

Page Object Design Pattern

- Represent part of a web page as a class.
- Two parts of the page object:
 - Description of web elements
 - Services provided
- Goals:
 - Separate specification and implementation.
 - Take advantage of OOP features (encapsulation, inheritance, polymorphism).

Steps to create a Page Object

1. Identify the boundaries.
2. Locate the widgets.
3. Define the services.

Page Description

Page Services

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit



PLATFORM
FOR SCIENTIFIC
COLLABORATION



[LOGIN](#) | [REGISTER](#) | [HELP](#)

get started

download

documentation

support

community

hosting

about

HUBzero® is a powerful, open source software platform for creating dynamic web sites that support scientific research and educational activities



Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Widgets

Links

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username: OPTIONAL

Name: REQUIRED

E-mail: REQUIRED

Please answer the question: REQUIRED

Eye, ankle or arm: which is part of the head?

Problem: REQUIRED

Attach a screenshot: OPTIONAL

 Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Widgets

Text Inputs

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Widgets

Text Area

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

 Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Widgets

File Upload

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Widgets

Button



Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Services

Navigation:

- 1.Knowledge Base
- 2.Q&A
- 3.Wishlist
- 4.Support Tickets

Support Ticket:

- 1.Populate Form
- 2.Submit Form

Other:

- 1.Close

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Services

Navigation:

- 1.Knowledge Base
- 2.Q&A
- 3.Wishlist
- 4.Support Tickets

Support Ticket:

- 1.Populate Form
- 2.Submit Form

Other:

- 1.Close



Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Services

Navigation:

- 1.Knowledge Base
- 2.Q&A
- 3.Wishlist
- 4.Support Tickets

Support Ticket:

- 1.Populate Form
- 2.Submit Form

Other:

- 1.Close

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

```
class TroubleReportForm(BasePageWidget):
```

```
    def __init__(self):
```

```
        ...
```

```
    def goto_knowledge_base(self):
```

```
        ...
```

```
    def goto_questions_answers(self):
```

```
        ...
```

```
    def goto_wishlist(self):
```

```
        ...
```

```
    def submit_ticket(self, data):
```

```
        ...
```

```
    def open(self):
```

```
        ...
```

```
    def close(self):
```

```
        ...
```

Page Description

Page Services

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Writing to a Text Input:

1. Find the element
2. Clear the element
3. Send keys to the element

```
<label for="trName">
```

```
  Name:
```

```
  <span class="required">required</span>
```

```
  <input id="trName" type="text" value=""  
    name="reporter[name]">
```

```
</label>
```

```
...
```

```
e = browser.find_element(by=BY.CSS, value="input[id='trName']")
```

```
e.clear()
```

```
e.send_keys("testuser")
```

```
...
```

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Writing to a Text Input:

1. Find the element
2. Clear the element
3. Send keys to the element

```
<label for="trEmail">
```

```
  E-mail:
```

```
  <span class="required">required</span>
```

```
  <input id="trEmail" type="text" value=""  
        name="reporter[email]">
```

```
</label>
```

```
...
```

```
e = browser.find_element(by=BY.CSS, value="input[id='trEmail']")
```

```
e.clear()
```

```
e.send_keys("tu@hubzero.org")
```

```
...
```

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

test problem

Attach a screenshot:

OPTIONAL

Browse...
(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Writing to a Text Input:

1. Find the element
2. Clear the element
3. Send keys to the element

```
<label for="trProblem">
```

Problem:

```
<span class="required">required</span>
```

```
<textarea id="trProblem"
```

```
name="problem[long]"> </textarea>
```

```
</label>
```

...

```
e = browser.find_element(by=BY.CSS, value="textarea[id='trProblem']")
```

```
e.clear()
```

```
e.send_keys("test problem")
```

...

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

test problem

Attach a screenshot:

OPTIONAL

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

Pushing a button:

1. Find the element
2. Call the element's "click()" method

```
<div class="submit">  
  <input id="send-form" type="submit"  
    value="Submit">  
</div>
```

...

```
e = browser.find_element(by=BY.CSS,value="input[id='send-form']")  
e.click()
```

...

...

File a Support Ticket

Populate the Name field

```
e = browser.find_element(by=BY.CSS, value="input[id='trName']")  
e.clear()  
e.send_keys("testuser")
```

Populate the Email field

```
e = browser.find_element(by=BY.CSS, value="input[id='trEmail']")  
e.clear()  
e.send_keys("tu@hubzero.org")
```

Populate the Problem field

```
e = browser.find_element(by=BY.CSS, value="textarea[id='trProblem']")  
e.clear()  
e.send_keys("test problem")
```

Click the Submit button

```
e = browser.find_element(by=BY.CSS, value="input[id='send-form']")  
e.click()
```

...

...

