Xerte Online Toolkits Documentation
April 2019

Variables in Xerte Online Toolkits

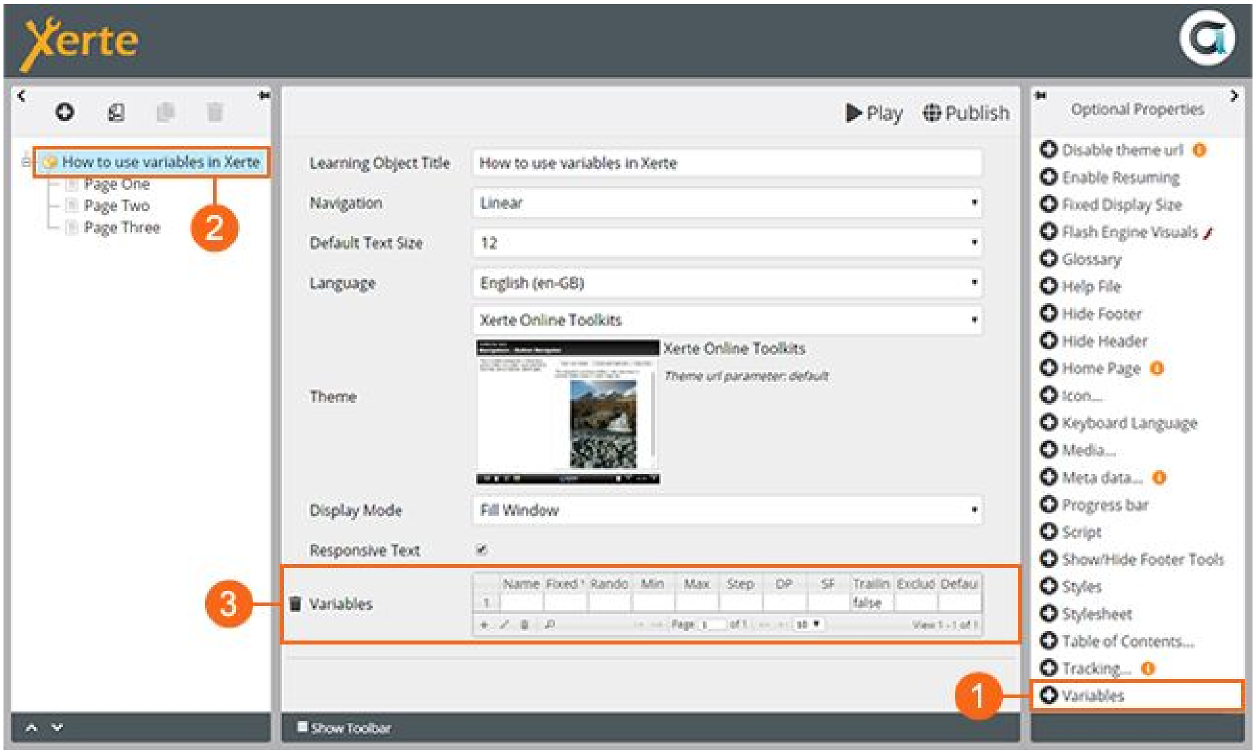
The variables optional property is available in Xerte Online Toolkits from version 3.4 onwards. Variables can be used to add variety to the data that students are shown as the value of each variable is calculated (according to the parameters set by the author) when the project is viewed. For example, instead of including the sum 1 + 2 = 3, variables allow the sum a + b = c to use a variety of numbers.

Using variables can encourage the reuse of content by students and helps to ensure they understand the logic behind what they are learning rather than just memorising one example.

This guide covers the basics of setting up and using variables, as well as more advanced functionality such as the new ability to allow students viewing projects to set the value of variables themselves:

Setting Up Variables

Variables can be used in any Xerte Online Toolkits template project[1] by adding the 'Variables' optional property (1) at project level (2). A datagrid (3) will appear in the editor in which the parameters for your variables can be set:



Learning Object Title: How to use variables in Xerte

Navigation: Linear

Default Text Size: 12

Language: English (en-GB)

Theme: Xerte Online Toolkits

Xerte Online Toolkits

Theme url parameter: default

Display Mode: Fill Window

Responsive Text: ☑

Variables

	Name	Fixed	Rando	Min	Max	Step	DP	SF	Trailin	Exclud	Defau
1									false		

Page 1 of 1 10 View 1 - 1 of 1

Each line in the datagrid represents one variable. Double click a line to edit a variable's parameters or click the 'Add new row' button to create a new variable via a dialog window.

