

Assessing Accessibility Automatically

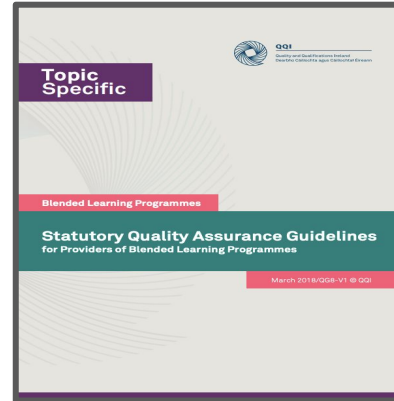
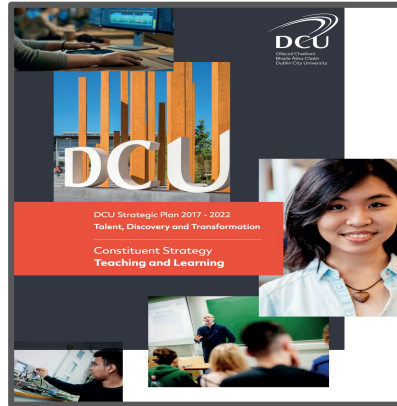
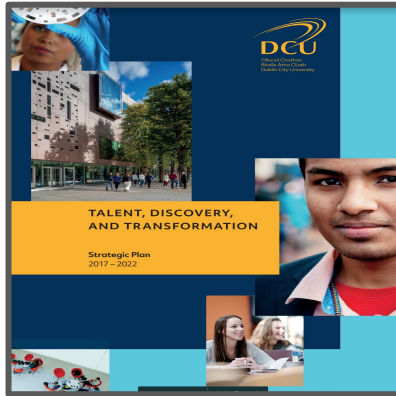
Dr Mark Glynn*, Gavin Henrick**, Karen Holland**, Dr Donal Fitzpatrick*

*DCU, ** Brickfield Education Labs

@glynmark, @ghenrick

<http://bit.ly/wcol2019aaa>

What are the drivers?



“To enhance the inclusivity of the learning experience, we will promote the principles of Universal Design in the design and delivery of all our programmes.”

EU Legislation

Member States shall ensure that public sector bodies take the necessary measures to make their websites and mobile applications more accessible by making them perceivable, operable, understandable and robust.



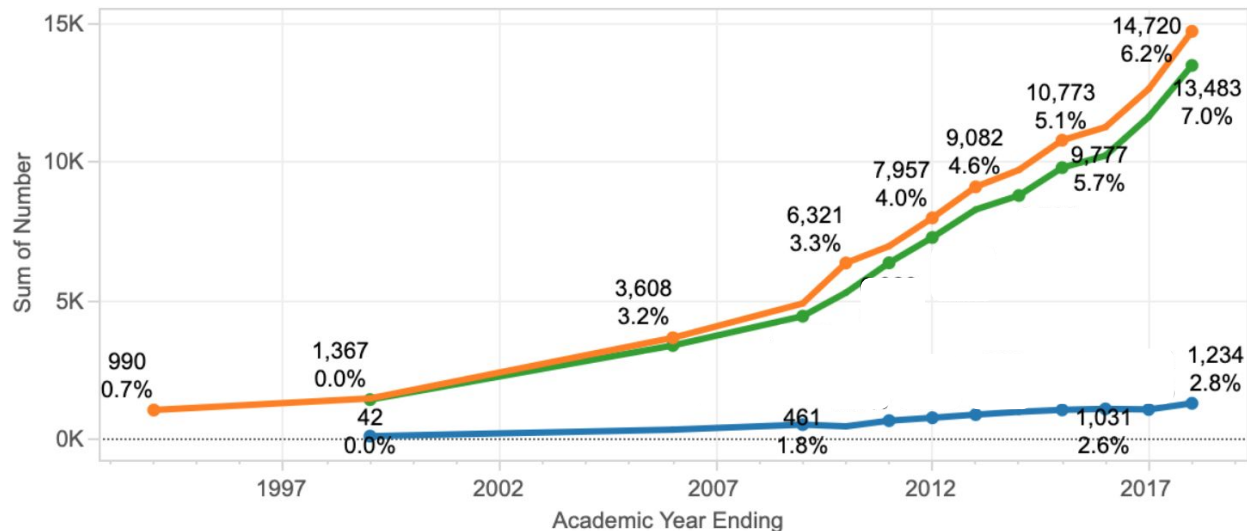
Timeline



Websites created after 23 September 2018 will have to be accessible by 23 September 2019.

