Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie
AGH University of Science
and Technology

PythonTEX

Damian Łączak

Edycja i prezentacja tekstów naukowych
Python: 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]

Część I

Basics of PythonTEX

Outline

Outline

Outline

First example

No brackets only indentation

```python
# And this is a comment.
from random import randint
number = randint(0, 9)
if number < 5:
    print "0-4"
else:
    print "5-9"
```

Example output

```
0-4
```

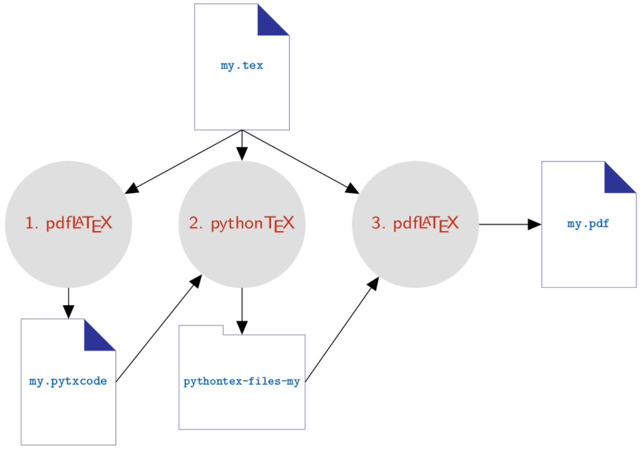
Functions

```python
def sayMyName(name):
    print ("Your name is {0}".format(name))
sayMyName("Damian")
```

Output

```
Your name is Damian
```

PDF creation process



Sessions

What are they for?

- Parallel execution

Sessions

What are they for?

- Parallel execution
  - Increase speed

Sessions

What are they for?

- Parallel execution
  - Increase speed
  - Different settings

Sessions

What are they for?

- Parallel execution
  - Increase speed
  - Different settings
- Default session

Sessions

What are they for?

- Parallel execution
  - Increase speed
  - Different settings
- Default session
- Session name

Sessions

What are they for?

- Parallel execution
  - Increase speed
  - Different settings
- Default session
- Session name
  - a-z

Sessions

What are they for?

- Parallel execution
  - Increase speed
  - Different settings
- Default session
- Session name
  - a-z
  - A-Z

Sessions

What are they for?

- Parallel execution
  - Increase speed
  - Different settings
- Default session
- Session name
  - a-z
  - A-Z
  - 0-9

Sessions

What are they for?

- Parallel execution
  - Increase speed
  - Different settings
- Default session
- Session name
  - a-z
  - A-Z
  - 0-9
  - hyphen and underscore

Commands

Overall look on them

Inline commands

- py
- pyc
- pys
- pyv
- pyb

Multi-line commands

- pycode
- pysub
- pyverbatim
- pyblock

Console commands

- pyconsole
- pycon

py

inline command

Usage

Returns text representation of it's argument.

```
\py{"Hello world"}

```

output

Hello world

pyc

inline command

AGH

www.agh.edu.pl

Usage

Prints evaluated expressions that are inside curly braces preceded by exclamation mark.

```
\pyc{a = 2**8}
\py{a}

```

output

256

pys

inline command

www.agh.edu.pl

Usage

Evaluates and then substitute expressions that are surrounded by curly braces proceeded by exclamation mark by their string representation.

```
\pys{$1 + 1 = !{1+1}$}

```

output

$1 + 1 = 2$

pyv
inline command

Usage

It typesets but do not execute the code.

```
\pyc{a = 1}
\pyv{a = 256} \\
\py{a}

```

output

```
a = 256
1
```

pyb

inline command

Usage

It typesets and executes the code.

```
\pyc{a = 1}
\pyb{a = 256} \\
\py{a}

```

output

```
a = 256
256
```

pycode
environment

Usage

Enclose the code that is going to be executed but not typeset.

```
\begin{pycode}
def sayMyName(name):
  return "Your name is {0}".format(name)
sayMyName("Damian")
\end{pycode}
\py{sayMyName("Damian")}

```

output

Your name is Damian

pysub

environment

Usage

Similar to \pys. But this time this is an environment.

```
\begin{pysub}
1 + 5 = !{1 + 5} \\
Function output: !{sayMyName("Damian")} \\
2*32 = !{2**32}
\end{pysub}

```

output

1 + 5 = 6

Function output: Your name is Damian

2*32 = 4294967296

pyblock

environment

Usage

This environment enclose the code that is typeset and executed. Does not print any printed content even if autoprint flag is set to true.

```
\begin{pyblock}
sayMyName("Damian")
a = 125
a + a
\end{pyblock}

```

output

```
sayMyName("Damian")
a = 125
a + a
```

pyverbatim
environment

Usage

This environment enclose the code that is typeset and not executed.

```
\begin{pyverbatim}
sayMyName("Damian")
a = 125
a + a
\end{pyverbatim}