Each column in the datagrid represents a different variable parameter:

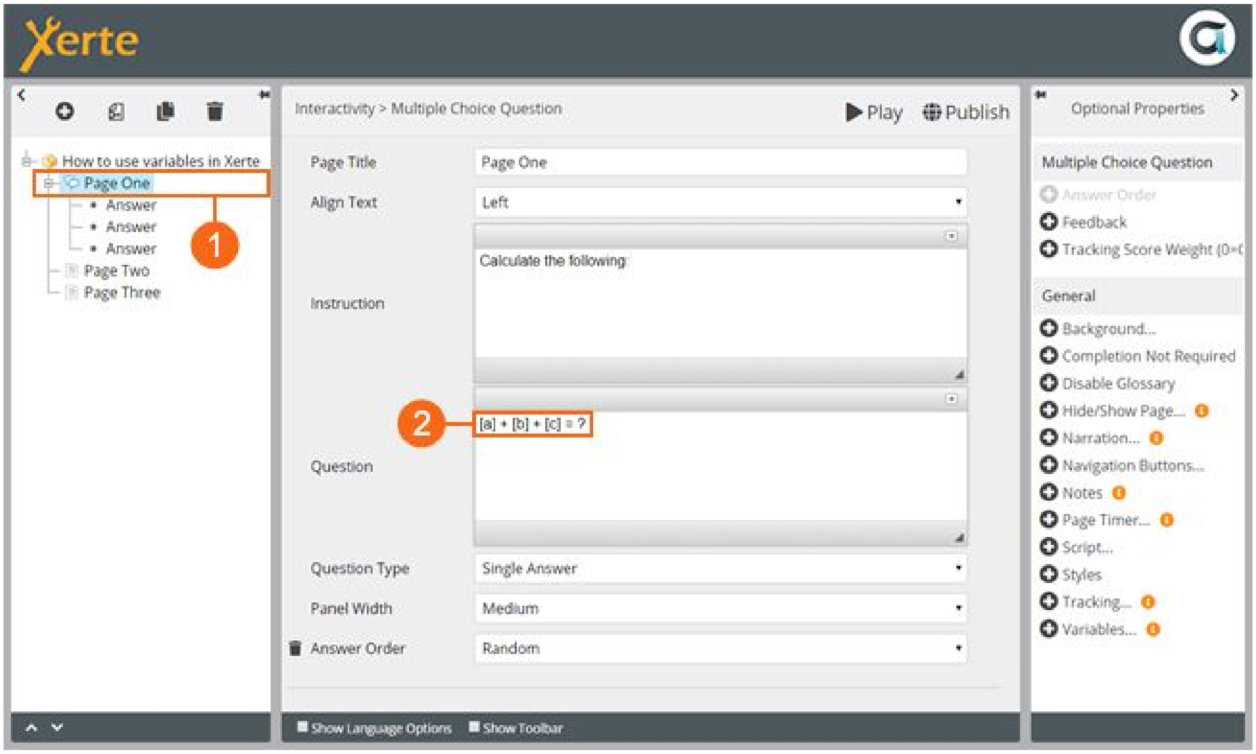
1. **Name** : the unique name for the variable that is used when referencing it on a page
2. **Fixed value** : a fixed value that will always be used for the variable
3. **Random** : a comma separated list of possible values for the variable from which one is chosen at random
4. **Min** : use in conjunction with *max* to set boundaries for a number chosen at random
5. **Max** : use in conjunction with *min* to set boundaries for a number chosen at random
6. **Step** : use with *min* and *max* to set the steps between the possible randomly chosen numbers
7. **DP** (decimal points) : how a value should be rounded (e.g. 2 DP would change 1.541 to 1.54)
8. **SF** (significant figures) : the number of significant figures to display (e.g. 2 SF would change 329 to 330)
9. **Trailing zeros** : whether *DP* should include trailing zeros (e.g. if true then 2 DP would change 1.5 to 1.50)
10. **Exclude** : a comma separated list of any values that will not be accepted as valid values
11. **Default** : the fallback value used if calculating a valid value using the other parameters proves difficult

All variables must have a name but any combination of the other parameters can be used to fit your requirements. Note that it is possible to calculate a variable based on the value of another by putting the name of the variable you are referencing in square brackets e.g. [a]

[1] Variables are not currently available in the Bootstrap or Decision Tree templates.

