

WEBPROGRAMMIERUNG DATEN AUS DEM WEB AUTOMATISIERT VERARBEITEN

Martin Guggisberg

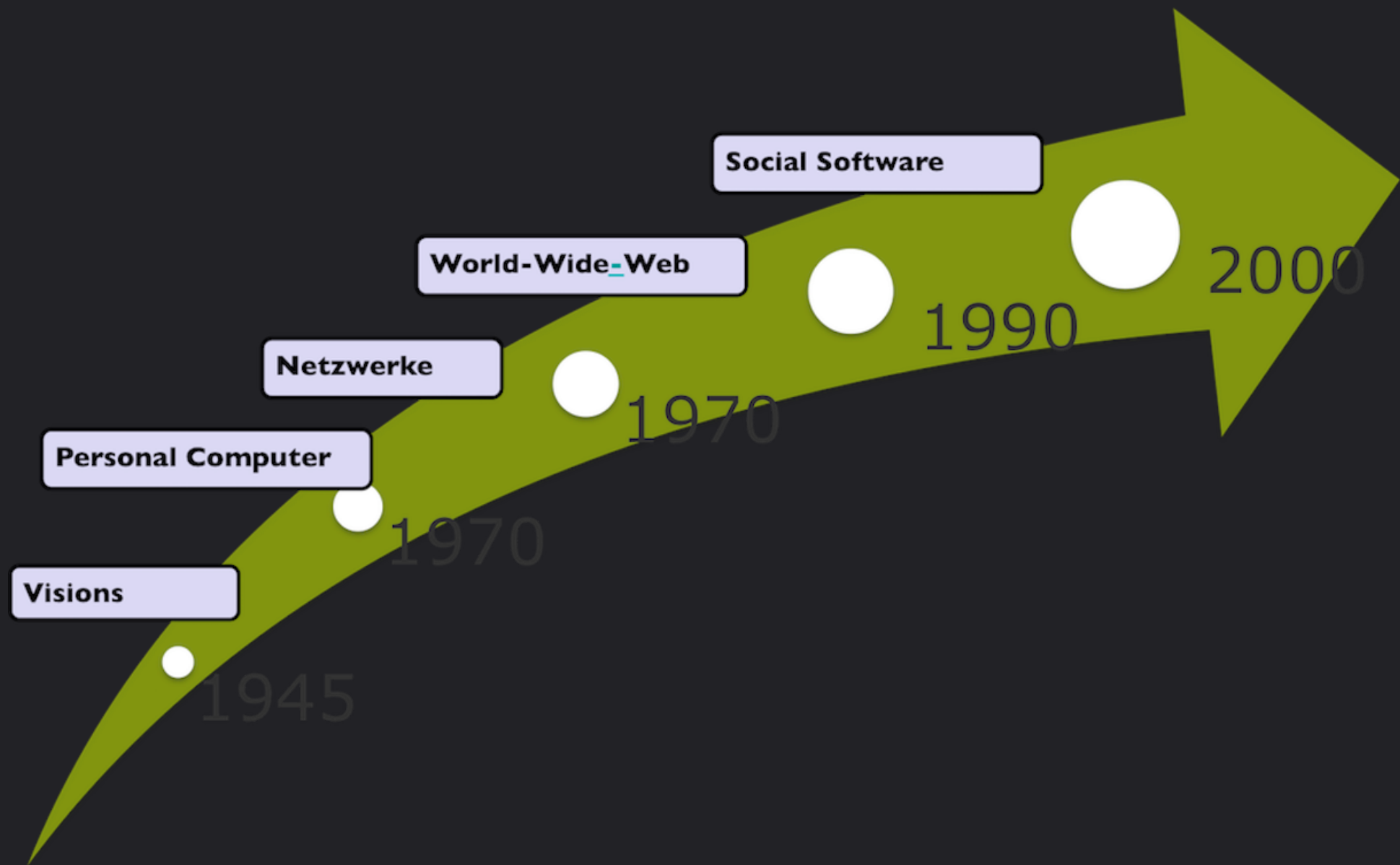
5.9.2015

Weiterbildung: Programmieren im Unterricht mit Python

AGENDA

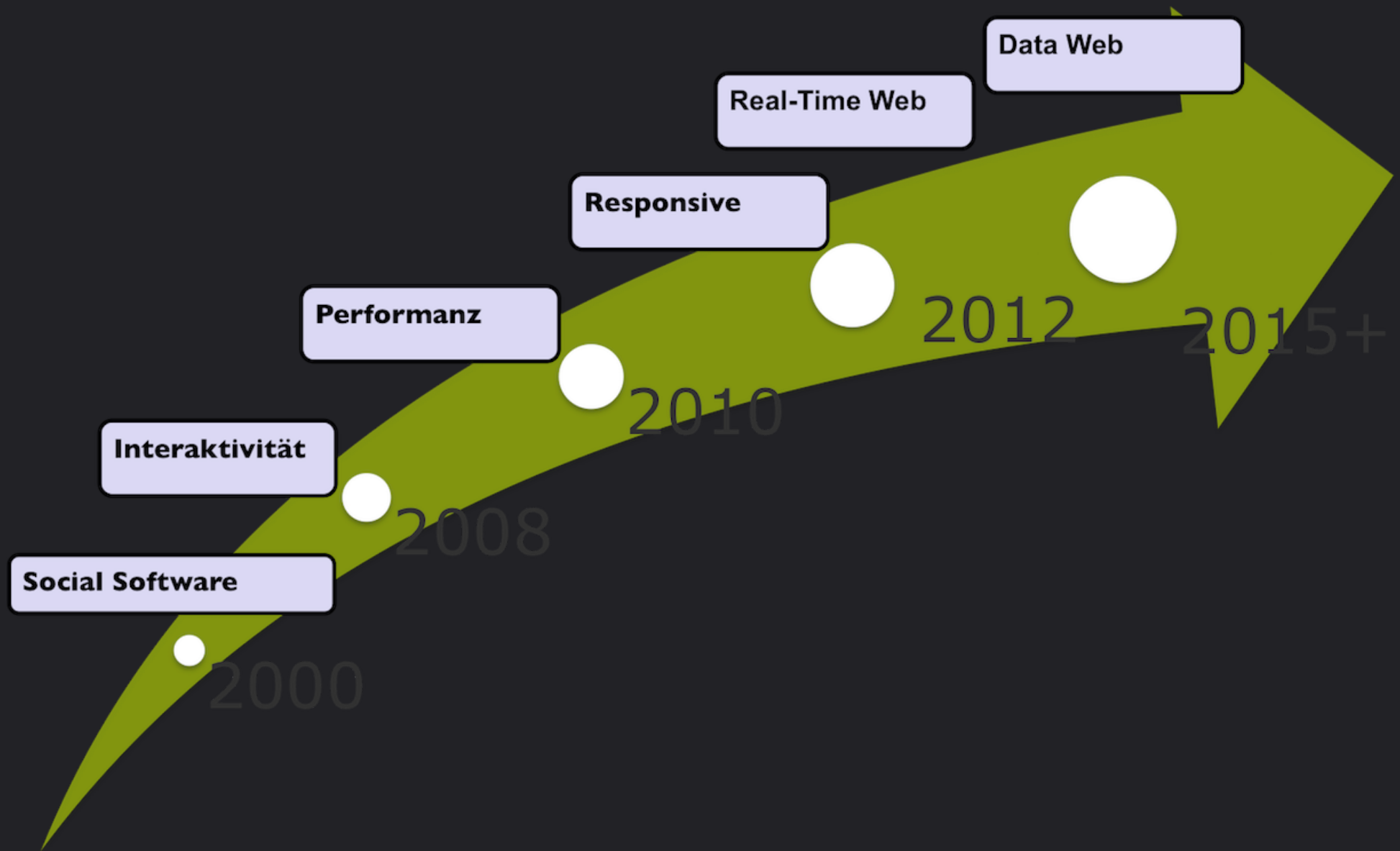
- Fakten und Grundlagen zum Web
- Strukturierte Daten (JSON) anfragen und auswerten
- Webseiten parsen und Daten sammeln

WEB: VON DER VERGANGENHEIT ZUR GEGENWART



Grafik von H. Burkhart aus Veranstaltung Web Data Management HS 2014

WEB: VON DER VERGANGENHEIT ZUR GEGENWART II



Grafik von H. Burkhart aus Veranstaltung Web Data Management HS 2014

ONLINE-INFORMATIONEN ZUM WEB

- [Web Platform Docs](#)
- [Webtechnologien für Entwickler](#)
- [W3C Standards](#)
- [DIVE INTO HTML5](#)
- [Codecademy HTML CSS](#)

WEBPROGRAMMIERUNG

WELCHE PROGRAMMIERSPRACHE ?

Front-End:

- ~~Adobe Flash~~
- ~~Java~~
- **JavaScript**

Back-End:

- Java (J2EE, Business)
- PHP (hiphop Compiler, FB)
- **Python** (Django, Zope2, Flask)
- JavaScript (nodeJs)

