Python als JavaScript im Browser mit Transcrypt

Carsten Grohmann

3. Februar 2021

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Agenda

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Agenda

1. Warum?
2. Möglichkeiten?
3. Was ist Transcrypt?
4. Unterschiede zu CPython
5. Pragmas
6. Erfahrungen
7. Vor- und Nachteile
8. Mein Projekt – OOMAnalyser
9. Weiterführende Informationen
10. Danke
11. Lizenz

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Warum?

Python als JavaScript im Browser mit Transcrypt

Carsten Grohmann

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Warum?

- Neugierde
- “Läuft Python überhaupt im Browser und wenn ja, wie?”

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Möglichkeiten?

Welche Möglichkeiten gibt es dafür?

1. Python zu JavaScript compilieren

- Transcrypt[1]
- pyjs[2]
- PScript[3]

2. Python direkt im Browser laufen lassen

- Brython[4]
- PyPy.js[5]

Es gibt sicher mehr Möglichkeiten. Ich habe mir nur Transcrypt näher angesehen.

---

[1] http://transcrypt.org/
[2] http://pyjs.org/
[3] https://github.com/flexxui/pscript
[4] https://brython.info/
[5] https://pypyjs.org/

Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Was ist Transcrypt?

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Was ist Transcrypt?

- Ein Transpiler, der Python-Quellcode in JavaScript übersetzt
- Ziel
  - gut lesbarer Code
  - schneller und kompakter Code ("schlank und rank" / "lean and mean")
  - nahe an einer 1:1 Umsetzung
- Direkte Unterstützung für 90 % CPython (siehe "The main differences with CPython"[6])
- Unterstützung für weitere 9 % über Pragmas und Kommandozeilenparameter
  - z. B. Überladen von Operatoren
  - Mini-String-Format-Sprache
- 1 % wird nicht unterstützt

[6] http://www.transcrypt.org/docs/html/differences_cpython.html

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Beispiel: 1:1 Umsetzung

Python

```
def print_hello():
    print("Hello!")
print_hello()
```

JavaScript

```
export var print_hello = function () {
        print ('Hello!');
};
print_hello ();
```

Ausgabe

```
Hello!
```


Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Unterschiede zu CPython

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Was geht nicht

- `exec()`
- `eval()`
- Bedingter Import von Modulen
- Schreibender Zugriff aufs Dateisystem
- Nutzung von C/C++ Modulen
- Formatierung mit %, besser `.format()` oder f-Strings
- Überladen von Operatoren ist eingeschränkt
- ...

Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Was geht dafür mehr

- Ereignisgesteuert
- Integration von JavaScript-Bibliotheken, wie jQuery
- JavaScript `eval()`

Pragmas

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Pragmas

- Steuern die lokale/globale Umwandlung des Quellcodes
- Nicht “scope aware”
- Manche lassen sich global aktivieren und lokal wieder deaktivieren

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

4 Formen

```
# Form 1 - Funktionsaufruf
from org.transcrypt.stubs.browser import __pragma__
__pragma__ (<parameters>)

# Form 2 - Kommentar für Blöcke
# __pragma__ (<parameters>)

# Form 3 - Zeilenweise
<line of code> # __:<single parameter>
# Identisch mit:
__pragma__ (<single parameter>); <line of code>; __pragma__ (no<single parameter>)

# Form 4 - Kommandozeilenschalter
$ transcrypt --<parameter>
```

Python als JavaScript im Browser mit Transcrypt

Carsten Grohmann

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Bedingte Kompilierung im C-Stil

- Auswertung erfolgt zur Übersetzungszeit
- Es gibt
  - `__pragma__ ('ifdef', <symbol>)`
  - `__pragma__ ('ifndef', <symbol>)`
  - `__pragma__ ('else')`
  - `__pragma__ ('endif')`

Beispiel

```
#__pragma__ ('ifdef', '_NSAKEY')
class Encoder:
    def hash(self, value):
        return value ^ 5
#__pragma__ ('else')
import random
class Encoder:
    def hash(self, value):
        return value ^ random.randbytes(1)
#__pragma__ ('endif')
```

Ausnahme --sform

- Aktiviert die erweiterte Formatierung in Zeichenketten z. B. `"{0:x}".format(42)`
- Nur global über die Kommandozeile