Number of students reporting a disability

Year on Year Totals 17/18



Type

- (All)
- Postgraduate
- Total
- Undergraduate

Type

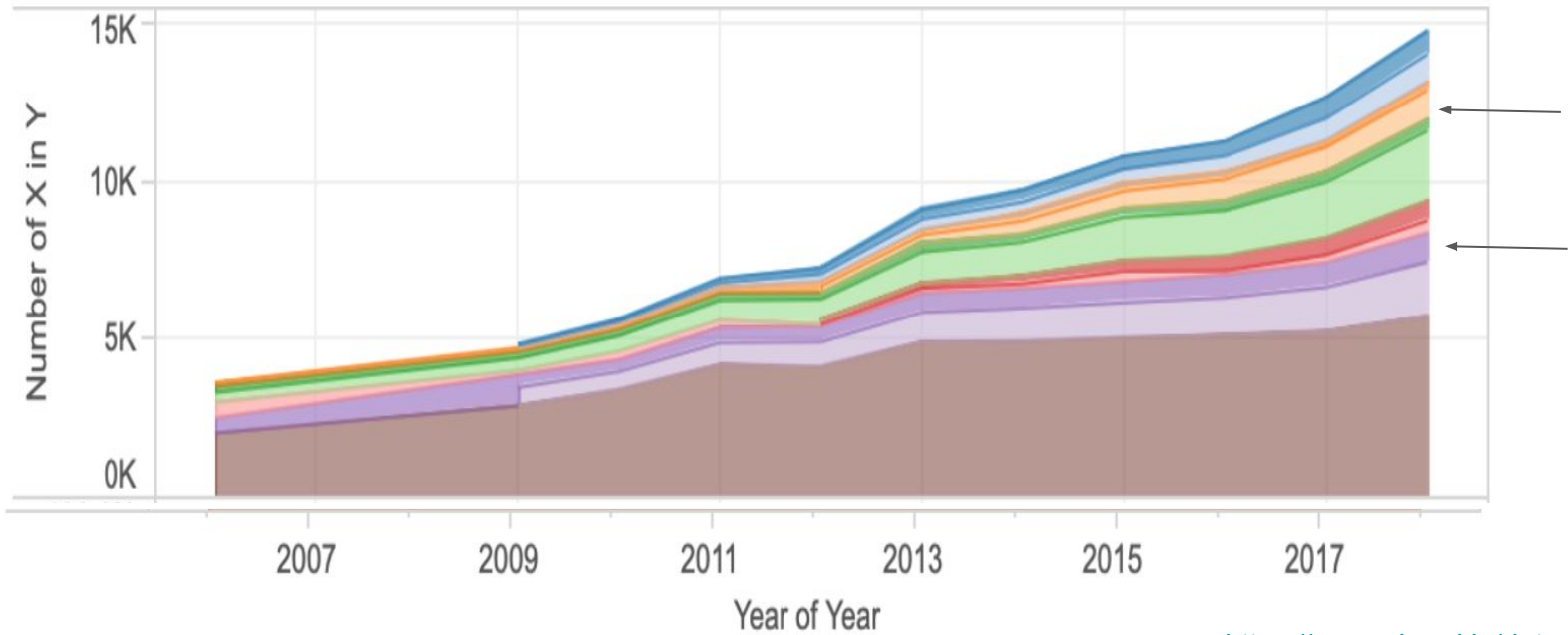
- Postgraduate
- Total
- Undergraduate

Year Ending

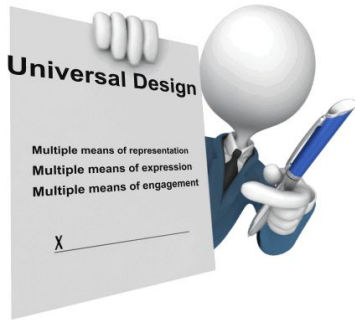




Year on Year 17/18