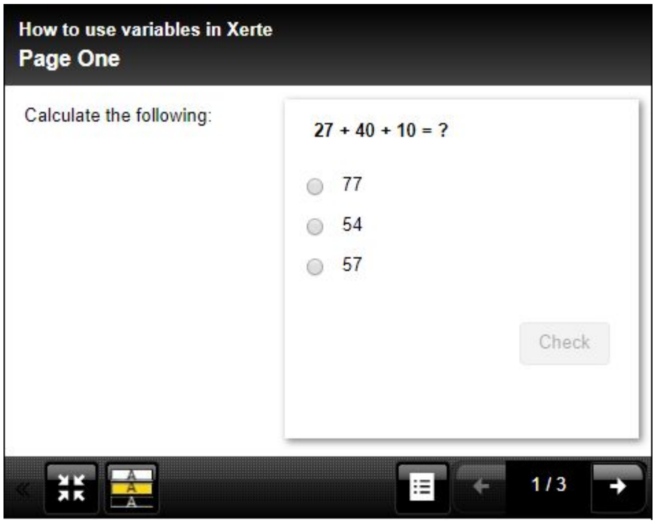
Using Variables on Pages

Once variables have been set up they can be used on any page in your project. Select the page the variable will appear on (1) and enter the name of the variable surrounded by square brackets into a text field in the editor (2):



Page Title: Page One
Align Text: Left
Instruction: Calculate the following:
Question: [a] + [b] + [c] = ?
Question Type: Single Answer
Panel Width: Medium
Answer Order: Random

When the project is viewed the variable values will be calculated from the parameters you have set and any instances of the square brackets and variable names will be replaced with the relevant value:



Calculate the following:

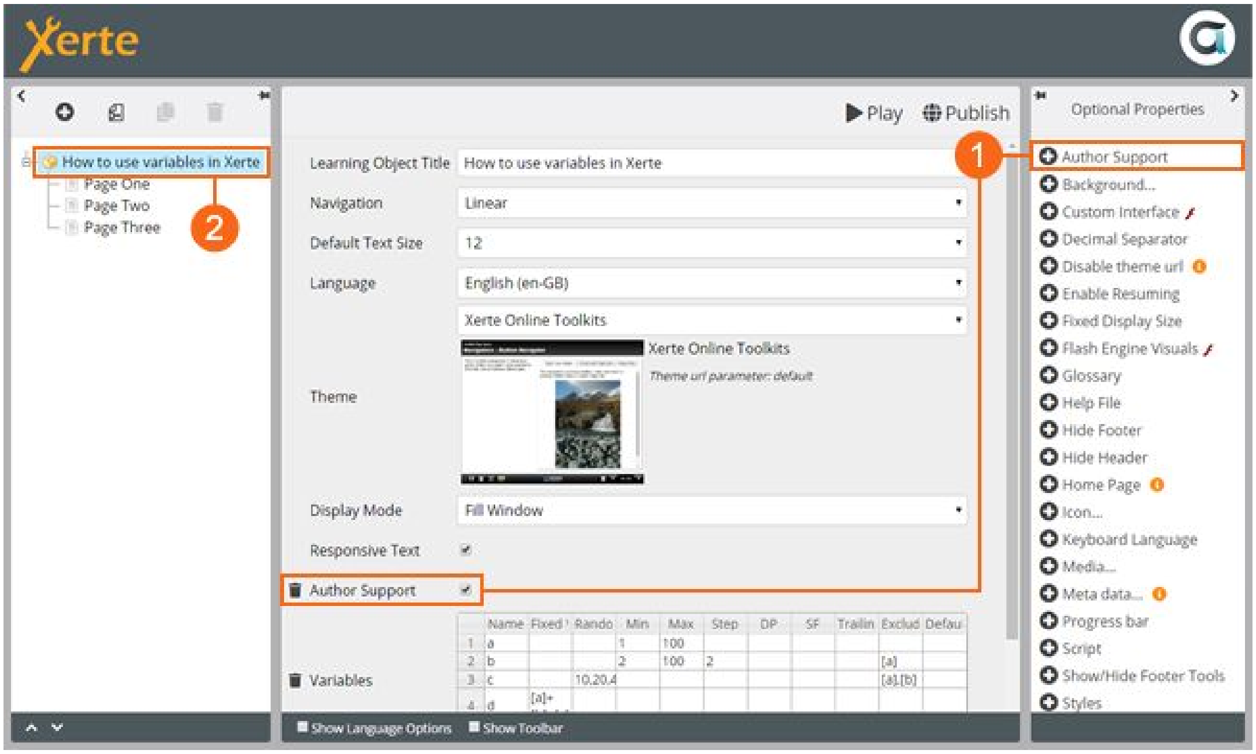
27 + 40 + 10 = ?