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Pragma alias/noalias

- Ermöglich die Definition eines Alias unter Python
- Löschen mit `__pragma__ ('noalias', <Python id part>)`
- Liste vordefinierter Aliases: http://transcrypt.org/docs/html/special_facilities.html#pragma-alias

Beispiel

Python

```
# js_undefined ist ein Alias für "undefined" in JavaScript
# Identitätsvergleich oder Gleichheitsvergleich
if value is None or value == js_undefined:
    i=1
```

JavaScript

```
if (value === null || value == undefined) {
        var i = 1;
}
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Pragma iconv/noiconv (Kommandozeile: -i)

- Erzeugt einen Wrapper-Aufruf um ein nicht-iterierbares Objekt zu iterieren
- Mit `noiconv` (default):
  - Verhalten hängt davon ab, ob das Objekt bereits iterierbar ist oder nicht
- Mit `iconv`:
  - Objekt wird durch einen Wrapper iterierbar
- Siehe auch “Pragma jsiter/nojsiter”

Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Beispiel

Python

```
d1 = {'a': 1, 'b': 2}
print('Mit iconv:')
# __pragma__('iconv')
for i in d1:
    print(d1[i])
# __pragma__('noiconv')
print('Ohne iconv:')
# verursacht: Uncaught TypeError: Result of the Symbol.iterator method is not an object
for i in d1:
    print(d1[i])
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

JavaScript

```
export var d1 = dict ({'a': 1, 'b': 2});
print ('Mit iconv:');
for (var i of __i__ (d1)) {
        print (d1 [i]);
}
print ('Ohne iconv:');
for (var i of d1) {
        print (d1 [i]);
}
```

Ausgabe

```
Mit iconv:
1
2
Ohne iconv:
/home/carsten/Unix-Stammtisch202102/beispiel_iconv-bundled.js:1566
    for (var i of d1) {
                  ^

TypeError: Result of the Symbol.iterator method is not an object
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Pragma jsiter/nojsiter

- Steuert die Umsetzung von Dictionaries nach JavaScript
- Mit `nojsiter` (default):
  - Dicts werden zu `dict({...})` übersetzt
  - mit Python-Zusatzfunktionen
  - nicht iterierbar (Doku irrt hier)
- Mit `jsiter`:
  - Dicts werden zu reinen JavaScript-Dictionaries `{}`
  - keine Python-artigen Zusatzfunktionen
  - iterierbar
- Siehe auch “Pragma iconv/noiconv (Kommandozeile: -i)”
- `jsiter` erzeugt ein iterierbares Objekt, während `iconv` den Aufruf durch eine Wrapper-Funktion kapselt

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Beispiel

Python

```
print('Mit jsiter:')
# __pragma__('jsiter')
d1 = {'A': 1, 'B': 2}
for i in d1:
    print(d1[i])
# __pragma__('nojsiter')
print('Ohne jsiter:')
# verursacht: Uncaught TypeError: Result of the Symbol.iterator method is not an object
d2 = {'a': 1, 'b': 2}
for i in d2:
    print(d2[i])
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

JavaScript

```
print ('Mit jsiter:');
export var d1 = {'A': 1, 'B': 2};
for (var i in d1) {
        print (d1 [i]);
}
print ('Ohne jsiter:');
export var d2 = dict ({'a': 1, 'b': 2});
for (var i of d2) {
        print (d2 [i]);
}
```

Ausgabe

```
Mit jsiter:
1
2
Ohne jsiter:

/home/carsten/Unix-Stammtisch2021/beispiel_jsiter-bundled.js:1564
    for (var i of d1) {
                  ^
TypeError: Result of the Symbol.iterator method is not an object
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Pragma js

- Fügt literalen JavaScript-Code ein

Beispiel

Python

```
conn = None
errObj = None
__pragma__('js', '{}',  '''
   try {
         conn = new(XMLHttpRequest || ActiveXObject)('MSXML2.XMLHTTP.3.0');
   } catch( err ) {
         errObj = err;
   }
            ''')
```

JavaScript

```
export var conn = null;
export var errObj = null;

   try {
         conn = new(XMLHttpRequest || ActiveXObject)('MSXML2.XMLHTTP.3.0');
   } catch( err ) {
         errObj = err;
   }
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Pragma keycheck/nokeycheck (Kommandozeile: -kc)

- Steuert das Verhalten Dictionaries bei der Abfrage nicht existierender Schlüssel
- Mit `nokeycheck` (default):
  - gibt `null` oder `undefined` (Alias `js_undefined`)
- Mit `keycheck`:
  - Zugriffe auf nicht existierende Dictionary-Schlüssel lösen eine `KeyError` Exception aus

