

# Functions

## Overview

**Functions** are <u>test cases</u> that can be reused for a specific task and can be called multiple times throughout a test case with or without different arguments. For example, take the case of an ecommerce shop. Test cases like *Add to Cart* are common actions inside their web services and mobile applications. These test cases are repeated actions and can be listed under the functions category.

Functions can be referred to in test cases using its **Link Name** or using the task <u>call function</u>.

#### **Create Function**

Creating a function is similar to a creating a test case. To do so:

1. Click Create Test Case.



2. Enter the test case details, such as *Name*, *Link Name*, and *Module* to which it needs to be added. Additionally, providing a Description offers a better overview of the test case, but is not mandatory.



The link name will be unique for each function and will be auto-filled by default, similar to the test case name. They should be descriptive and accurately reflect the purpose of the function. For example, if your test case name is **Contact Us**, the link name would be contact\_us by default. However, they can be edited as per the following conditions:

- The link name only supports lower case letters. For example, if the test case name is Addtocart, the link name would be addtocart.
- Special characters are not supported. They will be replaced with underscores (\_). For example, if the test case name is Replacement&Return , the link name would be replacement\_return.
- If there are two or more special characters consecutively, they will be replaced with a single underscore character. For example, if the test case name is details & @contact, the link name would be details\_contact.
- 3. Select the **Mark test case as a function** checkbox below the description to convert the test case into a function.
- Note: If the test case isn't marked a function, then it's considered as a regular test case.
  - 4. Mention the return type of the function. This specifies the data type of the value that the function will return after its execution. For example, if we create a function to greet every user inside an ecommerce site, then the function is a void type, as it doesn't have a return statement. It will simply print a greeting message for

every user. However, if the function was created to sum the total price of a product and the price value are integer data types, then the function is an integer type as it gives an integer sum in return.

5. Specify the arguments, if any. The arguments of a function are the input values passed when it is called inside a test case or another function, which the function can use to perform its operations. For example, an ecommerce function to show featured products doesn't take arguments but displays products, while a function calculating the total price takes quantity and price per unit arguments and returns the total cost.

zw	🐻 Test Cases Manage	Buy a Product	+ Create Test Case
Eo		E EP E ☉ Record 🖳 Web Browser 🗸	• Ready 🗠 🔚 🗮
Test Cases		General 1 accounts. 2 functions Create Test Case ×	
Test Suites		OpenURL 4 click(".t. Description	
		Click 5 click/Ezz Add to cart function allows the castmers to choose items to purchase without add to cart function allows the castmers to choose items to purchase without add to cart function the same of the	
EB Test Plans		SetValue 8 mit(2):	
		Assert Text 9 Click("#z: 10 writ/10 * Mark test case as a function ?	
IIO. Results		Assert Value 11 cltcl(2, F. Bellum Tune	
- 01		Wait 13 click("pc void	
Users		14 click(ui.; 15 click(ui.;	
~		Form Actions         16         Click(ui, :         Specify Arguments           17         assertText         handling.STOP.ON_ERROR);	
्ुः Settings		Select 18 Name Type Value	
		Beselect ∷ ItemName string ✓ Chrono 44	
		Uncheck III ItemNo Int V	
		Clear	
		Upload Carbon	
		Assertion	
		Assert Selected	
		Assert Disabled	
		Assert Visible	
		Assert Focused	
		Assert Hidden	
		Assert ImageExists	
		Assert File Download	
		Assert Not Present	
•			
_ <b>()</b>	+ Add Module <=	8.8	ave 🕞 Run 👻

- Note: The fields return type and specify arguments will only be enabled once you check the box to make the test case a function.
  - 6. Once the required details are filled, click **Create**.

