



code4girls

ERSTES PROGRAMMIEREN FÜR  
MÄDCHEN AB KLASSE 8

<http://www.bildung.koeln.de/code4girls>

code4girls



Patricia



Lisa

NICE TO MEET YOU!

# Zeitplan

Do 31.03

14:00

Vorstellungsrunde - 30

14:30

Einleitung und Command-Line - 30

15:00

Yay Python - Geisterspiel - 30

15:30 - 15:45

Pause - 15

15:45

Scratch - 60

16:45 - 17:00

Pause - 15

17:00

Scratch-Quiz - 60

18:00

Genug für heute

# code4girls

Fr 01.04

09:30

Python Vokabeln lernen - 60

10:30 - 10:45

Pause

10:45

Python Horoskop Code - 30

11:15

Git, GitHub, Flask, HTML, CSS

12:30 - 13:30

Pause

13:30

Java Script, eigene App

14:30

Pythonanywhere: Deploy and enjoy

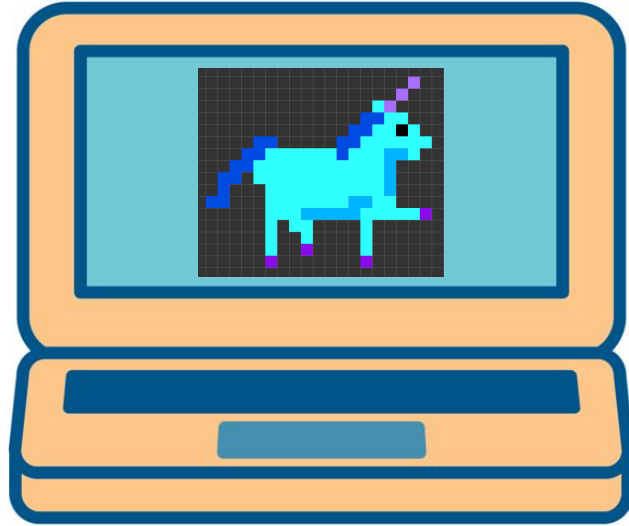
15:00

Abschluss: Wie weiter lernen?

15:30

Feierabend

code4girls



**“Software is eating the world.”**

MARC ANDREESSEN, The Wall Street Journal, August 20, 2011



**Wo überall begegnen Euch Dinge, die jemand programmiert hat?**



Wie groß ist der Anteil von Frauen in IT-Berufen?



## Wie groß ist der Anteil von Frauen in IT-Berufen?

2014 Branchenverband Bitkom:

[https://infogr.am/frauen\\_in\\_der\\_it](https://infogr.am/frauen_in_der_it)

- 60.000 IT-Stellen nicht besetzt

Während der technologiegetriebene Wandel die gesellschaftliche Entwicklung in großem Maße bestimmt, haben Frauen kaum aktiven Anteil an der Gestaltung dieses Wandels.





## Command Line

```
Whoami  
ip addr show  
Pwd  
Ls  
cd Desktop  
mkdir test  
cd ..  
rm -r test  
exit
```





**Und jetzt mal was Cooles :)**

```
mkdir code4girls
```

```
cd code4girls
```

```
git clone https://github.com/code4girls/marie.git
```

```
cd marie      ls
```

```
display marie_0.png
```

Git = Versionskontrolle

GitHub = Plattform, um zusammen zu arbeiten, Code auszutauschen

```
convert -delay 60 -loop 0 *.png mariechen.gif
```

```
iceweasel mariechen.gif
```



## ERSTE SCHRITTE PYTHON

```
$ python exit() $python3
>>>
print("Hello world!"), print('Hello Lisa!')
5 + 3
print(5+3)
print(2478956857895789*1545785869578996)
5/20
'Hallo'+' '+'Welt'
print('Hallo ' * 20)
a = 'Hello'      b='World'      print(a,b)
name = "Patricia" name print(name)
import this
exit()
```

Yay  
Python



```
$ touch laenge.py
```

```
-----
```

```
print('Wie lang ist dein Name?')
```

```
name = input('Bitte gib deinen Namen ein: \n')
```

```
def laenge_name(name):
```

```
    laenge = len(name)
```

```
    print(laenge)
```

```
laenge_name(name)
```

```
-----
```

```
$ mv laenge.py ~/Code4Girls
```



## Geisterspiel

```
git clone https://github.com/code4girls/geisterspiel.git
```

```
cd geisterspiel
```

```
ls
```

```
python3 geisterspiel_0.py
```

```
gedit geisterspiel_0.py
```

code4girls

Pause!