Beispiel

Python

```
d = {1: 1, 42: 32}
# __pragma__('keycheck')
print('Mit keycheck:')
try:
    print('Mit keycheck: ', d['non-existing'])
except KeyError:
    print('KeyError Exception aufgetreten')
# __pragma__('nokeycheck')
print('Ohne keycheck:')
print('42 in original:', 42 in d)
print('43 in original:', 43 in d)
print('original[42] =', d[42])
print('original[43] =', d[43])
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

JavaScript

```
export var d = dict ({1: 1, 42: 32});
print ('Mit keycheck:');
try {
        print ('Mit keycheck: ', __k__ (d, 'non-existing'));
}
catch (__except0__) {
        if (isinstance (__except0__, KeyError)) {
                print ('KeyError Exception aufgetreten');
        }
        else {
                throw __except0__;
        }
}
print ('Ohne keycheck:');
print ('42 in original:', __in__ (42, d));
print ('43 in original:', __in__ (43, d));
print ('original[42] =', d [42]);
print ('original[43] =', d [43]);
```

Ausgabe

```
Mit keycheck:
KeyError Exception aufgetreten
Ohne keycheck:
42 in d: True
43 in d: False
d[42] = 32
d[43] = None
```

Pragma kwargs

- Schleife ohne Memory Decorator 38ms
- Mit Memory Decorator und kwargs 1,08s
- langsamer als ohne Optimierung

Beispiel

```
# vergessen :-/
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Pragma noanno

- Mit der Option -a erhalten die JavaScript-Dateien Kommentare mit Ursprungsdatei und -zeilennummer
- das Pragma `noanno` deaktiviert dieses Verhalten

Pragma opov/noopov (Kommandozeile: -o)

- Aktiviert das Überladen der Operatoren: `*`, `/`, `+`, `-`, `@`, `[]`, `()`, `==`, `!=`, `<`, `<=`, `>`, und `>=`

Beispiel

Python

```
l = ['A', 'B', 'C', 'D', 'E', ]
l1 = l + ['Notes']
# __pragma__('opov')
l2 = l + ['Notes']
# __pragma('noopov')
print(l1)
print(l2)
```

JavaScript

```
export var l = ['A', 'B', 'C', 'D', 'E'];
export var l1 = l + ['Notes'];
export var l2 = __add__ (l, ['Notes']);
__call__ (print, null, l1);
__call__ (print, null, l2);
```

Ausgabe

```
A,B,C,D,ENotes
['A', 'B', 'C', 'D', 'E', 'Notes']
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Pragma skip/noskip

Code der nur vom Python-Interpreter ausgeführt, aber nicht nach JavaScript übersetzt wird

Beispiel

```
# __pragma__ ('skip')
document = window = Math = Date = 0 # Prevent warning from static code checkers
# __pragma__ ('noskip')
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Pragma tconv/notconv (Kommandozeile: -t)

- Steuert den Wahrheitswert von leeren Listen, Sets und Dictionaries
- Mit `notconv` (default):
  - Leere Listen, Dictionaries und Set werden wie in JavaScript als wahr ausgewertet
- Mit `tconv`:
  - Leere Listen, Dictionaries und Set werden wie in Python als falsch ausgewertet
- Fallstrick:
  - `elements = [document.getElementById('non-existing')]` ergibt `elements = [null]` und damit wahr
  - ist nicht JavaScript-spezifisch, sondern auch in Python so

Beispiel

Python

```
print('Ohne tconv:')
empty = []
if empty:
    print("Leere Liste: wahr")
else:
    print("Leere Liste: nicht wahr")
print('Mit tconv')
# __pragma__ ('tconv')
if empty:
    print("Leere Liste: wahr")
else:
    print("Leere Liste: nicht wahr")
# __pragma__ ('notconv')
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

JavaScript

```
print ('Ohne tconv:');
export var empty = [];
if (empty) {
        print ('Leere Liste: wahr');
}
else {
        print ('Leere Liste: nicht wahr');
}
print ('Mit tconv');
if (__t__ (empty)) {
        print ('Leere Liste: wahr');
}
else {
        print ('Leere Liste: nicht wahr');
```

Ausgabe

```
Ohne tconv:
Leere Liste: wahr
Mit tconv
Leere Liste: nicht wahr
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Weitere Pragmas