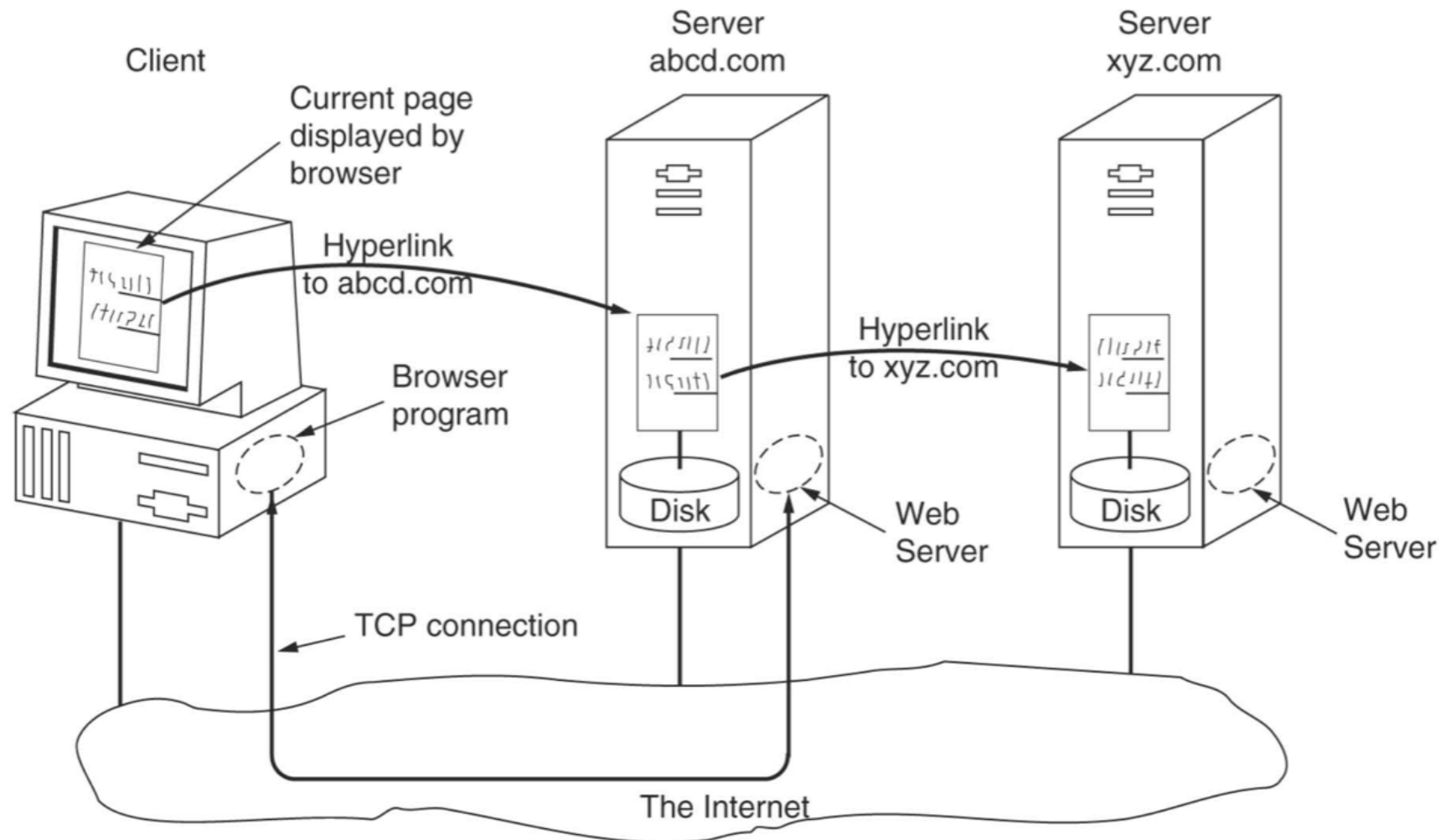
DIESER BLOCK FOKUSSIERT AUF
DIE EXTRAHIERUNG VON DATEN AUS DEM
WEB

EIN TEILGEBIET VON
DATA SCIENCE

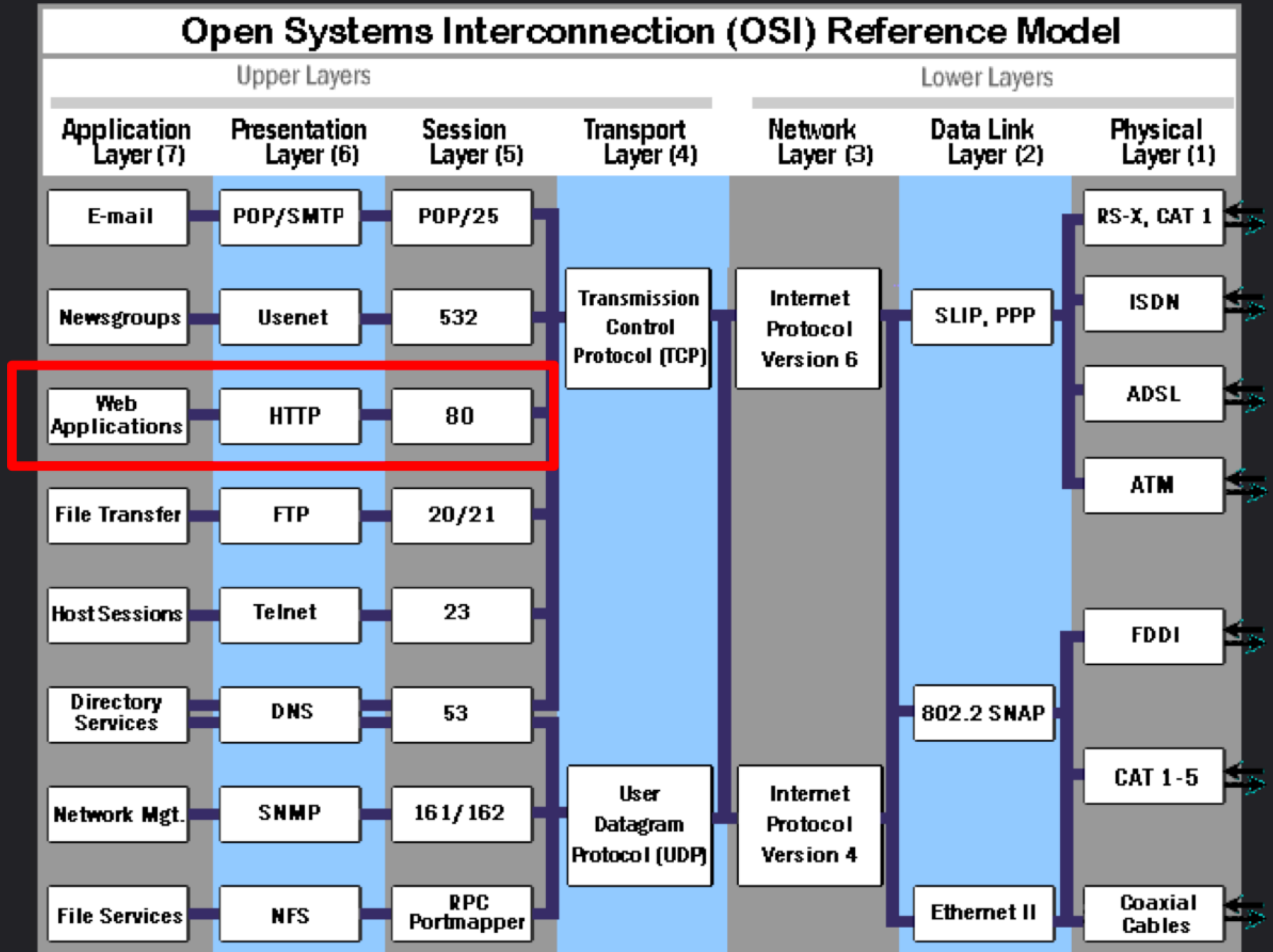
CATCH THE FISH



WEB ARCHITEKTUR



OSI-REFERENZ MODELL



HYPertext TRAnSFER PROTOCOL (HTTP)