Pause!

# code4girls

## SCRATCH

<https://scratch.mit.edu/>

Anmelden

Benutzer: code4girls1-13

Passwort: adalovelace





## **SCRATCH**

**Erstes Scratch Programm:  
Katze sagt: Hello World**

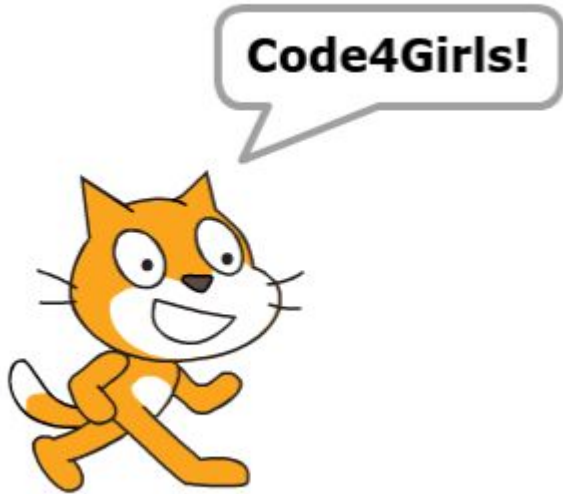


## SCRATCH

**Aufgabe: Katze macht was (mit Maus?)**

**10 min Zeit, um mal zu schauen, was man mit Scratch noch so machen kann**

**Was könnte die Katze noch tun?  
Könnte auch mal jemand anderes als die Katze etwas tun?**



# code4girls

**Katze ist ein** **Objekt**  
**Katze hat** **Eigenschaften**  
**Katze hat** **Funktionen**

Katze

-----

farbe

größe

kostüm

sage()

gehe()

spiele klang(miau)



## **SCRATCH**

**“Hello World” ist ein String**

**Für x Schritte = x ist Variable**

**Entscheidungen treffen : Falls, dann, sonst**

**Schleifen: Solange, wiederhole fortlaufend**



## **Aufgabe: Programmiererinnen-Quiz**

**Marissa Mayers**

**Ada Lovelace**

**Grace Hopper**

**Jean Bartik**

**Googeln, lesen, Quizfragen ausdenken, in Spreadsheet eintragen:**

<https://goo.gl/LJADsR>

Quelle: <http://www.frauen-informatik-geschichte.de/index.php?id=6>

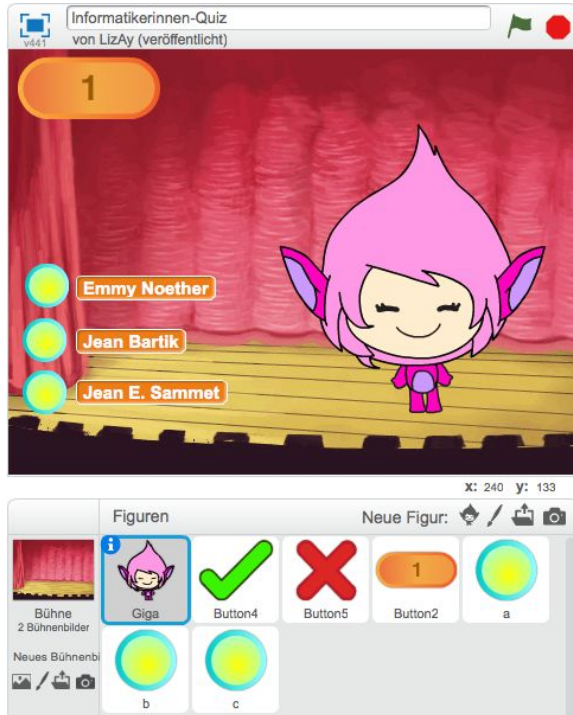


**Welche Objekte benötigen wir für ein Quiz?**

**Welche Eigenschaften oder Funktionen haben die Objekte?**

# code4girls

## Objekte zusammenstellen



## Funktionen hinzufügen





**Was ist Open Source?**

**Projekt veröffentlichen**

**Fremdes Projekt remixen**

**Informatikerinnen-Quiz:**

<https://scratch.mit.edu/projects/87354704/#editor>

code4girls

Pause!