- docat (Kommandozeile: -d)
  - Übernimmt Docstrings mit in den JavaScript-Code
- ecom (Kommandozeile: -ec)
  - Ausführbare Kommentare
  - Kommentare mit "?" als erstes und letztes Zeichen, werden nur durch Transcrypt ausgeführt
- fcall (Kommandozeile: -f)
  - Fast Calls

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

- gen
  - Aktiviert Generatoren und Iteratoren
  - Genaues weiß man nicht – keine Doku
- gsend
  - Aktiviert den Syntax für *send generators*
- jsmod (Kommandozeile: -jm)
  - Aktiviert den JavaScript-Syntax für % statt des Python-Syntaxes
- xpath (Kommandozeile: -xp / -xpath)
  - Erweitert den Modulsuchpfad
- xtrans
  - Code mit einem externen Programm verarbeiten / übersetzen

Erfahrungen

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Numerisch sortieren

JavaScript `sort()` sortiert die Elemente einer Liste alphabetisch, auch wenn es Integer sind.

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Beispiel

Python

```
l1 = [20, 1, 5, 2, 4, 7, 12]
l2 = [20, 1, 5, 2, 4, 7, 12]
print('Integers unsortiert:                  ', l1)
l1.sort()
print('Integers sortiert mit sort():         ', l1)
l1.sort(key=int)
print('Integers sortiert mit sort(key=int): ', l1)
# funktioniert auch mit reinem JS
# __pragma__ ('js', 'l2.sort((a, b) => a - b)')
print('Integers sortiert mit Javascript:     ', l2)
```

JavaScript

```
export var l1 = [20, 1, 5, 2, 4, 7, 12];
export var l2 = [20, 1, 5, 2, 4, 7, 12];
print ('Integers unsortiert:                  ', l1);
l1.py_sort ();
print ('Integers sortiert mit sort():         ', l1);
l1.py_sort (__kwargtrans__ ({key: int}));
print ('Integers sortiert mit sort(key=int): ', l1);
l2.sort((a, b) => a - b)
print ('Integers sortiert mit Javascript:     ', l2);
```

Ausgabe

```
Integers unsortiert:                  [20, 1, 5, 2, 4, 7, 12]
Integers sortiert mit sort():         [1, 12, 2, 20, 4, 5, 7]
Integers sortiert mit sort(key=int):  [1, 2, 4, 5, 7, 12, 20]
Integers sortiert mit Javascript:     [1, 2, 4, 5, 7, 12, 20]
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Reguläre Ausdrücke

- Reguläre Ausdrucke werden ins JavaScript-Format übersetzt
  - Doku: `match.group()`
    - JavaScript Match Object verwendet `undefined` für nicht verwendete Capture-Gruppen, d. h. ein Test auf `None` liefert nicht das gewünschte Resultat
  - Ist: `match.group('existiert-nicht')`
    - wirft `IndexError` Exception
  - Limit auf 1000 Token beim Umsetzen, d. h. komplexe RE aufteilen

Python

```
import re
PID = re.compile(r'^PID: (?P<pid>\d+)')
match = PID.search('PID: 42')
try:
    match.group('existiert-nicht')
except IndexError:
    print('Gruppe "existiert-nicht" nicht gefunden')
if match:
    print("PID gefunden match.group(1):     ", match.group(1))
    print("PID gefunden match.group('pid'): ", match.group('pid'))
else:
    print('PID nicht gefunden')
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

JavaScript

```
export var PID = re.compile ('^PID: (?P<pid>\\d+)');
export var match = PID.search ('PID: 42');
try {
        match.group ('existiert-nicht');
}
catch (__except0__) {
        if (isinstance (__except0__, IndexError)) {
                print ('Gruppe "existiert-nicht" nicht gefunden');
        }
        else {
                throw __except0__;
        }
}
if (match) {
        print ('PID gefunden match.group(1):     ', match.group (1));
        print ("PID gefunden match.group('pid'): ", match.group ('pid'));
}
else {
        print ('PID nicht gefunden');
}
```

Ausgabe

```
Gruppe "existiert-nicht" nicht gefunden
PID gefunden match.group(1):      42
PID gefunden match.group('pid'):  42
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Negativer Index

- `list[-1]` gibt `None` zurück
- Lösung `__pragma__ ('opov')`

Beispiel

Python

