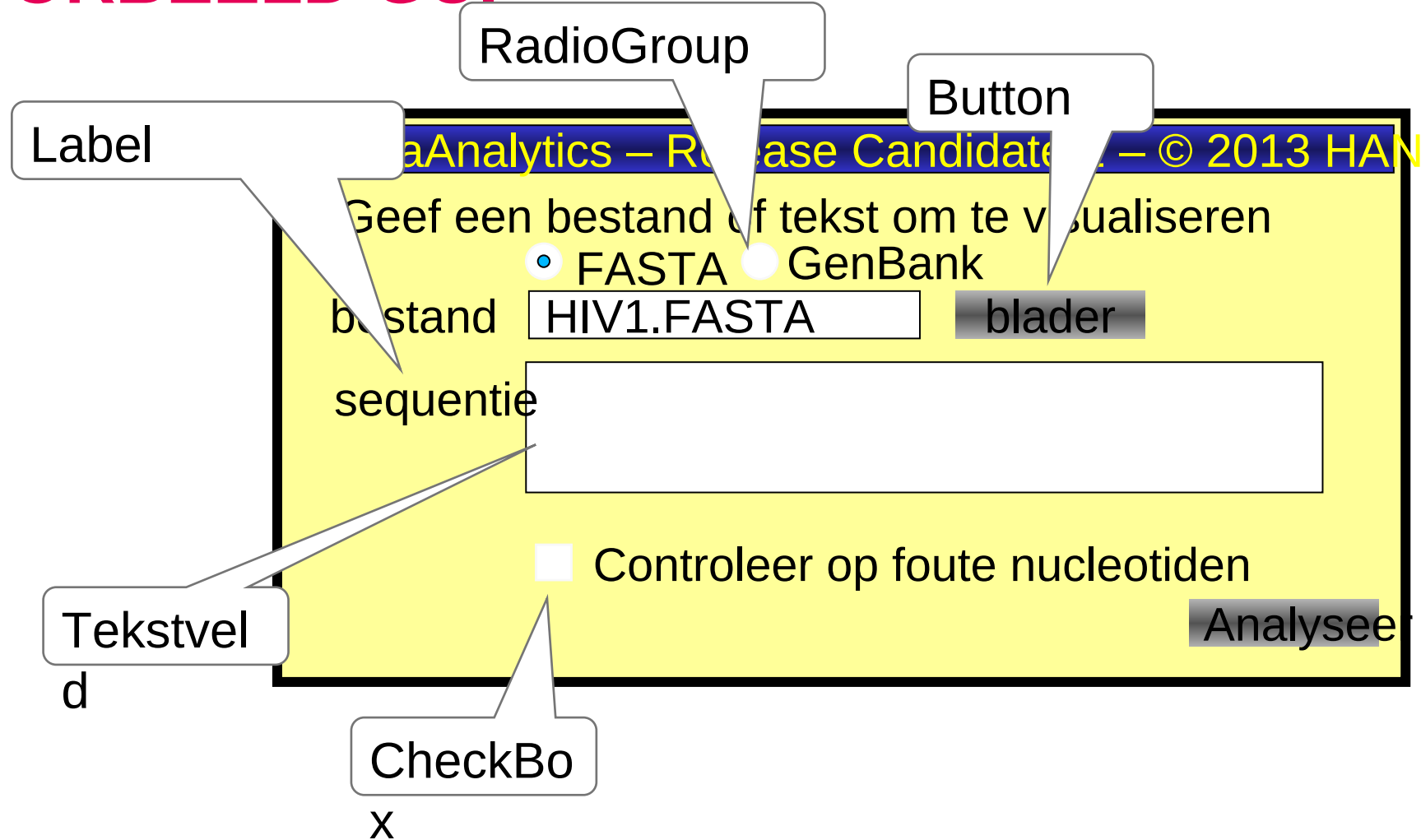
OVERZICHT OWE 1A INFORMATICA

Wee k	Onderwerpen	
1	IDE: <u>PyCharm</u> Debuggen en testen PEP richtlijnen	<u>PEP tutorial</u>
2	Grafieken met Matplotlib	<u>Matplotlib.org</u>
3	Datastructuren	H9 SowP
4	Regular Expressions	<u>H4/H5 DiP</u>
5	Object Oriëntatie	H10 & H11 SowP
6	Recursie	H12 SowP
7	Graphical User Interface	H13 SowP

AGENDA

- Graphical User Interfaces
- Using tkinter
- Text and Label
- Frames
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

VOORBEELD GUI



Eigen werk auteur

WAT DOET EEN GUI

- Een GUI is event driven, er zit geen vaste volgorde in de events
- De Python code die we tot nu toe schreven was data driven

DATA DRIVEN VS. EVENT DRIVEN

- Een **data driven** applicatie heeft (eventueel) input van gebruikers maar doorloopt een standaard flow
- Een **event driven** applicatie heeft buttons en andere input items en heeft geen vastgelegde flow voor dataverwerking Een flowchart tekenen voor een GUI is dus niet mogelijk

AGENDA

- Graphical User Interfaces
- Using tkinter
- Text and Label
- Frames
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

CODE VOOR HELLO WORLD!