Pause!





### **Listen in Python:**

```
>>> liste = ['eins', 'zwei', ,dreil']
```

ansprechbar über Index beginnend mit 0:

```
>>> liste [0]
```

```
,eins'
```

---

### **Wörterbuch in Python:**

```
>>> buch = {,sun': ,Sonne', ,moon': ,Mond'}
```

Zugriff durch Schlüsselwort (key):

```
>>> buch[,sun']
```

```
,Sonne'
```



**Python**

**Aufgabe:**

**Wir schreiben ein Programm, mit dem man Vokabeln lernen kann**

# code4girls

- Kombination aus list und dict
- Zähler ,fehler‘ zurücksetzen
- Liste aller keys in Variable ,vokabeln‘ übertragen
  - Solange etwas in ,vokabeln‘ drin steht (Schleife)
  - choice = Funktion zur zufälligen Auswahl aus einer Liste = wort
  - Frage stellen
  - Antwort entgegennehmen
  - Bedingung prüfen mit if
  - Wenn richtig, dann Wort aus dict löschen
  - Wenn falsch, dann richtige Antwort geben und Fehler hoch zählen (auch fehler = fehler + 1)
- Sobald Schleife beendet, gebe Fehler aus

```
from random import *
deutsch = {'key': ['Schlüssel', 'Taste'],
          'slice': ['Scheibe', 'Stück'],
          'value': ['Wert'],
          'sun': ['Sonne'],
          'moon': ['Mond'],
          'star': ['Stern']}

fehler = 0
vokabeln = list(deutsch.keys())
while vokabeln:
    wort = choice(vokabeln)
    print('Was heit', wort, 'auf Deutsch')
    antwort = input('---> ')
    if antwort in deutsch[word]:
        print('Richtig!')
        vokabeln.remove(word)
    else:
        print('Leider falsch.')
        print(word.capitalize(), 'bedeutet: ')
        for w in deutsch[word]:
            print(w)
        print()
        fehler += 1
print('Alle Vokabeln gelernt.')
print('Du hast', fehler, 'Fehler gemacht.')
```



## Python

### Aufgabe: Vokabeln lernen - umschreiben

Vokabelliste Latein: <https://goo.gl/Ints3x>

code4girls

Pause!

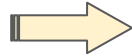


Pause!

## Python

### Aufgabe: Horoskop-App

Es könnte sein, dass am...  
Sei nicht überrascht, wenn am...  
Es ist gut möglich, dass am...  
Die Sterne sagen, dass am...



Montag  
Dienstag  
Mittwoch  
Donnerstag  
Freitag  
Samstag  
Sonntag



etwas Wunderbares passiert.  
ein interessanter Mensch in dein Leben tritt.  
eine gute Gelegenheit auf dich wartet.

- Erstellen von drei Listen mit den Textblöcken
- Variable 'horoskop' mit drei zufällig ausgewählten Elementen der drei Listen
- Überschrift ausgeben
- Zufällig erstellten Satz ausgeben

```
from random import choice
A = ['Es könnte sein, dass ',
     'Sei nicht überrascht, wenn ',
     'Es ist gut möglich, dass ',
     'Die Sterne sagen, dass ']
B = ['im nächsten Monat ',
     'diese Woche ',
     'am Mittwoch ',
     'im Zoo ',
     'auf dem Heimweg ',
     'durch eine übernatürliche Erscheinung ',
     'heute ']
C = ['etwas wunderbares passiert.',
     'ein interessanter Mensch in dein Leben tritt.',
     'eine gute Gelegenheit auf dich wartet.',
     'Kekse auf dich herab regnen.',
     'ein Pappnasenc clown dein Herz erobert.',
     'ein Papagei dich Eierkopf nennt.

horoskop = choice(A) + choice(B) + choice(C)
print('Dein Horoskop')
print(horoskop)
```

code4girls

Pause!



Pause!