Working group



Policy



Staff Resources



Staff workshops



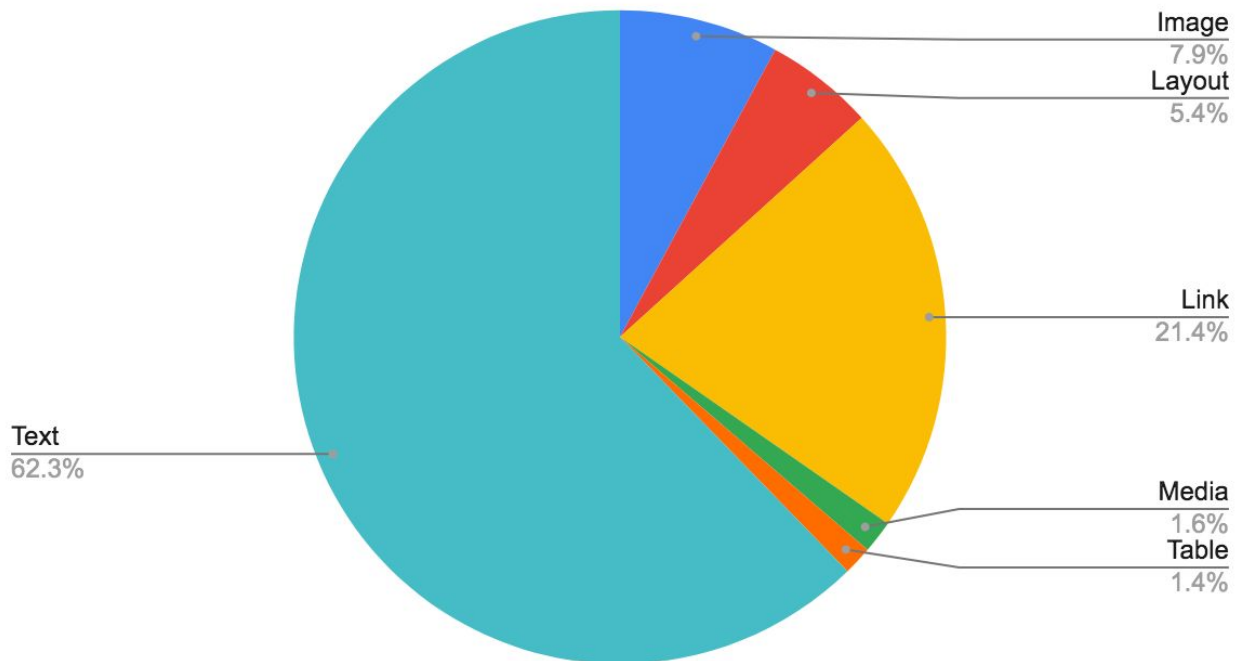
Moodle reports

Most common issues found

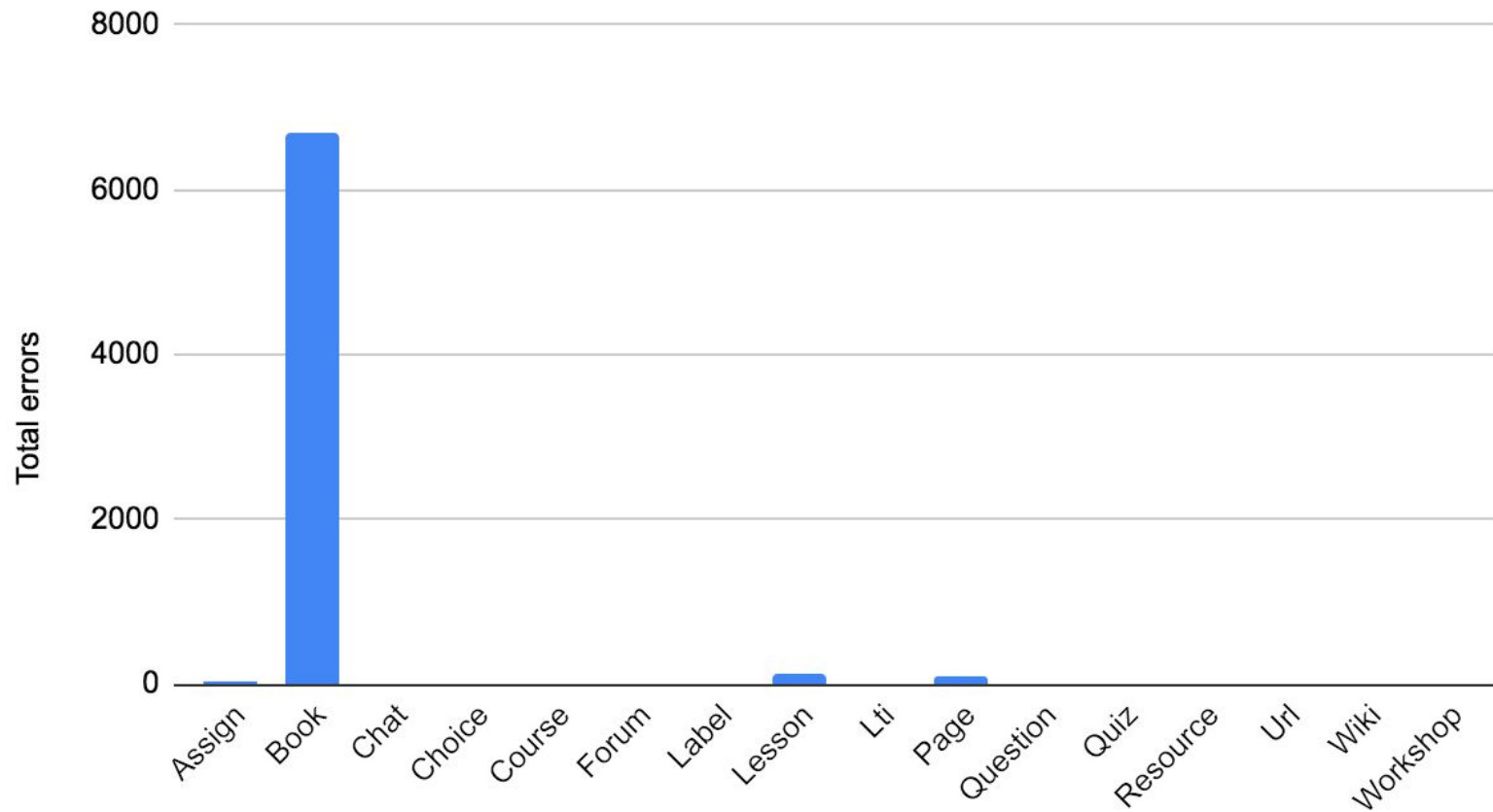
1. Inappropriate text formatting
2. Links using unsuitable text in name
3. Images which were not decorative but had no description or inappropriate one
4. Layout issues with no headings or imperfect structure

Number of errors per type of problem

Total errors in each group type of errors



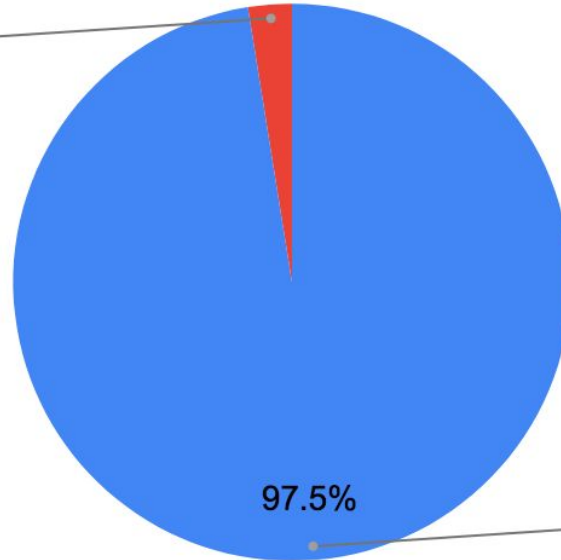
Number of errors in different areas of Moodle