```

output

```
sayMyName("Damian")
a = 125
a + a
```

pyconsole

console environment

Usage

This environment treats its contents as series of commands passed to an active Python console. It shows input and output of commands.

```
\begin{pyconsole}
a = [1, 2, 3]
dir(a)
print(a)
\end{pyconsole}
```

output

```
>>> a = [1, 2, 3]
>>> dir(a)
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__'
>>> print(a)
[1, 2, 3]
```

pycon

console inline command

www.agh.edu.pl

Usage

This command executes code using emulated interpreter and shows the output back into the document, discarding the input.

```
\pycon{dir(a)}
```

output

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__'

now exiting Console...
```

Other commands or functions

Which are not so important to have single slide for them.

- \setpythontexoutputdir

Other commands or functions

Which are not so important to have single slide for them.

- \setpythontexoutputdir
- \setpythontexworkingdir

Other commands or functions

Which are not so important to have single slide for them.

- \setpythontexoutputdir
- \setpythontexworkingdir
- str

Other commands or functions

Which are not so important to have single slide for them.

- \setpythontexoutputdir
- \setpythontexworkingdir
- str
- add_dependencies

Other commands or functions

Which are not so important to have single slide for them.

- \setpythontexoutputdir
- \setpythontexworkingdir
- str
- add_dependencies
- before

Other commands or functions

Which are not so important to have single slide for them.

- \setpythontexoutputdir
- \setpythontexworkingdir
- str
- add_dependencies
- before
- after

Beamer

Frames

PytonTEX is compatible with Beamer. But beware, you need to use Beamer's fragile option for any frame containing typeset code.

Other languages

- Ruby

Other languages

- Ruby
- Octave

Other languages

- Ruby
- Octave
- Julia

Other languages

- Ruby
- Octave
- Julia
- Rust

Other languages

- Ruby
- Octave
- Julia
- Rust
- Bash

Bash

Available commands and environments

- bash

Bash

Available commands and environments

AGH

- bash
- bashblock

Bash

Available commands and environments

- bash
- bashblock
- bashverbatim

Bash

Available commands and environments

- bash
- bashblock
- bashverbatim
- bashsub

Część II

Python TEXamples

Outline

Outline

Outline

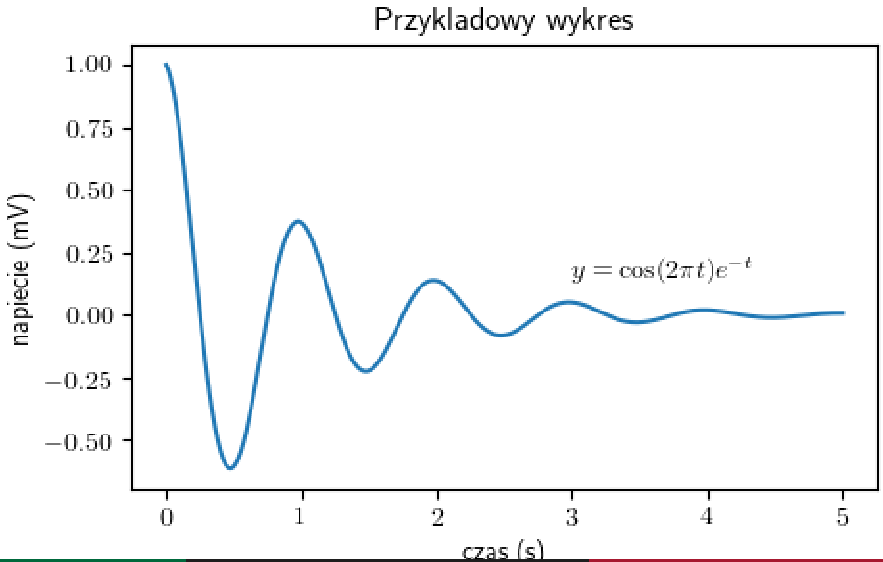
Chart

Matplotlib

```
\begin{pycode}[chart]
from pylab import *
def f(t):
    return cos(2 * pi * t) * exp(-t)
t = linspace(0, 5, 500)
y = f(t)
clf()
figure(figsize=(5, 3))
rc("text", usetex=True)
plot(t, y)
title("Przykladowy wykres")
text(3, 0.15, r"$y = \cos(2 \pi t) e^{-t}$")
xlabel("czas (s)")
ylabel("napiecie (mV)")
savefig("myplot.png", bbox_inches="tight")
print(r"\begin{center}")
print(r"\includegraphics[scale=1.0, keepaspectratio]{myplot.png}")
print(r"\end{center}")
\end{pycode}
```

Chart

File is saved to main folder by default



GPW

Using another session to improve speed of pythontex.

```
\begin{pycode}[internet]
from internet import getSymbolInfo
import time

wig20 = getSymbolInfo("WIG20")
kghm = getSymbolInfo("KGHM")
cd = getSymbolInfo("CDPROJEKT")
date = time.strftime("%Y/%m/%d");
\end{pycode}

\begin{exampleblock}{KGHM}
Current price: \py[internet]{wig20} PLN
\end{exampleblock}
```

GPW

Because it's funny to know current prices of stock.

KGHM

Current price: 2307.06

WIG20

Current price: 112 PLN

CDPROJEKT

Current price: 81.1 PLN

Actual price for date: 2017/06/12.

External files

```
\begin{pycode}[people]
from people import importPeople
people = importPeople()

print(r"\begin{tabular}{ l | r }")

print(r"{0} & {1} \\ \hline".format(people[0][0], people[0][1]))
people.pop(0)
for person in people:
  print(r"{0} & {1} \\".format(person[0], person[1]))

print(r"\end{tabular}")
\end{pycode}
```

External files

The biggest hassle of creating tables is finally diminished.

List of people

Name	Surname
John	Smith
Victoria	Volpe
James	Jansen
Janice	Bishop
Charles	Stevens
Felicia	Appling
Nora	Sinkler

Część III

Dodatek

Bibliography I

CTAN
PythonTEX Package documentation
http://piotrkosoft.net/pub/mirrors/CTAN/macros/latex/contrib/pythontex/pythontex.pdf

CTAN
Beamer user guide
http://mirrors.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf