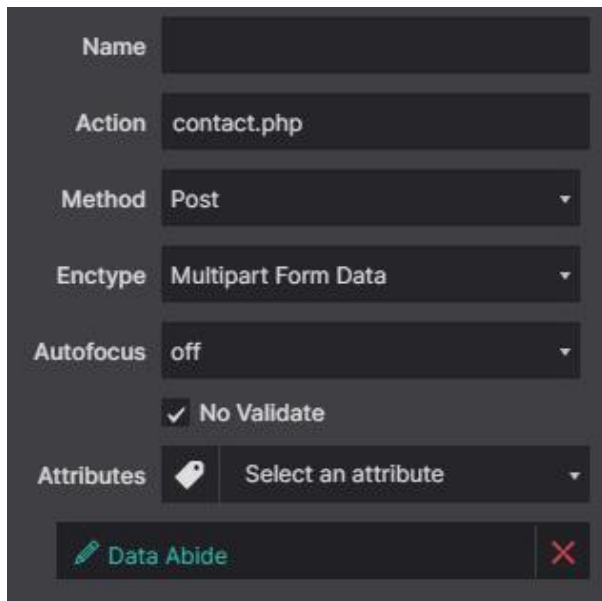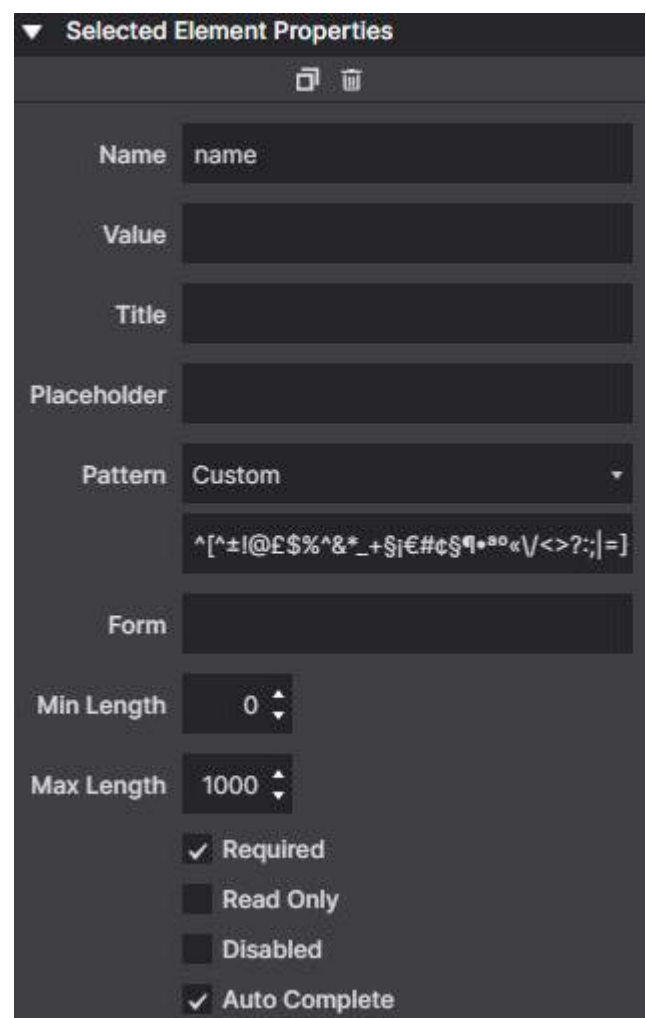# Help for the contact form

Forms can be styled and made very pretty in Site Designer (SD). But there you don't have any back-end scripts; therefore, they don't work. With the php script provided here they do work. When creating a form, you need to be aware of the following, which is the same for all the frameworks, including Bare-Bone ('Frameworkless' in SD), from version 5, build 3360 and later:

1. This is showing the Elements tree pane in SD 5.0.3496 for the element Form Container. Under Action the php file name must appear, and you must use the Post method.

2. When creating the input fields, it is important to give each of them a 'name'. That is because the php script is relying on those names to identify the various fields.