Checks - Pass and Fail Rate

Audit Checks Completed

Failed
2.5%



Passed
97.5%

Problem area pass rates per course

Course	Image	Layout	Link	Media	Table	Text
Course 1	99%	100%	100%	100%	100%	99%
Course 2	99%	100%	100%	100%	100%	99%
Course 3	98%	97%	96%	99%	99%	97%
Course 4	99%	99%	99%	100%	99%	99%
Course 5	99%	89%	95%	95%	99%	94%
Course 6	99%	94%	92%	99%	100%	95%
Course 7	99%	100%	99%	100%	100%	99%
Course 8	97%	97%	97%	99%	99%	93%
Course 9	100%	99%	99%	99%	99%	99%
Course 10	97%	94%	98%	97%	100%	96%

Layout A

Some Possible Criticisms of Hume's Position

Hume, like other empiricists, claims all knowledge is the result of experience. This claim is at least questionable because no amount of experience will of itself organise the experience. This is the criticism from rationalism. If there is nothing in the mind before experience how come we bring to experience coherence and structure? The

Layout B

Some Possible Criticisms of Hume's Position

Hume, like other empiricists, claims all knowledge is the result of experience. This claim is at least questionable because no amount of experience will of itself organise the experience. This is the criticism from rationalism. If there is nothing in the mind before experience how come we bring to experience coherence and structure? The

Layout A - Wrong HTML

```
<p></p><p><b>Some Possible Criticisms of Hume's Position</b></p>
```

<p>Hume, like other empiricists, claims all knowledge is the result of experience. This claim is at least questionable because no amount of experience will of itself organise the experience. This is the criticism from rationalism. If there is nothing in the

Layout B - Correct HTML

```
<h3>Some Possible Criticisms of Hume's Position</h3>
```

<p>Hume, like other empiricists, claims all knowledge is the result of experience. This claim is at least questionable because no amount of experience will of itself organise the experience. This is the criticism from rationalism. If there is nothing in the

Instructions A

Complete the following programming exercises in Python. Aim to make your code as concise and efficient as possible. Write functions to be self-contained units of code, i.e. the code inside your functions should **not** rely on variables you define outside the function (i.e. *global variables*).

Submit your answers as a ***single text .TXT document***, with each answer clearly numbered and your code properly formatted/indented (use Ctrl+A then Ctrl+I in Code::Blocks to properly format your code).

This assignment is due 11:00 pm Sunday 29th February 2019. No late submissions accepted.

Instructions B

Instructions

Complete the following programming exercises in Python.

1. Aim to make your code as concise and efficient as possible.
2. Write functions to be self-contained units of code, i.e. the code inside your functions should not rely on variables you define outside the function (i.e. *global variables*).
3. Submit your answers as a ***single text .TXT document***, with each answer clearly numbered and your code properly formatted/indented (use Ctrl+A then Ctrl+I in Code::Blocks to properly format your code).

Deadline

This assignment is due 11:00 pm Sunday 29th February 2019. No late submissions accepted.

Table A

Study competencies/approach to learning
Study in a quiet place
Study at a desk
Free up blocks of time to study
Set up and stick to a daily/weekly schedule

Table B

Study habits inventory

Study competencies/approach to learning

Study in a quiet place

Study at a desk

Free up blocks of time to study

Set up and stick to a daily/weekly schedule

Reading List A

Websites

DCU student learning <http://dcu.ie/studentlearning/index.shtml>

How-to-Study.com, Study Skills Resources site <http://www.how-to-study.com/>

(Note that this is an American site so spelling and grammar may vary from the standard UK English.)

Reading List B

Websites

[DCU student learning](#)

[How to Study dot com, Study Skills Resources site](#)

(Note that this is an American site so spelling and grammar may vary from the standard UK English.)

Conclusions

