

# Creating and Automating Exams with LaTeX & Friends

---

Uwe Ziegenhagen

August 11, 2019

Dante e. V. Heidelberg

# What's this talk about?


- As a lecturer I need to prepare exams
- Exams require significant amount of time:
  - not too simple, not too difficult
  - right amount of questions
  - unambiguous questions
- $\text{\LaTeX}$  offers various packages to typeset exams and exercises, in this talk we cover `exam`

# The exam document class

- Maintained by Philip Hirschhorn
- Current version 2.6 from November 2017
- Comes with a well-written manual
- Supports various types of questions, multiple choice questions, grading tables, etc.


# A very basic example

```
1 \documentclass[12pt]{exam}
2
3 \begin{document}\Large
4
5 \begin{questions}
6 \question[10] Who was Albert Einstein?
7 \question[10] Compute  $(e = m \cdot c^2)$ !
8 \end{questions}
9
10 \end{document}
```

Listing 1: A very basic exam example 

# Resulting document

1. (10 points) Who was Albert Einstein?
2. (10 points) Compute  $e = m \cdot c^2!$

**Figure 1:** Resulting output 

# Localizing the exam-specific terms


Exam-specific terms can be localized, here's an example for German:

```
1 \pointpoints{Punkt}{Punkte}
2 \bonuspointpoints{Bonuspunkt}{Bonuspunkte}
3 \renewcommand{\solutiontitle}{\noindent\textbf{Lösung:}\enspace}
4 \chqword{Frage}
5 \chpgword{Seite}
6 \chpword{Punkte}
7 \chbpword{Bonus Punkte}
8 \chsword{Erreicht}
9 \chtword{Gesamt}
10 \hpword{Punkte:}
11 \hsword{Ergebnis:}
12 \hqword{Aufgabe:}
13 \htword{Summe:}
```

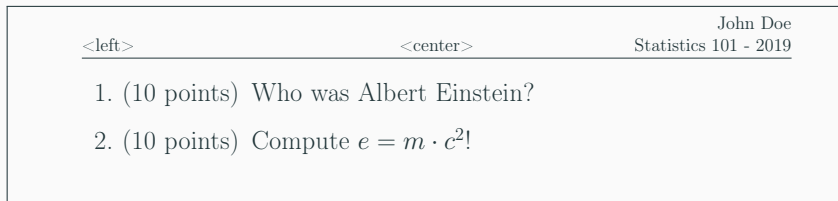
# Creating headers and footer


You can create headers and footers for the first resp. running pages.

```
1 \pagestyle{headandfoot}
2 \firstpageheadrule
3 \runningheadrule
4 \firstpageheader{<left>}{<center>}{John Doe \ Statistics 101 - 2019}
5 \runningheader{<l>}{<c>}{Statistics 101 - 2019}
6 \firstpagefooter{\today}{FOM Essen}{\thepage\,/\/,\numpages}
7 \runningfooter{\today}{FOM Essen}{\thepage\,/\/,\numpages}
8
9 \begin{document}\Large
10 \begin{questions}
11 \question[10] Who was Albert Einstein?
12 \question[10] Compute  $(e = m \cdot c^2)$ !
13 \end{questions}
14 \end{document}
```


Listing 2: Setting header & footer 

# Resulting document



**Figure 2:** Resulting output (top) 



**Figure 3:** Resulting output (bottom) 



# Dividing questions

- Questions can be further divided, exam provides the following environments:
  - parts
  - subparts
  - subsubparts
- Inside these environments individual questions are then added with
  - `\part`
  - `\subpart`
  - `\subsubpart`

# Example for subdivided questions

```
1 \question[10] Who was Albert Einstein?
2
3 \begin{parts}
4   \part[1] Where was he born?
5   \part[4] What has he become famous for?
6   \begin{subparts}
7     \subpart[2] What does  $(e=mc^2)$  mean?
8     \subpart[2] What did he get the Nobelprice for?
9   \end{subparts}
10 \end{parts}
11
12 \end{questions}
13 \end{document}
```


Listing 3: Subdivisions `\part` and `\subpart` 

# Resulting document

<left>	<center>	John Doe Statistics 101 - 2019
--------	----------	-----------------------------------

---

1. (10 points) Who was Albert Einstein?
  - (a) (1 point) Where was he born?
  - (b) (4 points) What has he become famous for?
    - i. (2 points) What does  $e = mc^2$  mean?
    - ii. (2 points) What did he get the Nobelprice for?