```
l = [1,2,3,4]
print('Letztes Element ohne opov: ', l[-1])
# __pragma__('opov')
print('Letztes Element mit opov:  ', l[-1])
# __pragma__('noopov')
```

JavaScript

```
export var l = [1, 2, 3, 4];
print ('Letztes Element ohne opov: ', l [-(1)]);
__call__ (print, null, 'Letztes Element mit opov:  ', __getitem__ (l, __neg__ (1)));
```

Ausgabe

```
Letztes Element ohne opov:  None
Letztes Element mit opov:   4
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Exceptions

- Sollten wie in Python funktionieren, tun sie aber nicht

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Beispiel

Python

```
try:
    raise NotImplementedError
except NotImplementedError:
    print('NotImplementedError aufgetreten')
except (object) as e:
    print('Exception aufgetreten: ', str(e))
```


Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

JavaScript

```
try {
        var __except0__ = NotImplementedError;
        __except0__.__cause__ = null;
        throw __except0__;
}
catch (__except0__) {
        if (isinstance (__except0__, NotImplementedError)) {
                print ('NotImplementedError aufgetreten');
        }
        else if (isinstance (__except0__, object)) {
                var e = __except0__;
                print ('Exception aufgetreten: ', str (e));
        }
        else {
                throw __except0__;
        }
}
```

Ausgabe (nicht ganz die erwartete)

```
Exception aufgetreten:  function () {
                var args = [] .slice.apply (arguments);
                return cls.__new__ (args);
            }
```


Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Mix von Python und JavaScript

Man kann beliebige JavaScript-Ausdrücke im Python-Quellcode nutzen, um so z. B. auf dem DOM-Tree zuzugreifen.

Python

```
element = document.getElementById('version')
```

JavaScript

```
var element = document.getElementById ('version');
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Slices

Beispiel

Python

```
all = range(16)
print('Alle Elemente:            ', all)
print('Elemente 8-11:            ', all[8:12])
print('Letzten beiden Elemente : ', all[-2:])
print('Jedes zweite Element zwischen 8-11: ', all[8:12:2])
```

JavaScript

```
export var all = range (16);
print ('Alle Elemente:            ', all);
print ('Elemente 8-11:            ', all.__getslice__ (8, 12, 1));
print ('Letzten beiden Elemente : ', all.__getslice__ (-(2), null, 1));
print ('Jedes zweite Element zwischen 8-11: ', all.__getslice__ (8, 12, 2));
```

Ausgabe

```
Alle Elemente:            [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
Elemente 8-11:            [8, 9, 10, 11]
Letzten beiden Elemente :  [14, 15]
Jedes zweite Element zwischen 8-11:  [8, 10]
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Comprehension

Es sind alle drei Arten von Comprehensions unterstützt:

- List Comprehensions
- Dict Comprehensions
- Set Comprehensions

Python

```
original = {i : chr(65 + i) for i in range(4)}
# __pragma__ ('iconv')
inverted = {original [key]: key for key in original}
# __pragma__ ('noiconv')
print('Original dict: ', original)
print('Invertiertes dict: ', inverted)
even = {2 * i for i in [0, 9, 1, 7, 2, 8, 3, 6, 4, 5]}
odd = {2 * i + 1 for i in [5, 6, 7, 8, 9, 4, 3, 1, 2, 0]}
print('Set mit geraden Zahlen:   ', even)
print('Set mit ungeraden Zahlen: ', odd)
squares = [i * i for i in range (10) if i % 2]
print('Quadrate: ', squares)
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

JavaScript

```
export var original = (function () {
        var __accu0__ = [];
        for (var i = 0; i < 4; i++) {
                __accu0__.append ([i, chr (65 + i)]);
        }
        return dict (__accu0__);
}) ();
export var inverted = (function () {
        var __accu0__ = [];
        for (var key of __i__ (original)) {
                __accu0__.append ([original [key], key]);
        }
        return dict (__accu0__);
}) ();
print ('Original dict: ', original);
print ('Invertiertes dict: ', inverted);
export var even = (function () {
        var __accu0__ = [];
        for (var i of [0, 9, 1, 7, 2, 8, 3, 6, 4, 5]) {
                __accu0__.append (2 * i);
        }
        return set (__accu0__);
}) ();
export var odd = (function () {
        var __accu0__ = [];
        for (var i of [5, 6, 7, 8, 9, 4, 3, 1, 2, 0]) {
                __accu0__.append (2 * i + 1);
        }
        return set (__accu0__);
}) ();
print ('Set mit geraden Zahlen:   ', even);
print ('Set mit ungeraden Zahlen: ', odd);
```