Import van GUI functionaliteit

```
import tkinter
```

GUI's zijn bijna altijd object georiënteerd

```
class MyGUI:
```

```
    def __init__(self):
```

```
        self.main_window = tkinter.Tk()
```

```
        self.label = tkinter.Label(self.main_window, text='Hello World!')
```

```
        self.label.pack()
```

```
        tkinter.mainloop()
```

```
my_gui = MyGUI()
```

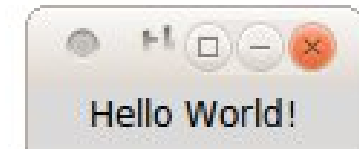
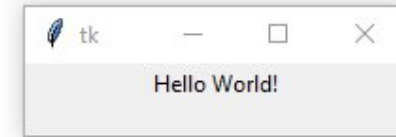
Creatie van een GUI object

Loop tot exit

Pack zorgt voor plaatsing

Maak een window

Gebruik de init methode voor creatie van het window



Creatie van een label, tekst voor in de GUI

AGENDA

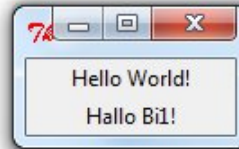
- Graphical User Interfaces
- Using tkinter
- Text and Label
- Frames
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

TEXT EN LABEL

```
import tkinter

class MyGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.label = tkinter.Label(self.main_window, text='Hello World!')
        self.label.pack()
        self.label2 = tkinter.Label(self.main_window, text='Hallo Bi1!')
        self.label2.pack()
        tkinter.mainloop()

my_gui = MyGUI()
```



PARAMETERISEREN VAN LABELS

```
import tkinter

class MyGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.top_frame = tkinter.Frame(self.main_window)
        self.bottom_frame = tkinter.Frame(self.main_window)
        self.label1 = tkinter.Label(self.top_frame, text='Hello World!', \
                                     bg="red", relief="sunken")

        self.label1.pack(side="top")
        self.label2 = tkinter.Label(self.bottom_frame, text='Hallo Bi!', \
                                     relief="raised", bg="yellow", \
                                     fg="blue")

        self.label2.pack(side="top")
        self.top_frame.pack()
        self.bottom_frame.pack()
        tkinter.mainloop()

my_gui = MyGUI()
```



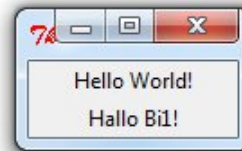
AGENDA

- Graphical User Interfaces
- Using tkinter
- Text and Label
- **Frames**
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

FRAMES

```
import tkinter

class MyGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.top_frame = tkinter.Frame(self.main_window)
        self.bottom_frame = tkinter.Frame(self.main_window)
        self.label1 = tkinter.Label(self.top_frame, text='Hello World!')
        self.label1.pack(side="top")
        self.label2 = tkinter.Label(self.bottom_frame, text='Hallo Bi1!')
        self.label2.pack(side="top")
        self.top_frame.pack()
        self.bottom_frame.pack()
        tkinter.mainloop()
```



```
my_gui = MyGUI()
```

<http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>

Online referentie van tkinter

Eigen werk auteur

AGENDA

- Graphical User Interfaces
- Using tkinter
- Text and Label
- Frames
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

BUTTON MET DIALOOG

```
import tkinter

class MyGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.label = tkinter.Label(self.main_window, text='Hello World!')
        self.label.pack()
        self.my_button = tkinter.Button(self.main_window, text='klik!', \
                                         command=self.doe_iets)

        self.my_button.pack()
        tkinter.mainloop()

    def doe_iets(self):
        tkinter.messagebox.showinfo('response', 'Bedankt voor het klikken!')

my_gui = MyGUI()
```



FILEDIALOG

```
import tkinter as tk
from tkinter import ttk
from tkinter import filedialog as fd
from tkinter.messagebox import showinfo

# create the root window
root = tk.Tk()
root.title('Tkinter Open File Dialog')
root.resizable(False, False)
root.geometry('300x150')
```

```
def select_file():
    filetypes = (
        ('text files', '*.txt'),
        ('All files', '*.*')
    )

    filename = fd.askopenfilename(
        title='Open a file',
        initialdir='/',
        filetypes=filetypes)

    showinfo(
        title='Selected File',
        message=filename
    )
```

```
# open button
open_button = ttk.Button(
    root,
    text='Open a File',
    command=select_file
)

open_button.pack(expand=True)

# run the application
root.mainloop()
```

AGENDA

- Graphical User Interfaces
- Using tkinter
- Text and Label
- Frames
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

ENTRY WIDGET

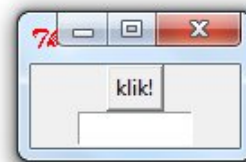
```
import tkinter

class MyGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.frame = tkinter.Frame(self.main_window)
        self.my_button = tkinter.Button(self.main_window, text='klik!', \
                                         command=self.doe_iets)

        self.my_button.pack()
        self.invoer = tkinter.Entry (self.main_window,width=10)
        self.invoer.pack()
        self.frame.pack()
        tkinter.mainloop()

    def doe_iets(self):
        tekst = self.invoer.get()
        tkinter.messagebox.showinfo('response', 'Invoer is '+tekst)

my_gui = MyGUI()
```