3. Under Pattern you have different options which can be used according to what kind of input field you are creating. There are special ones for Email and URLs. Also for phone numbers, here the 'International' has been used. The options for the fields Title, Name and Subject are a bit too limited, so I have used a Custom string of RegEx', which is excluding all the characters NOT allowed. That means that you can also use non-English characters. The full string will be displayed below (when I don't have to share the page width with the screenshot) It can be edited if you find it necessary.

4. The full regex string: **^[^±!@£$%^&*_+§¡€#¢§¶•ªº«\/<>?:;|=.,]{1,20}$** is excluding comma and full stop, but it may be an idea to remove them from the string. At the end of the string you find {1,20} which means how many characters you can type in a field. If 20 is not enough, you can change it.

5. For the CAPTCHA field I have used the pattern option 'Letters only' since the correct answer in this case is just one word. What you do, will depend on what kind of 'correct answer' you want to have.

The contact.php file.

Earlier PHP files couldn't be added to the resources of a SD project, but now it is possible. Pop it into the 'root' folder of Site Designer, then it will be uploaded to the same folder as the form's 'index.html' and 'thanks.html' files. You need to have it hosted on a different server than the Coffeecup S-drive, though.

The 'contact.php', can be used as good as untouched if you want to create the same kind of form as I have done. But most likely you don't want to ask everybody about Coffeecup related details, so further down are the things you need to observe.

```php
 * contact.php                          x
1    <?php
2    /* Set e-mail recipient */
3    $myemail   = "site-owner@domain.com";
4
5    /* Check all form inputs using check_input function */
6    $title = check_input($_POST['title']);
7    $name = check_input($_POST['name'], "Enter your name");
8    $phone = check_input($_POST['phone']);
9    $email     = check_input($_POST['email'], "Enter your e-mail address");
10   $website  = check_input($_POST['website']);
11   $subject  = check_input($_POST['subject'], "Write a subject");
12   $program   = check_input($_POST['program']);
13   $how = check_input($_POST['how']);
14   $comments = check_input($_POST['comments']);
15   $regex = check_input($_POST['regex']);
16
```

6. On line 3 you need to put in the recipient's mail address.
7. You can see that the names of the different input fields are being used, look at e.g. $phone which has the input name 'phone'. If you wish to have input for other things, e.g. for your car registration number plate, you give the input field the name of, say, 'car' in SD, and then here add a line like

   ```php
   $car    = check_input($_POST['car']);
   ```

```php
29   /* Set the right answer for the regex captcha */
30   if($regex != "seven" && $regex != "Seven" && $regex != "SEVEN")
31   {
32   show_error("You don't seem to be human");
33   }
34
```

8. This validates if the user's answer in the 'human' question is correct. So if you want to set up a different question, you need to pop the correct answers in here.

```
35
36    /* Let's prepare the message for the e-mail
37    $message = "Hello!
38
39    Your contact form has been submitted by:
40    Title: $title
41    Name: $name
42    Phone: $phone
43    E-mail: $email
44    URL: $website
45    Subject: $subject
46    Favourite program? $program
47    How did you find CC? $how
48
49    Comments:
50    $comments
51
52    End of message
53    ";
```

9. This is showing the details for the email message going to the recipient. You can add or subtract from that, e.g. Car: $car

```
54
55    /*  mail headers for from address */
56    $headers = 'From: the-boss@domain.com' . "\r\n" .
57        'Reply-To: the-boss@domain.com' ;
58
59    /* Send the message to the owner using mail() function */
60    mail($myemail, $subject, $message, $headers);
61
62    $subject2 = "Thanks for filling in my contact form";
63    $message2 = "I will get back to you shortly";
64
65    /* Send a message to the form user */
66    mail($email, $subject2, $message2, $headers);
67
68    /* Redirect visitor to the thank you page */
69    header('Location: thanks.html');
70    exit();
```

10. First there are the 'from' addresses, which may, or may not be the same as the address up on line 3, depending on what you want. Then there is a list of parameters that are needed for the mail to the recipient: The sender's Subject and a summary of all the details submitted. And finally, also the details for a confirmation mail going to the sender. The texts inside the double quotes can be edited.

11. At the bottom of this screenshot is the page mentioned that the visitor will be directed to after the form has been submitted. So obviously, such a 'thanks.html' (or 'thanks.php') page has to be created, or you need to set another redirect location.

12. Below there again is some code added for security reasons. That should best be left as it is.

Good luck!