- HTTP basiert auf dem **Frage - Antwort - Prinzip**
- HTTP ist zustandslos
- HTTP Kommunikation verläuft über TCP/IP Sockets

<http://www.w3.org/Protocols/>

TYPISCHE FRAGE (REQUEST)

```
GET /index.html HTTP/1.1
Host: www.unibas.ch
User-Agent: Mozilla/4.0
```

ANTWORT (RESPONSE)

```
HTTP/1.1 200 OK
Content-Length: 2579
Content-Type: text/html
```

```
<!doctype html>
<html lang="de">
  <head>
    . . .
  </head>
</html>
```

HTTP ANFRAGE (HTTPLIB)

http_ex_1.py

```
from httplib import HTTPConnection

conn = HTTPConnection("www.tigerjython.ch")

conn.request("GET", "/index.html")
res = conn.getresponse()
print res.status, res.reason

for header in res.getheaders():
    print header[0] + " : " + header[1]

conn.close()
```

HTTP ANFRAGE (**URLLIB2**)

[http_ex_2.py](#)

```
from urllib2 import urlopen

conn = urlopen("http://www.tigerjython.ch")
status = conn.getcode()
reason = conn.msg

print status, reason
print con.headers

conn.close()
```

RESPONSE HEADER

```
200, "OK"  
Date: Thu, 27 Aug 2015 15:52:06 GMT  
Server: Apache  
Last-Modified: Sat, 26 Jul 2014 16:05:28 GMT  
ETag: "129000000003925-e1-4ff1adbc7fa0e"  
Accept-Ranges: bytes  
Content-Length: 225  
Content-Type: text/html  
Age: 354  
X-Cache: HIT from login.fdxteneded.com  
Via: 1.0 login.fdxteneded.com (squid/3.0.STABLE20)  
Proxy-Connection: close
```

WAS IST NUN
DIE ANTWORT ?

ANTWORT

http_ex 3.py

```
from urllib2 import urlopen
endpoint = "http://www.tigerjython.ch"
response = urlopen(endpoint)
html = response.read()
print html
```

```
<html>
<body>
<meta http-equiv="Content-Type" content="text/html; c
<meta HTTP-EQUIV="REFRESH" content="0; url=index.php?
</body>
</html>
```

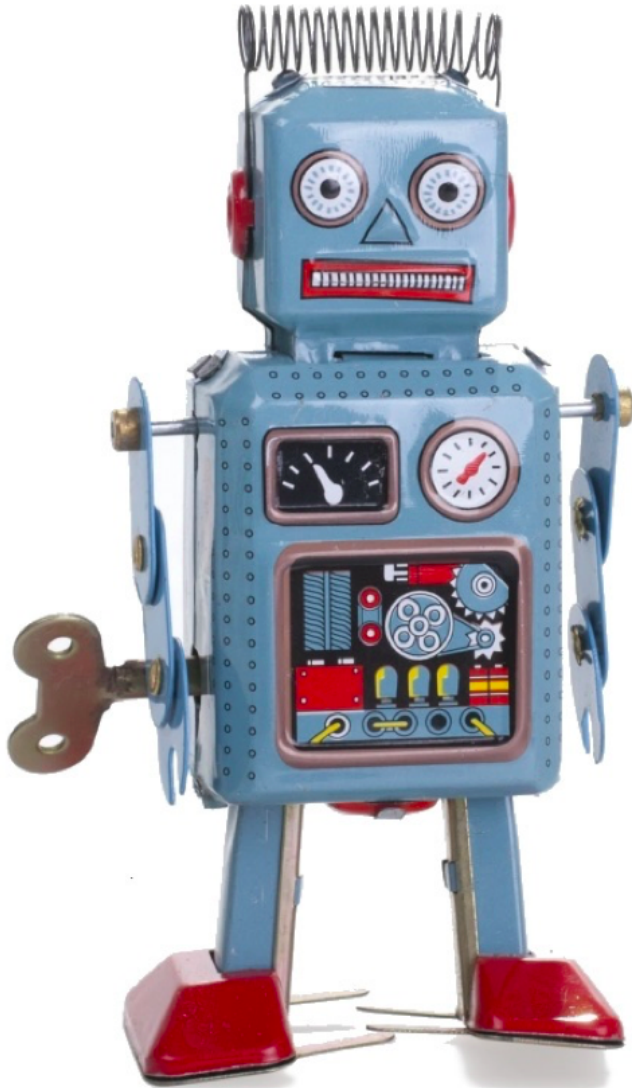
SPRACHEN FÜR DIE DATEN- UND DOKUMENTSPEZIFIKATION

- XML
- HTML
- JSON

JSON

JSON

(JAVASCRIPT OBJECT NOTATION)



- einfach
- plattformunabhängig
- maschinenlesbar

BEISPIEL

```
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York"},
  "phoneNumber": [
    {
      "type": "home",
      "number": "212 555-1234"},
    {
      "type": "fax",
      "number": "646 555-4567"}
  ]
}
```

GOOGLE SEARCH

TigerJython Beispiel aus dem Modul Internet

```
import urllib2, json
search = input("Enter a search ...")
url = "http://ajax.googleapis.com/ajax/services/search
responseStr = urllib2.urlopen(url).read()
response = json.loads(responseStr)
responseData = response["responseData"]
results = responseData["results"]

for result in results:
    title = result["title"]
    url = result["url"]
    print title + " **** " + url
```

Welche Geo-Koordinaten hat die Adresse:

Universitätstrasse 6 ETH Zürich ?