○ 77
○ 54
○ 57

Note that when using variables in multiple choice questions and answers it is a good idea to set up the wrong answer options using variables too. This means that each time the project is viewed it won't just be the question and correct answer that changes - the incorrect answers will also change in a meaningful way.

Testing & Troubleshooting

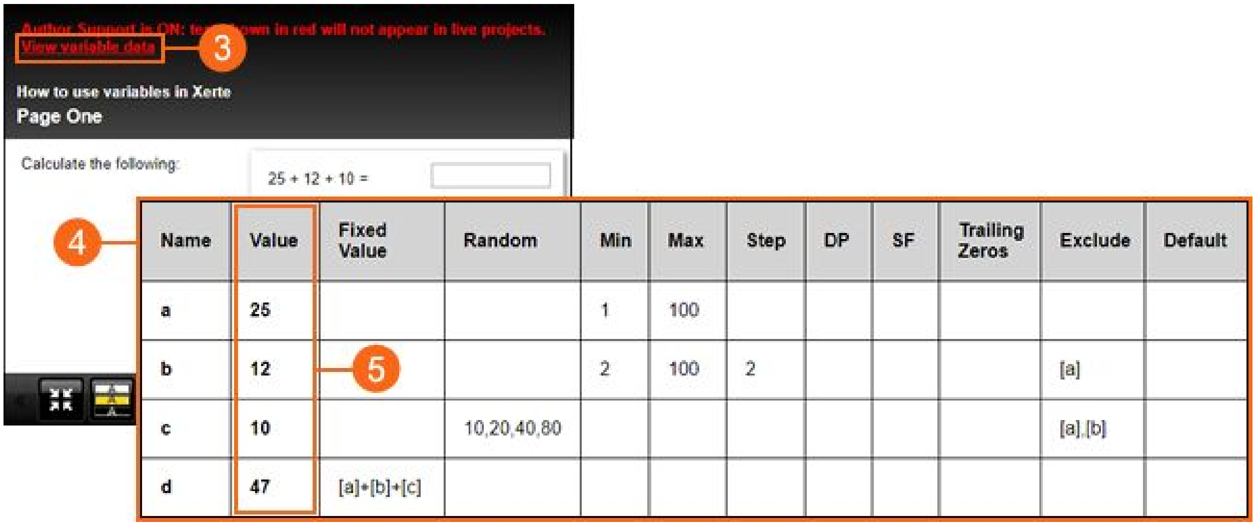
As the value of variables are only calculated when the project is viewed and because they often depend upon the value of other variables, it is important to thoroughly test how they are being calculated before your project goes 'live'. It is recommended that you add the project level 'Author Support' optional property (1) at project level (2) to help you do this:



Learning Object Title: How to use variables in Xerte
Navigation: Linear
Default Text Size: 12
Language: English (en-GB)
Theme: Xerte Online Toolkits
Xerte Online Toolkits
Theme url parameter: default
Display Mode: Fill Window
Responsive Text: ☑
Author Support: ☑
Variables:

	Name	Fixed	Rando	Min	Max	Step	DP	SF	Trailin	Exclud	Defau
1	a			1	100						
2	b			2	100	2				[a]	
3	c		10,20,4							[a],[b]	
4	d	[a]+									

When a project is previewed with author support on a red link will appear in the header of your project (3). Click this to access information about all of the variables you have created (whether shown on the current page or not). This information includes the parameters used in attempting to calculate them (4) and the current calculated value (5):



Author Support is ON: te own in red will not appear in live projects.
View variable data