**Figure 4:** Resulting output 

## Multiple choice and fill-In questions

- The `exam` class offers several environments for multiple choice and fill-in questions:
  - **choices** for vertical choices using letters
  - **checkboxes** for vertical checkboxes
  - **oneparcheckboxes** for horizontally aligned checkboxes
- with `\fillin[solutiontext]` horizontal lines are created there, where the students are supposed to put their answer

# Multiple choice and fill-In questions I

```
1 \question Who was not a Beatle?  
2  
3 \begin{choices}  
4 \choice John  
5 \choice Paul  
6 \choice George  
7 \CorrectChoice Benedict  
8 \end{choices}
```

Listing 4: Example for choices 


## Multiple choice and fill-In questions II

```
1 \question Who was not a Beatle?  
2  
3 \begin{checkboxes}  
4 \choice John  
5 \choice Paul  
6 \choice George  
7 \CorrectChoice Benedict  
8 \end{checkboxes}
```

Listing 5: Example for checkboxes 


# Resulting document: checkboxes and choices

<left>	<center>	John Doe Statistics 101 - 2019
1. Who was not a Beatle?		
A. John		
B. Paul		
C. George		
D. Benedict		
2. Who was not a Beatle?		
<input type="radio"/> John		
<input type="radio"/> Paul		
<input type="radio"/> George		
<input type="radio"/> Benedict		

**Figure 5:** Resulting output 

# Multiple choice and fill-In questions III

```
1 \question Who was not Beatle?
2
3 \begin{oneparcheckboxes}
4 \choice John
5 \choice Paul
6 \choice George
7 \choice Ringo
8 \CorrectChoice Benedict
9 \end{oneparcheckboxes}
10
11 \question \fillin[James Bond][7em] has the \enquote{
    license to kill}.
```

Listing 6: oneparcheckboxes and fillin 



# Resulting document


<left>	<center>	John Doe Statistics 101 - 2019		
1. Who was not Beatle?				
<input type="radio"/> John	<input type="radio"/> Paul	<input type="radio"/> George	<input type="radio"/> Ringo	<input checked="" type="checkbox"/> <b>Benedict</b>
2. <u>James Bond</u> has the “license to kill”.				

**Figure 6:** Oneparcheckboxes and fillin 

# Resulting document

Using the “answers” class option

<left>	<center>	John Doe Statistics 101 - 2019
<hr/> <p>1. Who was not Beatle?</p> <p><input type="radio"/> John   <input type="radio"/> Paul   <input type="radio"/> George   <input type="radio"/> Ringo   <input checked="" type="radio"/> <b>Benedict</b></p> <p>2. <u>James Bond</u> has the “license to kill”.</p>		

**Figure 7:** Resulting output 

# Creating space for answers

```
1 % simple vertical space
2 \vspace*{<length>}
3
4 % vertical space to the end of the page
5 \vspace*{\stretch{1}}
6 \newpage
7
8 % empty framed box
9 \makeemptybox{<length>}
10
11 % empty framed box to the end of the page
12 \makeemptybox{\stretch{1}}
13 \newpage
```

# More space for answers

```
1 \fillwithlines{<length>} % for lines
2 % Remark: \linefillheight for the inter-line spacing
3
4 \fillwithdottedlines{<length>} % for dotted lines
5 % Remark: distance in \dottedlinefillheight
6
7 \fillwithgrid{<length>} %
8 % \setlength{\gridsize}{5mm}
9 % \setlength{\gridlinewidth}{0.1pt}
10
11 \answerline[answer] % for short answers
```

# Space for answers

<left>	<center>	John Doe Statistics 101 - 2019
--------	----------	-----------------------------------

---

1. Give a short overview of whatever!

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_


2. (5 points) Describe the general theory of relativity!

.....

.....

.....

.....

**Figure 8:** Resulting output 



# Printing solutions

- Global option `answers` controls printing of solutions
- `solution-environment` after each `\question`

```
1 \begin{solution}  
2   Some text containing the solution.  
3 \end{solution}
```

- Some environments for the solution space
  - `solutionorbox`
  - `solutionorlines`
  - `solutionordottedlines`
  - `solutionorgrid`

# Example for the solution environment

```
1 \begin{questions}
2 \question[1] How much does lead (Pb) weigh?
3
4 \begin{solution}
5 Pb weighs \SI{11,342}{\gram\per \centi\meter^3}
6 \end{solution}
7
8 \end{questions}
9 \end{document}
```

Listing 7: Multiple choice 



# Resulting document


Resulting document, class option “answers” set

<left>	<center>	John Doe Statistics 101 - 2019
--------	----------	-----------------------------------

---

1. (1 point) How much does lead (Pb) weigh?