[geolocation.py](#)

```
import urllib2
import pprint
import json
add = "Universitätstrasse 6 ETH Zürich"
add = urllib2.quote(add.encode("utf-8"))
geocode_url = "http://maps.googleapis.com/maps/api/ge
req = urllib2.urlopen(geocode_url)
jsonResponse = json.loads(req.read())
pprint.pprint(jsonResponse)
```

REVERSE GEOLOCATION

Ich brauche Information zu der Geo-Koordinate

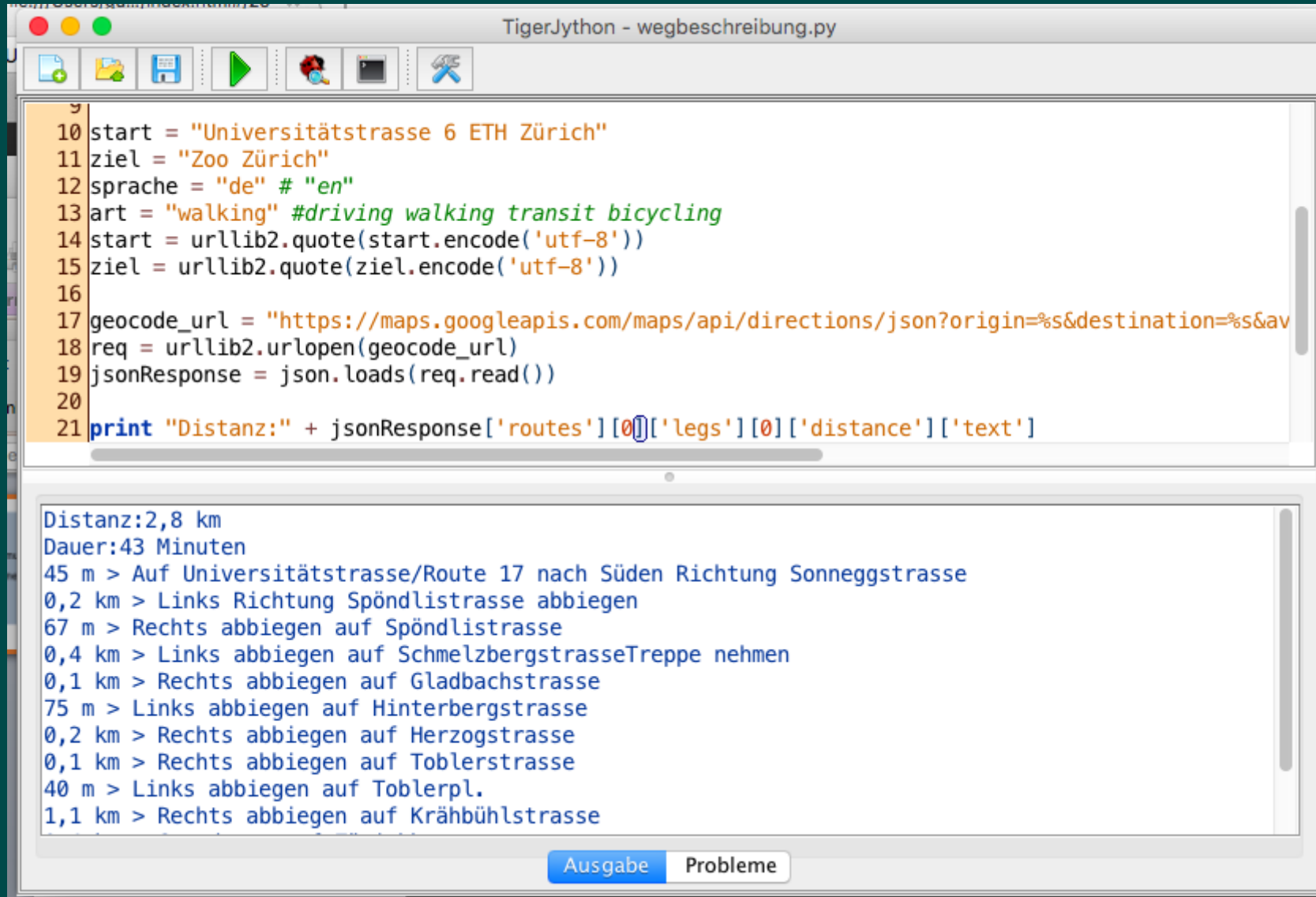
(47.3783606 , 8.5488485)

[reverse_geolocation.py](#)

```
import urllib2
import pprint
import json
lat="47.3783606"
lon="8.5488485"
geocode_url = "http://maps.googleapis.com/maps/api/ge
req = urllib2.urlopen(geocode_url)
jsonResponse = json.loads(req.read())
print jsonResponse["results"][0]["formatted_address"]
```


WIE KOMME ICH VON A NACH B ?