How to use variables in Xerte
Page One

Calculate the following:
25 + 12 + 10 =

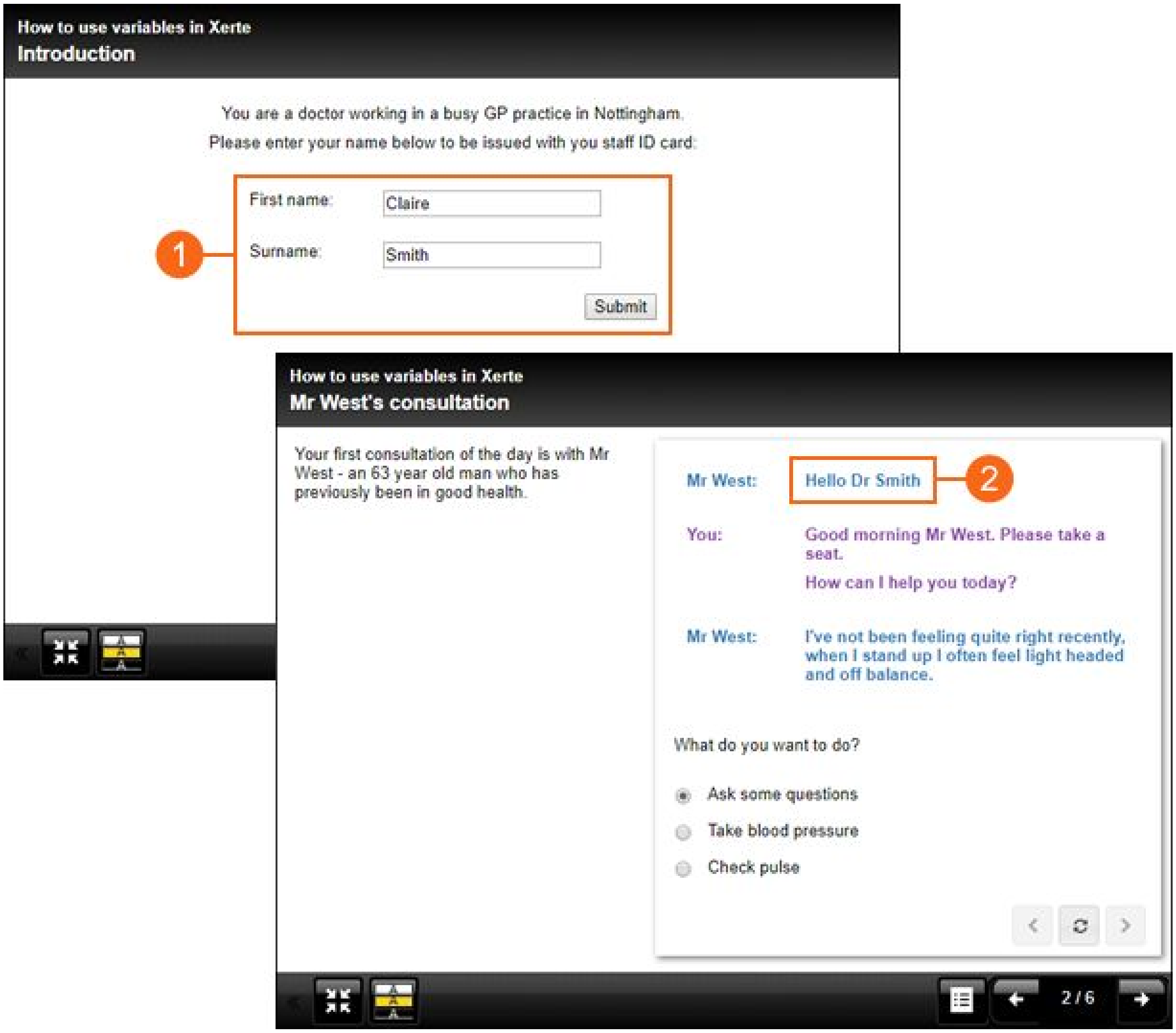
Name	Value	Fixed Value	Random	Min	Max	Step	DP	SF	Trailing Zeros	Exclude	Default
a	25			1	100						
b	12			2	100	2				[a]	
c	10		10,20,40,80							[a],[b]	
d	47	[a]+[b]+[c]									

Preview your project and check the values that are calculated for your variables several times to make sure they are being calculated reliably. When variables fail to calculate a message will be shown in the author support window explaining what the problem is. Common causes of variables that always fail to calculate are when variable values are circular (e.g. **a** is calculated using the value of **b** but **b** is calculated using the value of **a**) and when all possible values of a variable are excluded.

If your variables include complex calculations occasionally it might not be possible to calculate a figure that fits all the criteria you have set. If this happens up to 100 attempts will be made to recalculate the variables. If it is still unsuccessful the *default* value will be used. It is therefore a good idea to always include a *default* value when creating complex variables so that the project contents will always display correctly for students. The author support window includes a flag when the variable has had to fall back to the default value.