**Solution:** Pb weighs 11.342 g/cm<sup>3</sup>

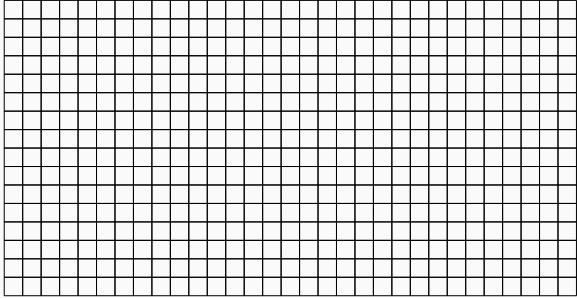
**Figure 10:** Resulting output 


# Example for solutionorgrid

```
1 \question[5] Draw the function  $3x^2+4x+5$ !  
2  
3 \begin{solutionorgrid}[8cm]  
4 \begin{tikzpicture}[baseline]  
5 \begin{axis}[  
6 axis y line=center,axis x line=middle,grid=both,  
7 xmax=5,xmin=-5,ymin=0,ymax=10,  
8 xlabel= $x$ ,ylabel= $y$ ,xtick={-5,...,5},  
9 ytick={0,...,11},anchor=center]  
10 \addplot[smooth,blue,thick,samples=100]{3*x^2+4*x+5} ;  
11 \end{axis}  
12 \end{tikzpicture}  
13 \end{solutionorgrid}
```

Listing 8: solutionorgrid 

# Resulting document (“answers” not set)

<left>	<center>	John Doe Statistics 101 - 2019
1. (5 points) Draw the function $3x^2 + 4x + 5$ !		
		

**Figure 11:** Resulting output 

# Resulting document (“answers” option set)

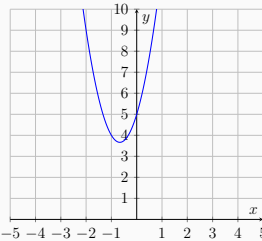
John Doe  
Statistics 101 - 2019

<left> <center>


---

1. (5 points) Draw the function  $3x^2 + 4x + 5$ !

**Solution:**



x	y
-2	5
-1	4
0	5
1	8

**Figure 12:** Resulting output with class option “answers” 

# Printing grade tables

- exam supports the output of grade tables
- grade tables can be arranged per page or question

```
1 \gradetable[v][questions] vertically per question
2 \gradetable[h][questions] horizontally per questions
3 \gradetable[v][pages] vertically per page
4 \gradetable[h][pages] horizontally per page
```

## Example for \gradtable


```
1 \begin{questions}
2 \question[2] What's the specific weight of air?
3
4 \question[2] What's the specific weight of air?
5
6 \end{questions}
7
8 \gradetable[h] [questions]
```

Listing 9: Grade table 

# Resulting document after two L<sup>A</sup>T<sub>E</sub>X-runs

1. (2 points) What's the specific weight of air?
2. (2 points) What's the specific weight of air?

Question:	1	2	Total
Points:	2	2	4
Score:			

**Figure 13:** Resulting output 

# Automating exams

---




## Automating the exam class

- Wouldn't it be nice to randomize exercise values?
- Each student could have his or her own exam, no more cheating!
- To simplify correcting, can't we use e. g. bar codes to assist correcting the exam?

Yes, we can!

# Preparing the L<sup>A</sup>T<sub>E</sub>X-Code I


```
1 \begin{questions}
2 \question[5] Calculate!
3
4 \begin{parts}
5   \part[1] \((12345 + 67890 = \) \fillin[80235]
6 \end{parts}
```

Listing 10: Some simple exercise... 

# Preparing the L<sup>A</sup>T<sub>E</sub>X-Code II


Use `\usepackage{qrcode}` for qrcode support

```
1 \begin{questions}
2 \question[5] Calculate!
3
4 \begin{parts}
5   \part[1] \((12345 + 67890 = \) \fillin[80235] \qrcode
6     {80235}
7 \end{parts}
```

Listing 11: Adding the qrcode 

# Preparing the L<sup>A</sup>T<sub>E</sub>X-Code III

```
1 \begin{questions}
2 \question[5] Calculate!
3
4 \begin{parts}
5   \part[1] \((12345 + 67890 = \) \fillin[80235] \
6     \hfill\qr{80235}\vspace{2em}
7   \part[1] \((12345 + 67890 = \) \fillin[80235] \
8     \hfill\qr{80235}\vspace{2em}
```

Listing 12: Adjust formatting... 