# code4girls

**Eigene Programme auf dem eigenen Computer = ½ Spaß**

**Eigene Programme als Anwendung im Internet = YAY!**





```
git clone https://github.com/code4girls/code4girls.git
```

## Hello World - Flask App:

Öffnen: `gedit 1.py`

```
python3 1.py
```

```
str+C
```



```
cd 2
```

```
ls
```

```
gedit views.py
```

**> Nutzt jetzt ein Template**

```
cd templates
```

```
gedit simple.html
```



**Hello HTML!**

**HTML - Hyper Text Markup language**

```
gedit html.html
```

```
gedit views.py
```

```
return render_template('html.html')
```

**views.py**

```
@app.route('/test')
```

```
def test():
```

```
    return render_template('html.html')
```

**[http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)**



## Geht's auch dynamischer?

**Umschreiben:**

**In simple.html**

```
<h1>Welcome to Code4Girls on Flask!</h1>
    <p>Your lucky number is:</p>
    <p>{{ lucky_num }}</p>
```

**In views.py**

```
return render_template('simple.html', lucky_num= 7)
```

**Dann in views.py**

```
return render_template('simple.html', lucky_num= randint(1,100))
```



**Wie lang ist dein Name?**

**In: simple.html**

```
<h1>Hello {{ name }}!</h1>
```

```
<p>So viele Buchstaben hat dein Name?</p>
```

```
<p> {{ length }} </p>
```

**In: views.py**

```
@app.route('/<name>/')
```

```
def name_length(name):
```

```
    length = len(name)
```

```
    return render_template('name.html', name=name.capitalize(),
```

```
    length=length)
```



**Horoskop im Web**

**Schreibt horoskop.html**

**und get\_horoskop in views.py**



## horoskop.html

```
<html>
```

```
<body>
```

```
<h1>Code4Girls Horoskop!</h1>
```

```
<p>Dein Horoskop:</p>
```

```
<p> {{ horoskop }} </p>
```

```
<p>Deine Glückszahl:</p>
```

```
<p> {{ lucky_num }} </p>
```

```
</body>
```

```
</html>
```





## Views.py

```
@app.route('/horoskop/')  
def get_horoskop():  
    a = ['In deinen Sternen steht, dass ', 'Auf dem Grunde der Kaffeetasse ist zu  
lesen, dass ', 'Das Orakel sagt, dass ']  
    b = ['Du bald ', 'Du für immer und ewig ', 'Du in deinem Leben ']  
    c = ['glücklich wirst.', 'Liebe findest.', 'reich und berühmt wirst.']  
    horoskop = choice(a)+choice(b)+choice(c)  
    lucky_num = randint(1, 100)  
    return render_template('horoskop.html', horoskop=horoskop,  
lucky_num=lucky_num)
```



## CSC Cascading style sheets - Makeup for websites

HTML war NIEMALS dafür gedacht, Webseiten mit Tags zu formatieren.

HTML **beschreibt den Inhalt** einer Webseite:

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

CSS ist dafür da, das Aussehen / die Style- Informationen einer Webseite zu beschreiben.

Diese Style-Definitionen sind normalerweise in einer externen Datei gespeichert: external.css

Mit einer externen CSS-Datei kannst Du das komplette Aussehen einer Webseite ändern, indem Du nur diese eine Datei änderst.

[http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp)



Selector

Declaration

Declaration

h1

{color:blue; font-size:12px;}

Property

Value

Property

Value

Selectors: by tag

```
All <p>:
p {
  text-align: center;
  color: red;
}
```

Selectors: by id

Just one special <p>:

```
<p id="para1">Hello World!
</p>

#para1 {
  text-align: center;
  color: red;
}
```

Selectors: by class

```
<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
```

```
.center {
  text-align: center;
  color: red;
}
```

6/css.html

6/mystyle.css

font-family: 'Lobster';

/css\_inline.html



Design = Harte Arbeit -> Hello Bootstrap :)

- Bootstrap ist ein **freies front-end framework** um schneller und einfacher Webseiten zu entwickeln
- Bootstrap enthält Design-Vorlagen auf der Basis von **HTML and CSS** für Schriften, Formulare, Buttons, Tabellen, Navigation, Bilder und viele andere, so wie optionale JavaScript plugins
- Bootstrap bietet zudem die Möglichkeit einfach **responsive Designs** zu erstellen

6/bootstrap.html 6/bootstrap\_1.html

[http://www.w3schools.com/bootstrap/bootstrap\\_get\\_started.asp](http://www.w3schools.com/bootstrap/bootstrap_get_started.asp)