wegbeschreibung.py



The screenshot shows a TigerJython IDE window titled "TigerJython - wegbeschreibung.py". The editor contains a Python script that uses the Google Maps API to calculate a route from "Universitätstrasse 6 ETH Zürich" to "Zoo Zürich". The script sets the language to German, the mode to walking, and constructs a URL to the Google Maps API. It then fetches the JSON response and prints the distance and duration. Below the code editor, the output window displays the calculated route details, including a list of directions with distances and bearings.

```
9
10 start = "Universitätstrasse 6 ETH Zürich"
11 ziel = "Zoo Zürich"
12 sprache = "de" # "en"
13 art = "walking" #driving walking transit bicycling
14 start = urllib2.quote(start.encode('utf-8'))
15 ziel = urllib2.quote(ziel.encode('utf-8'))
16
17 geocode_url = "https://maps.googleapis.com/maps/api/directions/json?origin=%s&destination=%s&av"
18 req = urllib2.urlopen(geocode_url)
19 jsonResponse = json.loads(req.read())
20
21 print "Distanz:" + jsonResponse['routes'][0]['legs'][0]['distance']['text']
```

Distanz:2,8 km
Dauer:43 Minuten
45 m > Auf Universitätstrasse/Route 17 nach Süden Richtung Sonneggstrasse
0,2 km > Links Richtung Spöndlistrasse abbiegen
67 m > Rechts abbiegen auf Spöndlistrasse
0,4 km > Links abbiegen auf SchmelzbergstrasseTreppe nehmen
0,1 km > Rechts abbiegen auf Gladbachstrasse
75 m > Links abbiegen auf Hinterbergstrasse
0,2 km > Rechts abbiegen auf Herzogstrasse
0,1 km > Rechts abbiegen auf Toblerstrasse
40 m > Links abbiegen auf Toblerpl.
1,1 km > Rechts abbiegen auf Krähbühlstrasse

Ausgabe Probleme

WOLFRAM|ALPHA API

Test Wolfram|Alpha API

API Explorer

[API Documentation »](#)



weather tomorrow in zurich



- ☐ Images ☐ Plaintext ☐ Wolfram Language Input
☐ Sound ☒ HTML ☐ Wolfram Language Cells

Submit

Enter a query

Select formats for XML output

[Web version of this result »](#)

<http://api.wolframalpha.com/v2/query?appid=xxx&input=weather%20tomorrow%20in%20zurich&format=html>

API response

```
<?xml version='1.0' encoding='UTF-8'?>
<queryresult success='true'
  error='false'
  numPods='2'
  datatypes='Weather'
  timedout='Data,Character'
  timedoutPods=''
  timing='1.798'
  parsetiming='0.589'
  parsetimedout='false'
  recalculate='http://www4a.wolframalpha.com/api/v2/recalc.jsp?id=MSPa24281g430114667i232g00005a4ah4h9b5df1
id='MSPa24291g430114667i232g00001ed3b40f81765i4b'
  best='http://www4a.wolframalpha.com/
```

XML , HTML

XML,HTML SIND WOHLGEFORMT

- Die **erste Zeile** identifiziert das Dokument als xml,html.
- Es gibt **ein äusserstes Element**, das alle anderen umschliesst.
- **Reguläre Schachtelung**
- Attribute eines Elements sind **eindeutig**.
- Attributwerte stehen in **Anführungszeichen**.

<http://www.w3.org/TR/REC-xml/>

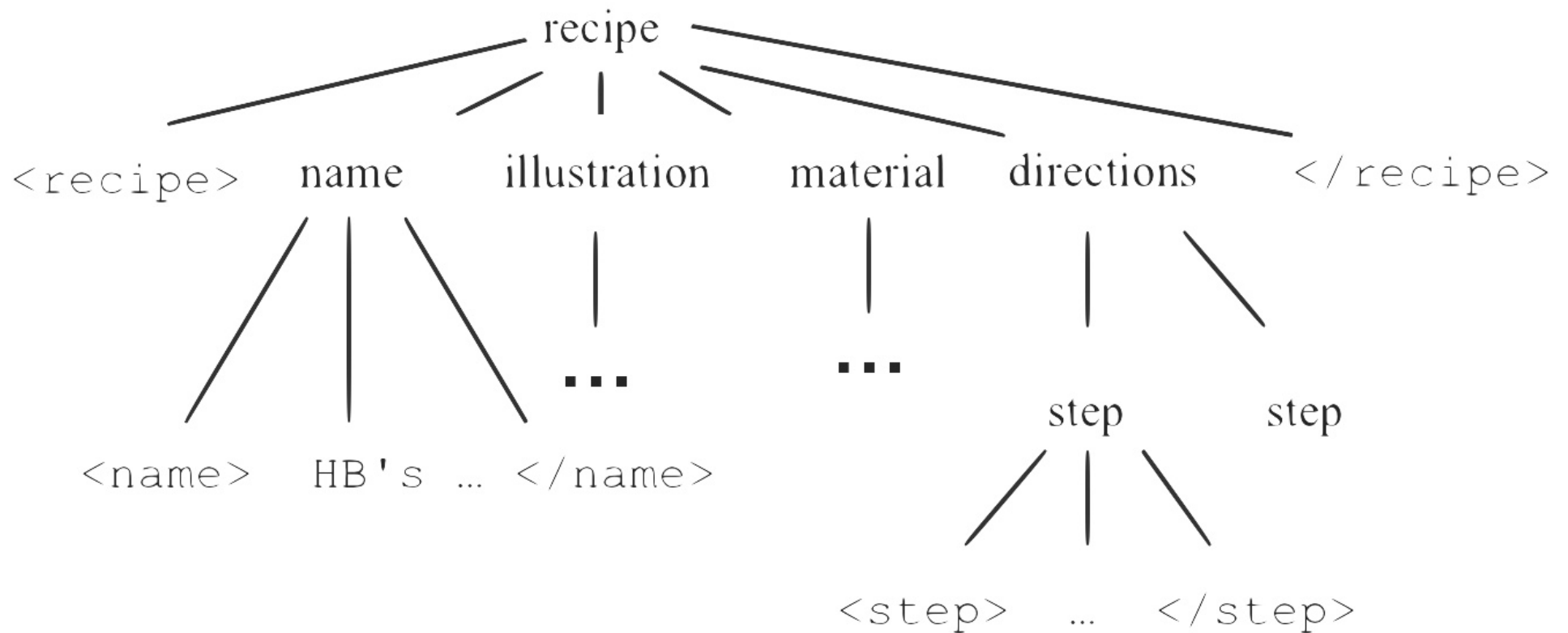
WAS IST HIER FALSCH?

```
<head>
<meta charset="utf-8">
<title>Welcome to Python.org</head></title>
<body class="python" class="home" id="homepage">
<div class=small started>
<h2 class="widget title">Get Started</h2>
<p>Whether you are new ... Python.
<p><a href="/about/gettingstarted/">Start ..</p>
</div>
</body>
</html>
```