Changing Variable Values at Runtime

The next release of Xerte Online Toolkits will include the ability to allow students viewing projects to change the value of variables themselves. Let's first look at this from a student's point of view:



How to use variables in Xerte

Introduction

You are a doctor working in a busy GP practice in Nottingham.
Please enter your name below to be issued with you staff ID card:

First name: Claire

Surname: Smith

Submit

How to use variables in Xerte

Mr West's consultation

Your first consultation of the day is with Mr West - an 63 year old man who has previously been in good health.

Mr West: Hello Dr Smith

You: Good morning Mr West. Please take a seat.
How can I help you today?

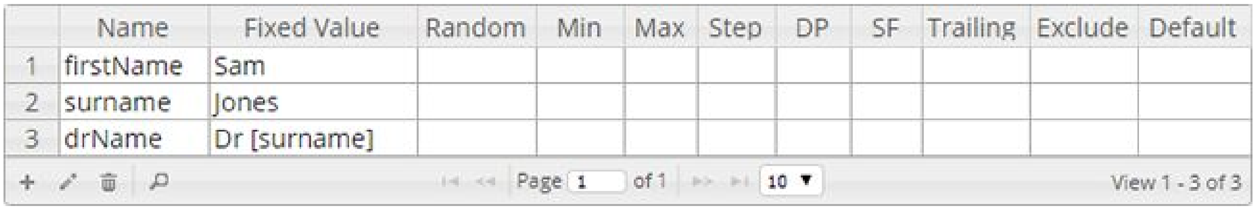
Mr West: I've not been feeling quite right recently, when I stand up I often feel light headed and off balance.

What do you want to do?

● Ask some questions
○ Take blood pressure
○ Check pulse

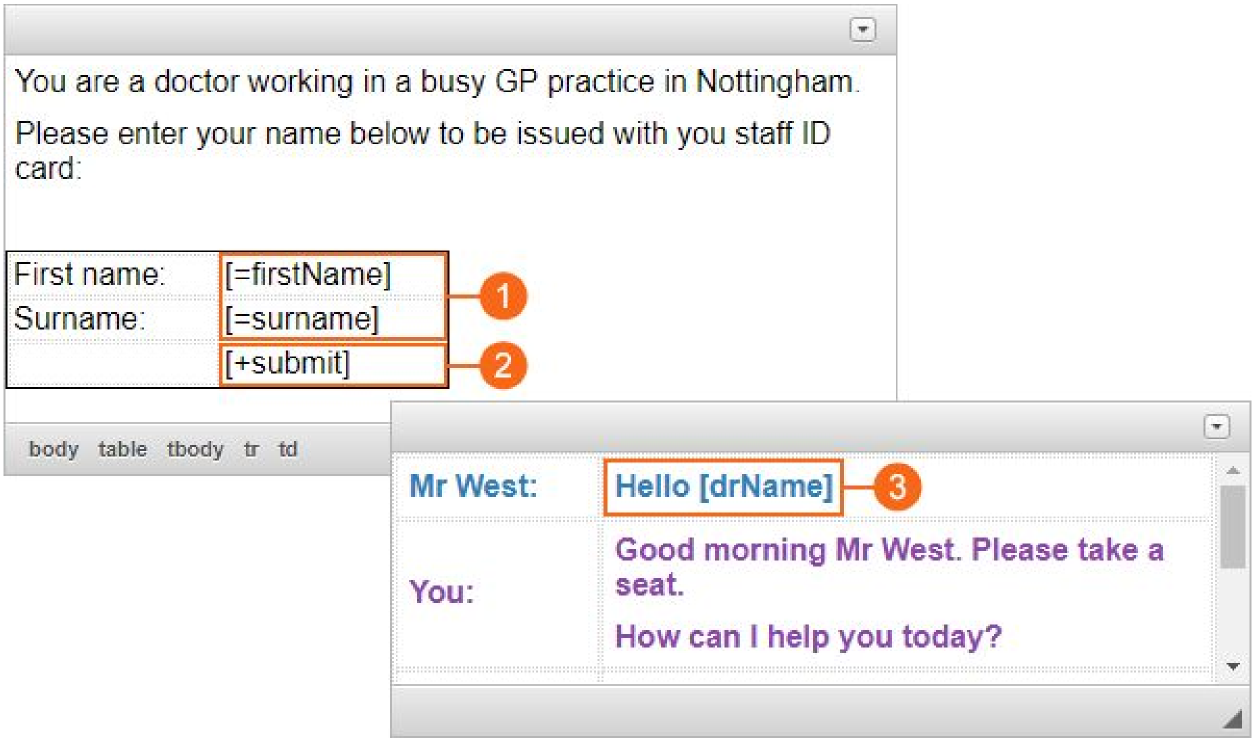
In this example the student is prompted to enter their name on the first page of the project (1). Their name is then used throughout the project (2) to make the scenario appear more personalised.