# Resulting document

1. (5 points) Calculate!

(a) (1 point)  $12345 + 67890 =$  \_\_\_\_\_

(b) (1 point)  $12345 + 67890 =$  \_\_\_\_\_

(c) (1 point)  $12345 + 67890 =$  \_\_\_\_\_




**Figure 14:** Resulting output (excerpt) 

# Preparing the Python-Code I

Generate a string adding two random numbers

```
1 from random import randrange
2
3 print(randrange(1000, 10000, 1), ' + ', \
4       randrange(1000, 10000, 1))
```


Listing 13: Multiple choice 

```
1 5274 + 2654
```

# Preparing the Python-Code II

Refactor to get the result of the addition

```
1 from random import randrange
2
3 a = randrange(1000, 10000, 1)
4 b = randrange(1000, 10000, 1)
5 c = a + b
6
7 print(a, '+', b, '=', c)
```


Listing 14: Multiple choice 

```
1 9183 + 9351 = 18534
```

# Preparing the Python-Code III

Create a callable function from the code

```
1 from random import randrange
2
3 def gen_exercise():
4     a = randrange(1000, 10000, 1)
5     b = randrange(1000, 10000, 1)
6     c = a + b
7     print(a, '+', b, '=', c)
8
9 gen_exercise()
```

Listing 15: Creating a function 

```
1 9183 + 9351 = 18534
```



# Preparing the Python-Code IV

```
1 from random import randrange
2
3 def gen_exercise():
4     a = randrange(1000, 10000, 1)
5     b = randrange(1000, 10000, 1)
6     c = a + b
7     a = str(a)
8     b = str(b)
9     c = str(c)
10    print('\n(' + a + ' + ' + b + ' = \n) \\\fillin[' + c +
11          ']' \\\hfill\\qr{code{' + c + '}}\\vspace*{2em}')
12 gen_exercise()
```

Listing 16: Adding the formatting 


# Combining L<sup>A</sup>T<sub>E</sub>X and Python

```
1 \usepackage{pythontex}
2 \begin{document}
3
4 \pyc{from random import randrange}
5
6 \begin{questions}
7 \question[5] Calculate!
8
9 \begin{parts}
10 \part[1] \((12345 + 67890 = \) \fillin[80235] \
    \hfill\qrcode{80235}\vspace{2em}
```

Listing 17: Load pythontex, import randrange 


# Combining L<sup>A</sup>T<sub>E</sub>X and Python

```
1 \pysrc{from random import randrange}
2
3 \begin{pycode}
4 def gen_exercise():
5     a = randrange(1000, 10000, 1)
6     b = randrange(1000, 10000, 1)
7     c = a + b
8     a = str(a)
9     b = str(b)
10    c = str(c)
11    return '\\(' + a + ' + ' + b + ' = \\)' \\fillin[' + c
12           + '] \\hfill\\qrcode{' + c + '}\\vspace*{3em}'
13 \end{pycode}
```

Listing 18: Add the function 

# Combining L<sup>A</sup>T<sub>E</sub>X and Python

```
1 \begin{questions}
2 \question[5] Calculate!
3
4 \begin{parts}
5 \part[1] \py{gen_exercise()}
6 \part[1] \py{gen_exercise()}
7 \part[1] \py{gen_exercise()}
8 \part[1] \py{gen_exercise()}
9 \part[1] \py{gen_exercise()}
10 \end{parts}
```

Listing 19: Add the code to generate the exercise 

# Resulting document

1. (5 points) Calculate!

(a) (1 point)  $5989 + 5087 =$  \_\_\_\_\_



(b) (1 point)  $7065 + 1500 =$  \_\_\_\_\_



(c) (1 point)  $6694 + 9484 =$  \_\_\_\_\_






**Figure 15:** Resulting output (excerpt) 

# Summary

- Creating exams in L<sup>A</sup>T<sub>E</sub>X is easy ✓
- The `exam` class offers rich set of functions ✓
- Generating individual exams per group/student can be done ✓
- For questions and comments please contact me

ziegenhagen@gmail.com

# This presentation

- Clicking  and  opens the example files (at least in Adobe Reader)
- $\text{\LaTeX}$ -source 
- Document class: Beamer
- Document theme: Metropolis
- Font: IBM Plex Sans