### **Using functions**

• To use functions in a test case or an existing function inside builder mode, simply call the function using the <u>Call Function</u> task. Drag-and-drop the **call function** task from the task panel to the editor. The right panel for the arguments will have a drop-down menu. Choose the module from the drop-down menu to select the desired function.

zw	B Test Cases Manage	Add to cart					+ Create	Test Case
5	Search Q 7		⊙ Rei	cord	Not Saved 🖵 Web Browser 🗸	-S	● Draft 〜	
Test Cases		Get Attribute Get Count	1 -	Θ	Open URL "https://zwatch.zohocommerce.com" 'same window"	∫x Call Fund	tion	Û
E.		Get Element Width	2	0	StrollTo ukhome_watch_shopping.h2	Select Functi	on Liliks	~
Irô. Resulta		Info	3	0	Click ui.home_watch_shopping.ma_watch	<ul> <li>Review</li> <li>Search</li> </ul>	1	
。祭 Users	<ul> <li>▼ Difference Differ</li></ul>		5 —	Ø	Click ui.home_watch_shopping.a	✓ E Shopp	ing r_Add_To_Cart	
(j) Settinos		Execute JS	6 —	ð	Click ulcart_watch_shopping.button	J* Go_To_	Home	
		Else IF Else	7 -	fx	CallFunction ()			
	C Quick Links     C Review	Set Variable Call Function						
	Search     Shopping	Finally Continue On Error						
	Social Media     ZylkerWatches	Browser Actions						
		Switch To Get URL						
		Refresh Resize						
		Forward Back						
		Close						
	+ Add Module <=					8	iave 🕞	Run 👻

For editor mode, simply refer to it inside using the format: <module\_link name>.<function\_link name> (arguments);

Parameter	Description
module_link_name	Link name of the module in which the function has been defined
function_link name	Name of the function to be invoked
arguments	Parameters of the function, if any

For example, if the module name is *Shopping* and the test case name is *Add to Cart*, in this case will be *shopping.add\_to\_cart()*;



Let's look at an example to understand how functions are used inside test cases. Take the case of the *Add to Cart* function of the website Zylker Watches. The tester created a new function to handle the Add to Cart functionality in an ecommerce web application, in order to implement the function inside the test case Buy a Product.

1. The test case belongs to the Shopping module. The Add to Cart function is called inside the Buy a Product test case. This automatically adds the function's script in the test case, as displayed below:

zw	🐻 Test Cases Manage	Buy a Product *							+ Create Test Case
E5	Search Q 7		⊙ Recor	Not Saved	⊥ We	b Browser	~	ی ا	• Ready ~
Test Cases		General	1 00	counts.login(\$IS_PREVIEW_RUN+""); Calling add to cart function					
Test Suites		Open URL	4 cl	opping. ick(".t Add To Cart(add to cart 2) 4(1	<ol> <li>.zpbutton");</li> </ol>				
reat outlea		Click	5 c1	<pre>click("#zs-address-box-form &gt; div.theme-continue-btn &gt; button");</pre>					
E Test Plans		Set Value	i cl	<pre>it(2); ick("#zs-submit-shipping-method"); it(2);</pre>					
		Assert Text	9 c1	ick("#zs-make-payment-button");					
요. Results		Assert Value	10 wo	<pre>it(10,".font-xs &gt; a",event.SHOW); ick(".font-xs &gt; a"); isk("Forme cond first pare]");</pre>					
	<ul> <li>Guest Shopping</li> </ul>	Wait	13 cl	ick("span:nth-child(2)");					
८ Users		Form Actions	14 cl 15 cl	<pre>ick(ui.payment_status.h); ick(ui.payment_status.h); ick(ui.payment_status.h);</pre>					
63	► 🗅 Review	Select	17 0	<pre>sertText(ui.payment_status.h,"Congratul</pre>	lations! Your order has bee	n confirmed.",error	handling.STOP_ON_ERROR);		
Settings		Deselect	18						
	<ul> <li>B Shopping</li> </ul>	Check							
	.∱ Add To Cart								
	Restalling Watch	Uncheck							
	Bestaening watch	Clear							
	Buy a Product     Grant	Upload							
		Assertion							
		Assert Selected							
		Assert Disabled							
		Assert Visible							
	Request Cancellation	Assert Focused							
		Assert Hidden							
	<ul> <li>Social Media</li> </ul>	Assort Image Eviste							
	<ul> <li>ZylkerWatches</li> </ul>	Assert ImageExists							
		Assert File Download							
		Assert Not Present							
	+ Add Module <=							8	Save 🕞 Run 👻

Understanding the use of functions is crucial. Now that we have learned how to create functions, the next step is to learn how to use them. Choose between <u>recorder</u>, or <u>editor and builder</u> mode to write the functions. Execute and analyze the functions with the preview run options: <u>Web</u>, Android, and iOS, for more reliable and reusable function scripts.

## **Related Links**

• Web Testing using Builder and Editor mode