AGENDA

- Graphical User Interfaces
- Using tkinter
- Text and Label
- Frames
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

```
import tkinter

class MyGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.frame = tkinter.Frame(self.main_window)
        self.my_button = tkinter.Button(self.main_window, text='klik!', \
                                         command=self.doe_iets)

        self.my_button.pack()
        self.invoer = tkinter.Entry (self.main_window,width=10)
        self.invoer.pack()

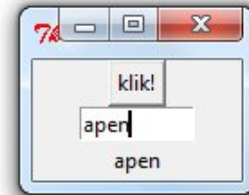
        self.value = tkinter.StringVar()
        self.value_label = tkinter.Label(self.main_window, \
                                         textvariable=self.value)

        self.value_label.pack()

        self.frame.pack()
        tkinter.mainloop()

    def doe_iets(self):
        tekst = self.invoer.get()
        self.value.set(tekst)

my_gui = MyGUI()
```



AGENDA

- Graphical User Interfaces
- Using tkinter
- Text and Label
- Frames
- Buttons en Dialogen
- Input met Entry Widget
- Labels als output
- Radio Buttons en Check Box

```

import tkinter

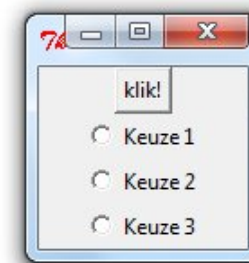
class MyGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.frame = tkinter.Frame(self.main_window)
        self.my_button = tkinter.Button(self.main_window, text='klik!', \
                                         command=self.doe_iets)

        self.my_button.pack()
        self.radio_var = tkinter.IntVar()
        self.rb1 = tkinter.Radiobutton(self.frame, \
                                       text='Keuze 1', variable=self.radio_var, value=1)
        self.rb2 = tkinter.Radiobutton(self.frame, \
                                       text='Keuze 2', variable=self.radio_var, value=2)
        self.rb3 = tkinter.Radiobutton(self.frame, \
                                       text='Keuze 3', variable=self.radio_var, value=3)

        self.rb1.pack()
        self.rb2.pack()
        self.rb3.pack()
        self.frame.pack()
        tkinter.mainloop()

    def doe_iets(self):
        tekst = str(self.radio_var.get())
        tkinter.messagebox.showinfo('response', 'Invoer is ' + tekst)

```



my_gui = MyGUI()
Eigen werk auteur

```
import tkinter
```

```
class MyGUI:
```

```
    def __init__(self):
```

```
        self.main_window = tkinter.Tk()
```

```
        self.frame = tkinter.Frame(self.main_window)
```

```
        self.my_button = tkinter.Button(self.main_window, text='klik!', \
                                         command=self.doe_iets)
```

```
        self.my_button.pack()
```

```
        self.cb_var1 = tkinter.IntVar()
```

```
        self.cb_var2 = tkinter.IntVar()
```

```
        self.cb_var3 = tkinter.IntVar()
```

```
        self.cb_var1.set(0)
```

```
        self.cb_var2.set(0)
```

```
        self.cb_var3.set(0)
```

```
        self.cb1 = tkinter.Checkbutton(self.frame, \
                                         text='Keuze 1', variable=self.cb_var1)
```

```
        self.cb2 = tkinter.Checkbutton(self.frame, \
                                         text='Keuze 2', variable=self.cb_var2)
```

```
        self.cb3 = tkinter.Checkbutton(self.frame, \
                                         text='Keuze 3', variable=self.cb_var3)
```

```
        self.cb1.pack()
```

```
        self.cb2.pack()
```

```
        self.cb3.pack()
```

```
        self.frame.pack()
```

```
        tkinter.mainloop()
```



Eigen werk auteur

CHECKBOX

```
def doe_iets(self):  
    self.message = "Keuzes:\n"  
    if self.cb_var1.get() == 1:  
        self.message = self.message + '1\n'  
    if self.cb_var2.get() == 1:  
        self.message = self.message + '2\n'  
    if self.cb_var3.get() == 1:  
        self.message = self.message + '3\n'  
    tkinter.messagebox.showinfo('response', self.message)
```

SAMENVATTING

- GUI programming draait om Event Driven applicaties
- In Python is het mogelijk om op verschillende manieren GUI's te bouwen
- In Python kunnen we gebruik maken van tkinter

SAMENVATTING

De basis GUI elementen:

- Window
- Frame
- Label
- Tekst Entry
- Dialogen
- RadioGroup
- CheckBox

VERANTWOORDING

- In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen
- Alle teksten © Martijn van der Bruggen/Teuntje Peeters/HAN tenzij expliciet externe bronnen zijn aangegeven
- Screenshots op basis van eigen werk auteur en/of vernoemde sites
- Eventuele images zijn opgenomen met vermelding van bron