## 7 app / templates

base.html      # Bootstrap hinzugefügt, nutzt ein base-template und blocks

Simple.html    # {% extends 'base.html' %}, blocks

```
cd 7/  
python run.py
```

## Web 2.0 - Input bitte! Auf dem Weg zu Social Media



### Forms /9

Requirements.txt      # installing, what is needed

```
sudo pip3 install -r requirements.txt
```

config.py              # CSRF - Security Cross-Site-Request-Forgery

\_\_init.py\_\_            # Nutzt config

forms.py               # Form anlegen

hrsk.html              # Lässt Namen eingeben

dein\_hrsk.html        # Nutzt Namen und generiert Horoskop und Glückszahl

views.py               # verknüpft alles



## Hello JavaScript

JavaScript ist eine der **3 Sprachen** des WWW::

1. **HTML** definiert den Inhalt von Webseiten
2. **CSS** beschreibt das Layout von Webseiten
3. **JavaScript** steuert das Verhalten von Webseiten

**/8/js.html**

**/8/js\_hello.html**

**/8/light\_js.html**

[http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp)

code4girls

Pause!



Pause!





## Aufgabe:

**mkdir** **patricia**

```
cp -r 9/* patricia
```

**# kopiere alle Inhalte von /9 nach /patricia**

**Rumspielen!**

**Ändert HTML, CSS**

**Fügt für @app.route('/name/') HTML, CSS ein**

**Fügt @app.route('/geist/') oder @app.route('/vokabeln/')**



**Fertig?**

**git status      # rot**

**git add .**

**git status      # grün**

**git commit -m 'Patricia fügt ihre Version hinzu'**

**git push origin master**

**Code4girls**

**adalovelace1815**



## DEPLOY on Python Anywhere

**code4girls**<sup>1</sup>

**adalovelace**

Consoles: bash

```
git clone https://github.com/code4girls/code4girls.git
```

```
Cd code4girls
```

```
cd patricia
```

```
virtualenv --python=python3.4 myvenv
```

```
source myvenv/bin/activate
```

```
cd ..
```

```
pip install -r requirements.txt
```



## Publishing our web app

**Web** tab. Finally, hit **Add a new web app: OK, MaueI configuration, Python 3.4,**

**Virtualenv:** /home/code4girls1/code4girls/patricia/myenv/

**code4girls1\_pythonanywhere\_com\_wsgi.py**

```
import sys
```

```
path = '/home/code4girls1/code4girls/patricia/' # change to 9 later
```

```
if path not in sys.path:
```

```
    sys.path.append(path)
```

```
    from app import app as application
```



Hit **Save** and then go back to the **Web** tab.

We're all done!

Hit the big green **Reload** button and you'll be able to go view your application. You'll find a link to it at the top of the page.