```
...
```


Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Ausgabe

```
Original dict:  {0: 'A', 1: 'B', 2: 'C', 3: 'D'}
Invertiertes dict:  {'A': '0', 'B': '1', 'C': '2', 'D': '3'}
Set mit geraden Zahlen:    {0, 18, 2, 14, 4, 16, 6, 12, 8, 10}
Set mit ungeraden Zahlen:  {11, 13, 15, 17, 19, 9, 7, 3, 5, 1}
Quadrate:  [1, 9, 25, 49, 81]
```

Python als JavaScript im Browser mit Transcrypt

Carsten Grohmann

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Vergleiche in JavaScript

- ==: Gleichheitsvergleich mit impliziter Typenumwandlung
- ===: Identitätsvergleich

```
typeof null           // "object" (not "null" for legacy reasons)
typeof undefined      // "undefined"
null === undefined    // false
null == undefined     // true
null === null         // true
null == null          // true
!null                 // true
isNaN(1 + null)       // false
isNaN(1 + undefined)  // true
```

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

None, Null, undefined

- JavaScript hat zwei Werte für `None`:
  - `null`
    - repräsentiert das absichtliche Fehlen eines Wertes
    - `x is None` wird zu `x === null`,
    - `undefined === null` ist `false`
  - `undefined`
    - Variable wurde zwar deklariert, aber ihr wurde kein Wert zugewiesen
- Hat gelegentlich Schmerzen verursacht
- Siehe auch “Pragma alias/noalias”

Node.js

- die Ausgaben für diese Präsentation wurden mit Node.js erzeugt
- allerdings funktioniert das Nachladen von Modulen nicht
- Lösung: JavaScript Modul-Packer wie rollup.js[7]

Beispiel