So how was this project made? Three variables are set up in the normal way with default values set in the *fixed value* column so that students can still work through the project without entering their own details if they wish:



	Name	Fixed Value	Random	Min	Max	Step	DP	SF	Trailing	Exclude	Default
1	firstName	Sam									
2	surname	Jones									
3	drName	Dr [surname]									

The mark-up used in the editor to specify when the text fields and submit buttons should be inserted onto a page is shown below. To create a text field used to alter a variable's value the markup is square brackets surrounding an equals sign and the variable name (1). To create a submit button square brackets should surround a plus sign and the text 'submit' (2). Note that simply including the text fields is not enough - the variable values will not be changed until submitted so it is essential to include a submit button on pages with variable text fields.



You are a doctor working in a busy GP practice in Nottingham.

Please enter your name below to be issued with you staff ID card:

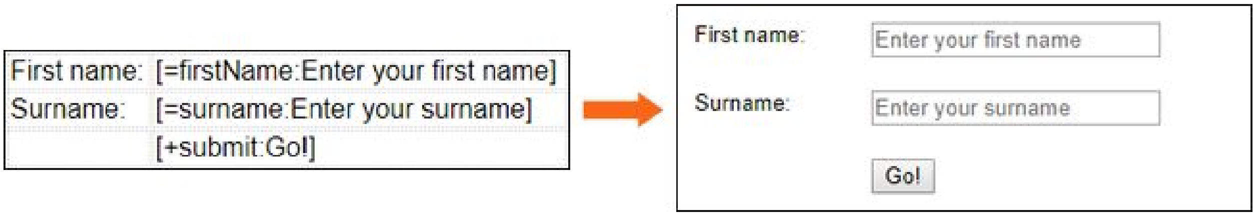
First name:	[=firstName]
Surname:	[=surname]
	[+submit]

Mr West:	Hello [drName]
You:	Good morning Mr West. Please take a seat. How can I help you today?

Variables that can be changed by students can be inserted into the page content in the normal way using the square bracket mark-up (3). Note that in the example above the text fields change the variables *firstName* and *surname* but when submitted the value of *drName* will also change as it is dependant on one of the newly changed variables.

Advanced Options

Create **placeholder text** for a text field or change the **default label** that appears on a submit button using the following mark-up:



First name:	[=firstName:Enter your first name]
Surname:	[=surname:Enter your surname]
	[+submit:Go!]

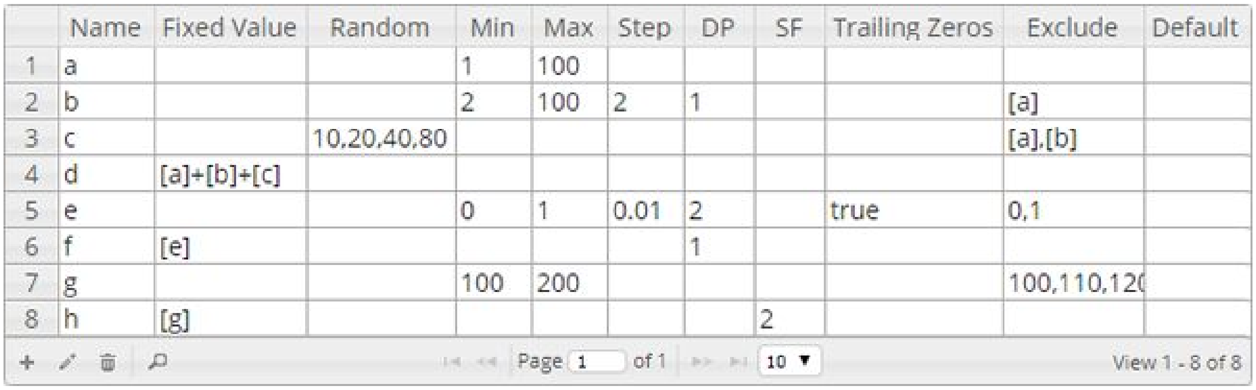
First name:	Enter your first name
Surname:	Enter your surname
	Go!