WOHLGEFORMT

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Welcome to Python.org</title>
</head>
<body class="python home" id="homepage">
<div class="small started">
<h2 class="widget-title">Get Started</h2>
<p>Whether you are new ... Python.</p>
<p><a href="/about/gettingstarted/">Start ..</a></p>
</div>
</body>
</html>
```

Ein korrektes HTML Dokument kann als Syntaxbaum gezeichnet werden.





Ein Parser hilft Inhalte aus einem html Dokument gezieht
zu extrahieren.



DIGITAL NEWS PARSER

[Tarife & Mediadaten](#) [E-Paper](#) [Friday](#) [Tilllate](#) de fr it Zürich 16° Inhalt A-Z [Login](#)

 [Schweiz](#) [Ausland](#) [Wirtschaft](#) [Sport](#) [People](#) [Entertainment](#) **[Digital](#)** [Wissen](#) [Lifestyle](#) [Community](#) [Mehr](#) [Mediathek](#)
[News](#) [Games](#) [Dossiers](#)




Nach System-Update

Auch Windows 7 und 8 sammeln jetzt Daten

Wer meinte, die Vorgänger von Windows 10 würden keine Daten sammeln, hat falsch gedacht. Auch Windows 7 und 8 haben die Features bekommen.

48 Kommentare

- » «Windows 10 könnte in der Schweiz verboten werden»
- » «Windows 10 wird zu Recht kritisiert»



Am 9. September

Die iPhone-Präsentation wird ein dickes Ding

Am 9. September enthüllt Apple das nächste iPhone. Es dürfte eine der grössten Keynotes überhaupt werden: Der gemietete Saal bietet 7000 Personen Platz.

[Bilder](#) [Infografik](#) 124 Kommentare

- » Swatch ärgert Apple mit neuem Slogan
- » Tim Cook beschert Apple 60 Milliarden – per E-Mail

Parser Digital Headline.py

```
class DigitalHeadLine(HTMLParser):
    capture_txt = False
    headlines = []
    def handle_starttag(self, tag, attrs):
        if tag == "h2":
            if "data-vr-contentbox" in dict(attrs):
                self.capture_txt = True

    def handle_endtag(self, tag):
        if tag == "h2":
            if self.capture_txt == True:
                self.capture_txt = False