# code4girls

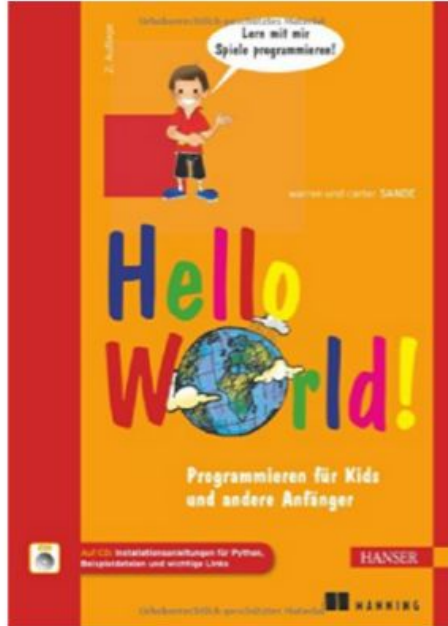
**Mehr lernen:**



**Programmieren supereasy:**  
**Einfacher Einstieg in**  
**SCRATCH und PYTHON**

# code4girls

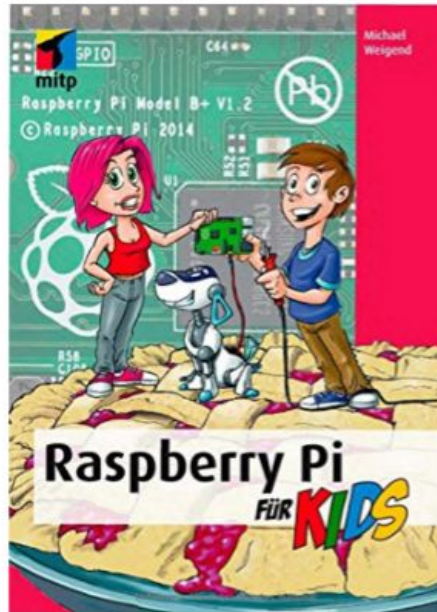
**Mehr lernen:**



[Hello World!: Programmieren für Kids und andere Anfänger](#)

# code4girls

**Mehr lernen:**



**Raspberry Pi für Kids**



# Python and Django

Youtube: [Coding is for girls](#)

[Django Girls](#)

# Coding lessons for beginners of all ages

- [Scratch ode to code](#): Multilingual Scratch tutorial for Europe Code Week
- [Codecademy](#): Learn to code interactively, for free, on the web.
- [Code School](#): Code School teaches web technologies in the comfort of your browser with video lessons, coding challenges, and screencasts.
- [Code Avengers](#): Learn to build websites, apps and games with HTML, CSS and JavaScript.
- [Code.org Tutorials](#): Simple tutorials for beginners that can be completed in an hour or less.
- [Computer Science Unplugged](#): A collection of free learning activities for the classroom or home that teaches Computer Science through engaging games and puzzles that use cards, string, crayons and lots of running around.
- [Angry Birds Fun Learning](#): Learn to code the fun way! Discover fun coding apps and courses for various difficulty levels.

# Ausbildung

Fachinformatiker/in - Anwendungsentwicklung

Fachinformatiker/in - Systemintegration

# **BA / Studium**

**Duales Studium:**

Informatik oder Wirtschaftsinformatik

**Studium**

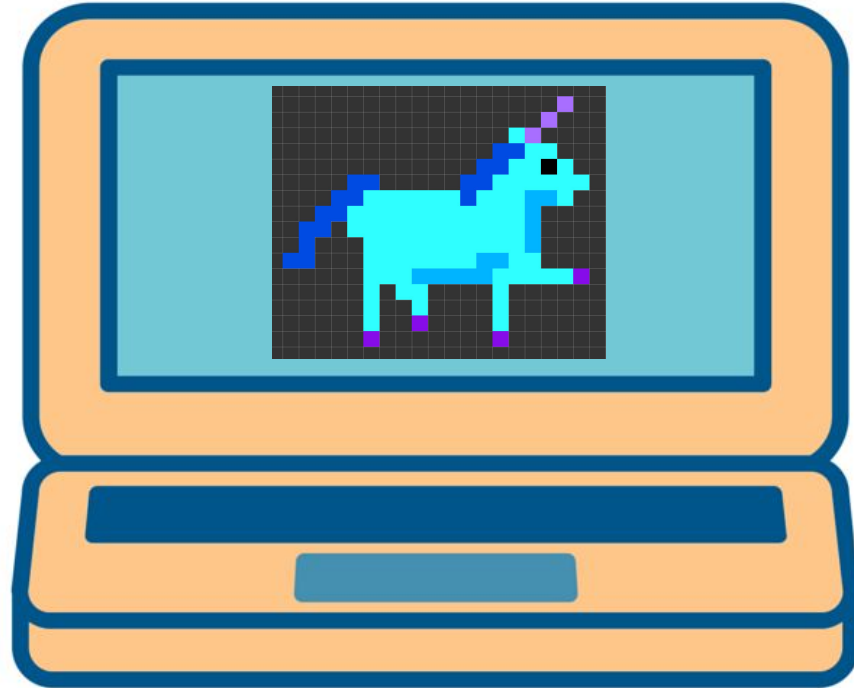
Medieninformatik z.B. in Köln

Informatik z.B. TH Köln- Studieren probieren

Informatik Uni Köln

code4girls

Dankeschön.  
Feedback?



Tschüs!

The logo for 'code4girls' is displayed in a white, cursive script font against a solid teal rectangular background.

code4girls

**YAY PYTHON**

**<http://pythonanywhere.com/>**

**benutzername code4girls1 - code4girls13**

**passwort adalovelace**