Add the **'Variables' optional property** to individual pages to control the text that appears after the submit button is clicked & decide whether instances of variables on the page should be updated if changed by the student.

Examples

The following examples illustrate how each of the parameters can be used when constructing variables. The resulting project containing these examples can be viewed in this Xerte project.

Numerical Calculations:

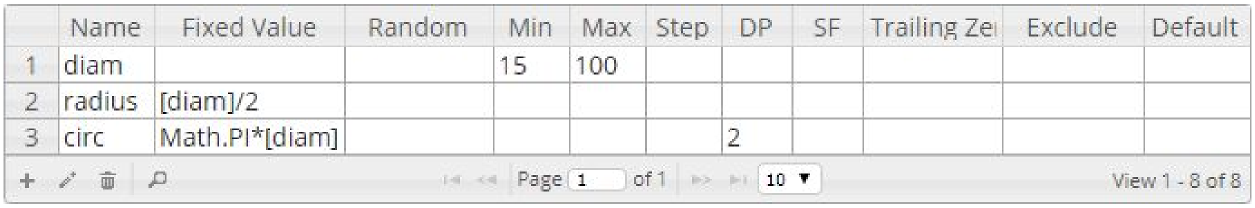


	Name	Fixed Value	Random	Min	Max	Step	DP	SF	Trailing Zeros	Exclude	Default
1	a			1	100						
2	b			2	100	2	1			[a]	
3	c		10,20,40,80							[a],[b]	
4	d	[a]+[b]+[c]									
5	e			0	1	0.01	2		true	0,1	
6	f	[e]					1				
7	g			100	200					100,110,12	
8	h	[g]						2			

These show the following:

a = any whole number between 1 and 100 (using *min* & *max* columns)
b = any even number between 2 and 100 that isn't **a** (using *exclude* column to not allow it to be equal to **a** & the *step* column of 2 means that the possible numbers it can be between 2 and 100 will always be even… 2, 4, 6, 8 etc.)
c = a number taken at random from the *random* column that doesn't equal either **a** or **b** (*exclude* column)
d = the sum of **a**, **b** and **c**. This is in the *fixed value* column as it is always calculated in the same way. It will of course still change as **a**, **b** and **c** will be recalculated each time the project is viewed
e = a decimal number between 0 and 1 (using *min*, *max* & *step* columns). This is forced to always be 2 decimal places (*DP* column) as *trailing zeros* is set to true (e.g. 0.2 will be 0.20)
f = the result of rounding **e** (*fixed value* column) to one decimal place (*DP* column)
g = any number between 100 and 200 (using *min* & *max* columns) that isn't divisible by 10 (*exclude* column)
h = the result of calculating **g** (*fixed value* column) to two significant figures (*SF* column)

Javascript Math Object:



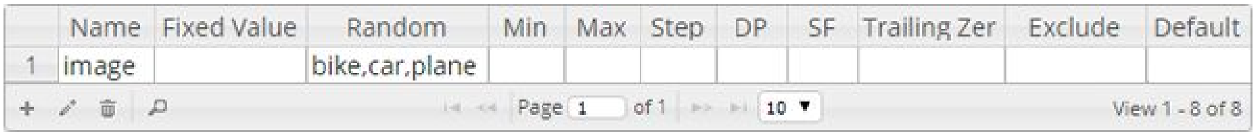
	Name	Fixed Value	Random	Min	Max	Step	DP	SF	Trailing Ze	Exclude	Default
1	diam			15	100						
2	radius	[diam]/2									
3	circ	Math.PI*[diam]					2				

These show the following:

diam = any whole number between 15 and 100 (using *min* & *max* columns)
radius = half of **diam**. This is in the *fixed value* column as it is always calculated in the same way. It will of course still change as **diam** will be recalculated each time the project is viewed
circ = the circumference of a circle with the diameter of **diam**. This is calculated using the JavaScript Math.PI property.

The JavaScript Math object allows you to perform a variety of mathematical tasks on numbers. See w3schools Math Object pages for further information and examples.

Text Strings:



	Name	Fixed Value	Random	Min	Max	Step	DP	SF	Trailing Zer	Exclude	Default
1	image		bike,car,plane								

This shows the following:

image = a word taken at random from the *random* column

Words and phrases can be included in the *fixed value*, *random*, *exclude* and *default* columns. Text variables are added to pages in your project in the same way as numerical variables by using the square bracket mark-up. They can be used as simple text as well as to change the name of the image file shown or a link that is included on the page.