    def handle_data(self, data):
        if self.capture_txt == True:
            self.headlines.append(data)
```

PROJEKT:

STORY FINGERPRINT

- Joël Simonet & Alexander Gröflin (Universität Basel) untersuchen das Potenzial von Rückmeldungen auf Online-Artikel (z.B. 20Min).

[Simonet2015] Joël Simonet, Leserkommentare des Newsportals 20 Minuten - Analyse von Webinhalten mithilfe des Condition Action Tools WebAPI ECA-Engine, Abschlussarbeit Informatik, Universität Basel

EIN ARTIKEL 554 WORTMELDUNGEN



[Schweiz](#) [Ausland](#) [Wirtschaft](#) [Sport](#) [People](#) [Entertainment](#) [Digital](#) [Wissen](#) [Lifestyle](#) [Community](#) [Mehr](#) [Mediathek](#)

[Zürich](#) [Bern](#) [Basel](#) [Zentralschweiz](#) [Ostschweiz](#) [Dossiers](#)

Ihre Story, Ihre Informationen, Ihr Hinweis? feedback@20minuten.ch

«Besorgniserregend» 31. August 2015 08:05; Akt: 31.08.2015 12:56

Schweizer Jugendliche leiden unter Stress

Die vierte Juvenir-Studie der Jacobs Foundation zeigt, wie Schweizer Jugendliche mit Stress und Leistungsdruck umgehen. Die Resultate seien alarmierend.

554 Kommentare [i](#) [Login](#)

 [Eigenen Beitrag verfassen](#)

Die beliebtesten Leser-Kommentare



1436
80

 **lauretta** am 31.08.2015 08:20

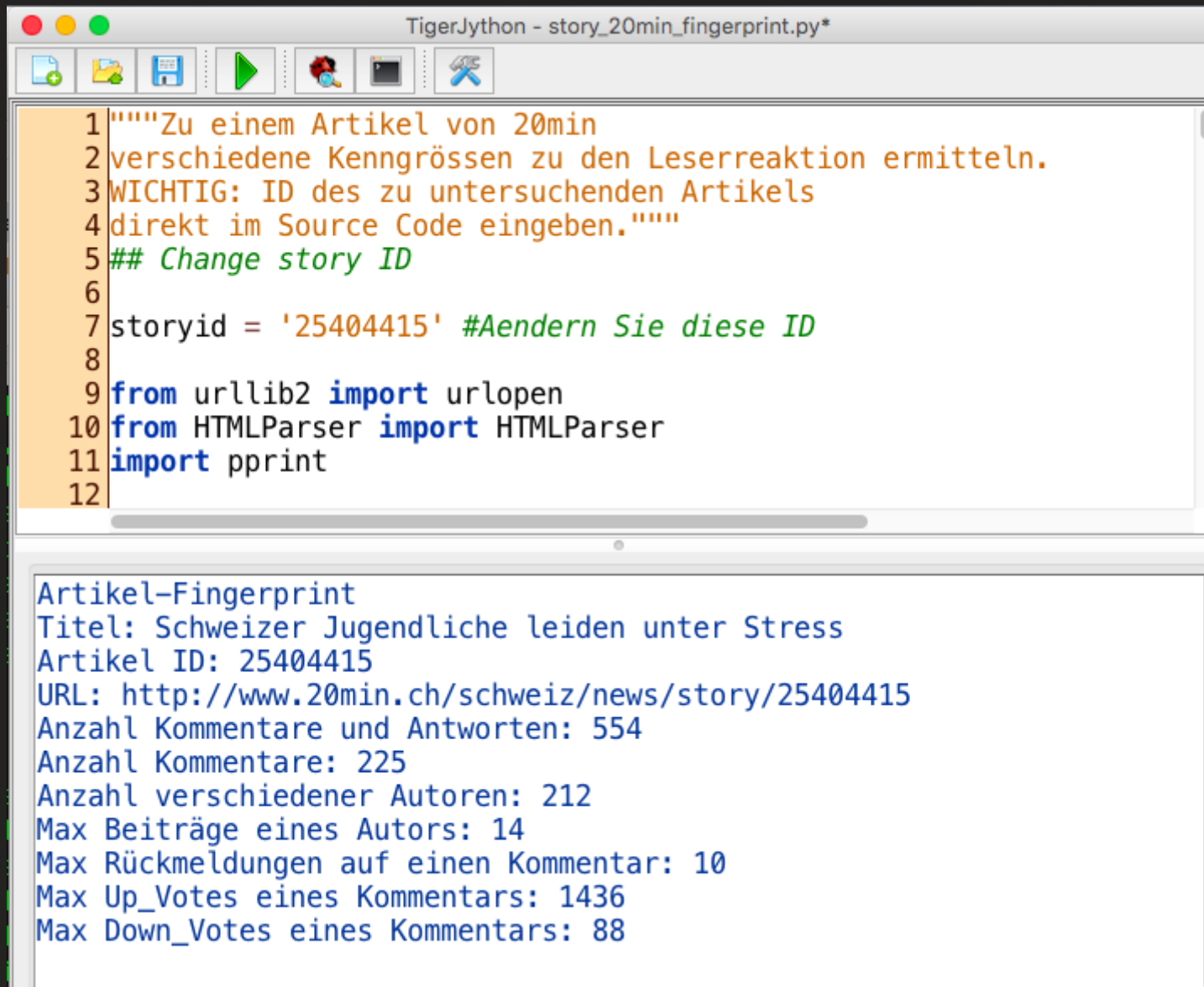
wahrheit
nicht nur kinder...die meisten schweizer leiden unter stress. mit oder ohne arbeit / schule. viele erkranken ernsthaft daran, diese bekommen dann den stempel, nicht belastbar, nicht anpassungsfähig. das mag wohl daran liegen, dass die ganze swissraff-gesellschaft nur auf wachstum fokussiert und einfach

[▲ Diesen Beitrag melden](#)

AUTOMATISIERTE DATENERHEBUNG

- Ist Zustand: Quantitative Auswertung der Rückmeldungen
 - Totale Anzahl
 - Anzahl Erstkommentare
 - Anzahl unterschiedlicher Autoren
 - Up Votes
 - Down Votes

check external links.py



The image shows a screenshot of a TigerJython IDE window. The title bar reads "TigerJython - story_20min_fingerprint.py*". The window contains a Python script in the editor and its output in the console.

```
1 """Zu einem Artikel von 20min
2 verschiedene Kenngrößen zu den Leserreaktion ermitteln.
3 WICHTIG: ID des zu untersuchenden Artikels
4 direkt im Source Code eingeben."""
5 ## Change story ID
6
7 storyid = '25404415' #Aendern Sie diese ID
8
9 from urllib2 import urlopen
10 from HTMLParser import HTMLParser
11 import pprint
12
```

Artikel-Fingerprint
Titel: Schweizer Jugendliche leiden unter Stress
Artikel ID: 25404415
URL: <http://www.20min.ch/schweiz/news/story/25404415>
Anzahl Kommentare und Antworten: 554
Anzahl Kommentare: 225
Anzahl verschiedener Autoren: 212
Max Beiträge eines Autors: 14
Max Rückmeldungen auf einen Kommentar: 10
Max Up_Votes eines Kommentars: 1436
Max Down_Votes eines Kommentars: 88

VIELEN DANK

Alle Unterlagen finden Sie auf:

<http://www.tigerjython.ch/kurs2015/>

oder

<https://github.com/mgje/PIUMP>