```
$ rollup --format=umd --name beispiel_slices \
  --file=beispiel_slices-bundled.js \
  -- __target__/beispiel_slices.js
$ node beispiel_slices-bundled.js
Ohne Pragma opov:  None
Mit Pragma opov:  4
```

[7] https://rollupjs.org/

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Vorteile & Nachteile

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Vorteile

- Wiedererkennbarkeit des Codes
- Ich habe viel mit Transcrypt 3.6 gearbeitet
  - erzeugt eine einzelne JavaScript-Datei und eine Sourcemap für den Debugger
  - Transcrypt 3.7 nutzt ein Modul-Konzept
- Nutzer-Community vorhanden
- Eigenes Test-Framework vorhanden

Python als JavaScript im Browser mit Transcrypt
Carsten Grohmann
Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Nachteile

- Ab Modul-Konzept ab 3.7
  - d.h. zusätzlicher Schritt mit einem JavaScript Bundler
  - Sourcemap wird dabei unbrauchbar
- Einmannshow, aktuell keine Entwicklung
- Eigenes Test-Framework arbeitet nur im Browser
- Integration bestehenden Python-Codes kann sich schwierig gestalten

Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Zusammenfassung

- Mit Transcrypt erzeugter Code ist näher an JavaScript als an Python
- JavaScript-Kenntnisse erforderlich
- Wahrscheinlich hätte ich eine bessere Umsetzung gewählt
- aber auch nichts anderes getestet

Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Mein Projekt – OOMAnalyser

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Mein Projekt – OOMAnalyser

- Start November 2017
- Webseite, die lokal einen Linux OOM analysiert und anzeigt
- Umfang
  - ca. 930 Zeilen HTML
  - 1400 Zeilen Python
  - 3750 Zeilen JavaScript (mit Runtime)

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

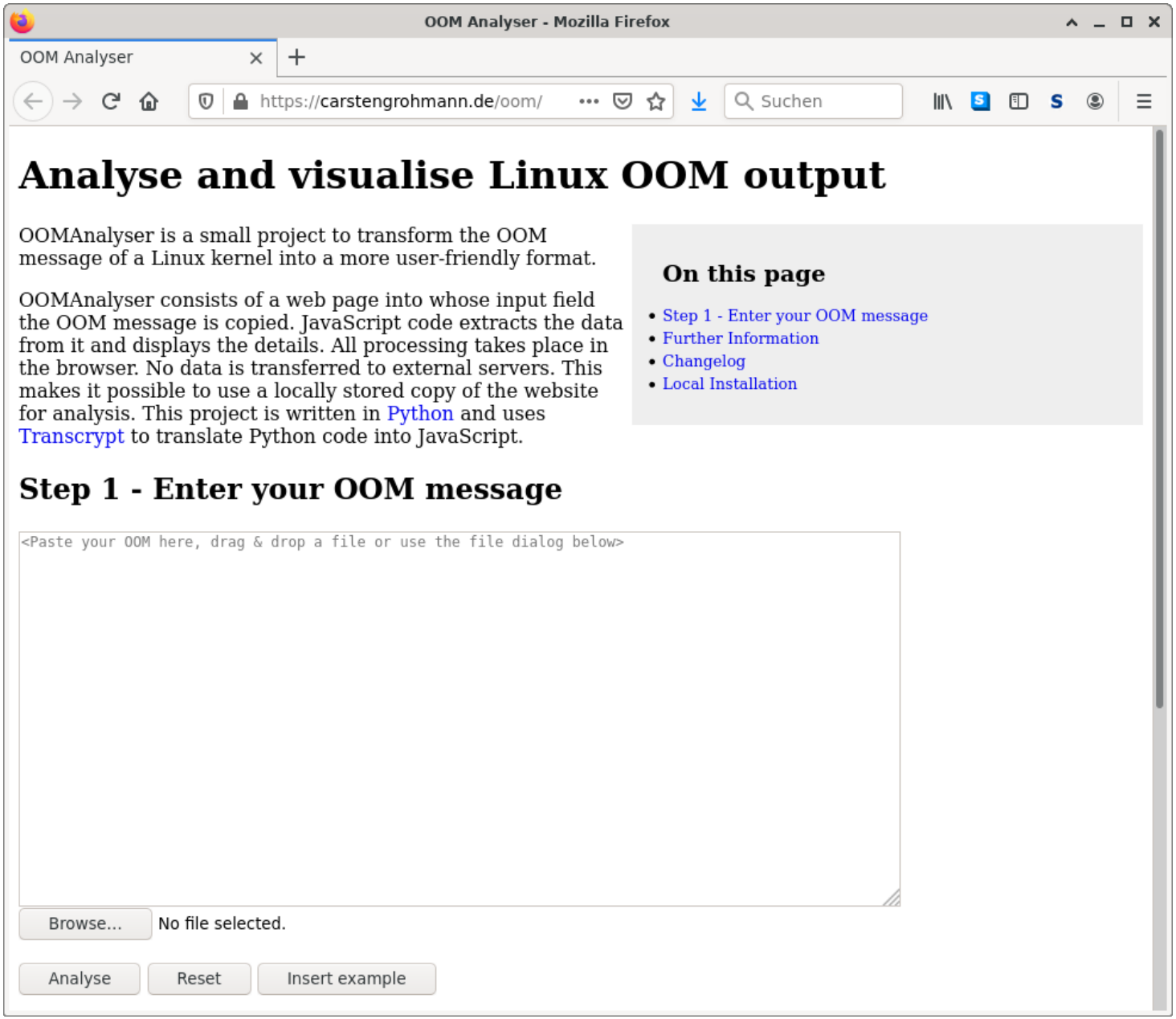
Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz



https://www.carstengrohmann.de/oom/

Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz

Weiterführende Informationen

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Weiterführende Informationen

- Homepage Transcrypt[8]
- Transcrypt Dokumentation[9]
- OOM Analyser[10]

[8] https://www.transcrypt.org
[9] http://transcrypt.org/docs/html/index.html
[10] https://www.carstengrohmann.de/oom/

Danke

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Danke

- Fragen, Anregungen, Meinungen
- Vielen Dank für die Aufmerksamkeit!

Agenda

Warum?

Möglichkeiten?

Was ist Transcrypt?

Unterschiede zu CPython

Pragmas

Erfahrungen

Vorteile & Nachteile

Mein Projekt – OOMAnalyser

Weiterführende Informationen

Danke

Lizenz

Lizenz

Lizenz

[11] http://creativecommons.org/licenses/by-nc-sa/4.0/

Agenda
Warum?
Möglichkeiten?
Was ist Transcrypt?
Unterschiede zu CPython
Pragmas
Erfahrungen
Vorteile & Nachteile
Mein Projekt – OOMAnalyser
Weiterführende Informationen
Danke
Lizenz