File a Support Ticket

Populate the Name field

```
e = browser.find_element(by=BY.CSS, value="input[id='trName']")  
e.clear()  
e.send_keys("testuser")
```

Text Input

Populate the Email field

```
e = browser.find_element(by=BY.CSS, value="input[id='trEmail']")  
e.clear()  
e.send_keys("tu@hubzero.org")
```

Text Input

Populate the Problem field

```
e = browser.find_element(by=BY.CSS, value="textarea[id='trProblem']")  
e.clear()  
e.send_keys("test problem")
```

Text Input

Click the Submit button

```
e = browser.find_element(by=BY.CSS, value="input[id='send-form']")  
e.click()
```

Button

...

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

test problem

Attach a screenshot:

OPTIONAL

Browse...
(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

```
class Text(BasePageWidget):
    def __init__(self, locator):
        ...
    def value(self, text):
        e = self.find_element(self.locator)
        e.clear()
        e.send_keys(text)
```

```
<label for="trName">
    Name:
    <span class="required">required</span>
    <input id="trName" type="text" value=""
        name="reporter[name]">
</label>
```

```
...
e = browser.find_element( by=BY.CSS,
    value="input[id='trName']")
e.clear()
e.send_keys("testuser")
...
```

```
...
name = Text("input[id='trName']")
name.value('testuser')
...
```

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

test problem

Attach a screenshot:

OPTIONAL

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

```
class Button(BasePageWidget):
    def __init__(self, locator):
        ...
    def click(self):
        e = self.find_element(self.locator)
        e.click()
```

```
<div class="submit">
    <input id="send-form" type="submit"
        value="Submit">
</div>
```

```
...
e = browser.find_element( by=BY.CSS,
    value="input[id='send-form']")
e.clear()
e.send_keys("testuser")
...
```

```
...
name = Text("input[id='send-form']")
name.value('testuser')
...
```

...

File a Support Ticket

Populate the Name field

```
name = Text("input[id='trName']")  
name.value("testuser")
```

Text Input

Populate the Email field

```
email = Text("input[id='trEmail']")  
email.value("tu@hubzero.org")
```

Text Input

Populate the Problem field

```
problem = Text("textarea[id='trProblem']")  
problem.value("test problem")
```

Text Input

Click the Submit button

```
submit = Button("input[id='send-form']")  
submit.click()
```

Button

...

...

File a Support Ticket

Populate the Name field

```
name = Text("input[id='trName']")  
name.value("testuser")
```

Text Input

Populate the Email field

```
email = Text("input[id='trEmail']")  
email.value("tu@hubzero.org")
```

Text Input

Populate the Problem field

```
problem = TextArea("textarea[id='trProblem']")  
problem.value("test problem")
```

Text Area

Click the Submit button

```
submit = Button("input[id='send-form']")  
submit.click()
```

Button

...

```
class TroubleReportForm(BasePageWidget):
    def __init__(self):
        ...
        self.username = Text(self, '#trUsername')
        self.name      = Text(self, '#trName')
        self.email      = Text(self, '#trEmail')
        self.problem    = TextArea(self, '#trProblem')
        self.upload     = Upload(self, '#trUpload')
        self.submit     = Button(self, '#send-form')
        ...
```

```
def submit_ticket(self, data):
    self.name.value(data['name'])
    self.email.value(data['email'])
    self.problem.value(data['problem'])
    ...
    self.submit.click()
```

```
def open(self):
    ...
```

```
def close(self):
    ...
```

Page Description

Page Services



Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

```
class TroubleReportForm(BasePageWidget):
```

```
    def __init__(self):
```

```
        ...
```

```
        self.username = Text(self, 'username')
```

```
        self.captcha   = Captcha(self, 'captcha')
```

```
        self.problem   = TextArea(self, 'problem')
```

```
        self.upload     = Upload(self, 'upload')
```

```
        self.submit     = Button(self, 'submit')
```

```
        ...
```

```
    def submit_ticket(self, data):
```

```
        ...
```

```
    def open(self):
```

```
        ...
```

```
    def close(self):
```

```
        ...
```

```
# open and close the need help pane
```

```
po = TroubleReportForm()
```

```
po.open()
```

```
po.close()
```


Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

```
class TroubleReportForm(BasePageWidget):
```

```
    def __init__(self):
```

```
        ...
```

```
        self.username = Text(self, 'username')
```

```
        self.captcha   = Captcha(self, 'captcha')
```

```
        self.problem   = TextArea(self, 'problem')
```

```
        self.upload     = Upload(self, 'upload')
```

```
        self.submit     = Button(self, 'submit')
```

```
        ...
```

```
    def submit_ticket(self, data):
```

```
        ...
```

```
    def open(self):
```

```
        ...
```

```
    def close(self):
```

```
        ...
```

```
# submit a blank ticket
```

```
po = TroubleReportForm()
```

```
po.open()
```

```
data = {}
```

```
po.submit_ticket(data)
```

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

test problem

Attach a screenshot:

OPTIONAL

myscreenshot.png

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

```
class TroubleReportForm(BasePageWidget):
```

```
    def __init__(self):
```

```
        ...
```

```
        self.username = Text(self, 'username')
```

```
        self.captcha   = Captcha(self, 'captcha')
```

```
        self.problem   = TextArea(self, 'problem')
```

```
        self.upload     = Upload(self, 'upload')
```

```
        self.submit     = Button(self, 'submit')
```

```
        ...
```

```
    def submit_ticket(self, data):
```

```
        ...
```

```
    def open(self):
```

```
        ...
```

```
    def close(self):
```

```
        ...
```

*# fill in the name, email, problem,
upload a file, then submit the ticket*

```
po = TroubleReportForm()
```

```
po.open()
```

```
data = {
```

```
    'name'      : 'testuser',
```

```
    'email'     : 'tu@hubzero.org',
```

```
    'problem'   : 'test problem',
```

```
    'upload'    : 'myscreenshot.png',
```

```
}
```

```
po.submit_ticket(data)
```

Support Options



Knowledge Base

Find information on common issues.



Ask the Community

Ask questions and find answers from other users.



Wish List

Suggest a new site feature or improvement.



Support Tickets

Check on status of your tickets.

Submit a Support Ticket

Username:

OPTIONAL

Name:

REQUIRED

E-mail:

REQUIRED

Please answer the question:

REQUIRED

Eye, ankle or arm: which is part of the head?

Problem:

REQUIRED

Attach a screenshot:

OPTIONAL

Browse...

(.jpg, .jpeg, .jpe, .bmp, .tif, .tiff, .png, .gif)

Submit

```
class TroubleReportForm(BasePageWidget):
```

```
    def __init__(self):
```

```
        ...
```

```
        self.username = Text(self, 'username')
```

```
        self.captcha   = Captcha(self, 'captcha')
```

```
        self.problem   = TextArea(self, 'problem')
```

```
        self.upload     = Upload(self, 'upload')
```

```
        self.submit     = Button(self, 'submit')
```

```
        ...
```

```
    def submit_ticket(self, data):
```

```
        ...
```

```
    def open(self):
```

```
        ...
```

```
    def close(self):
```

```
        ...
```

```
# fill in the name, email, problem,  
# upload a file, then submit the ticket
```

```
po = TroubleReportForm()  
po.open()
```

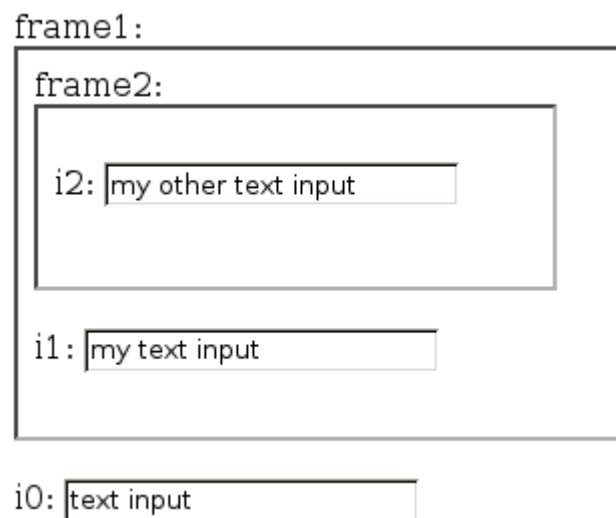
```
po.name.value('testuser')  
po.email.value('tu@hubzero.org')  
po.problem.value('test problem')  
po.upload.value('myscreenshot.png')
```

```
po.submit.click()
```

Interacting with Iframes

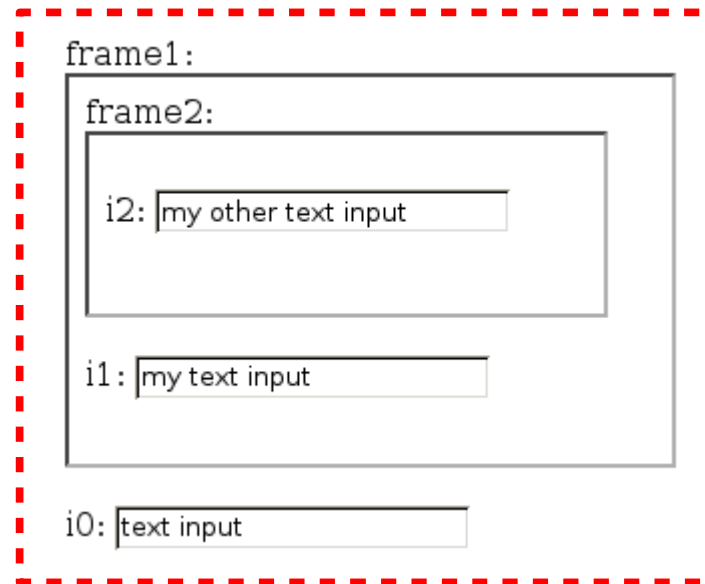
Interacting with Iframes

How do Iframes work



How do Iframes work

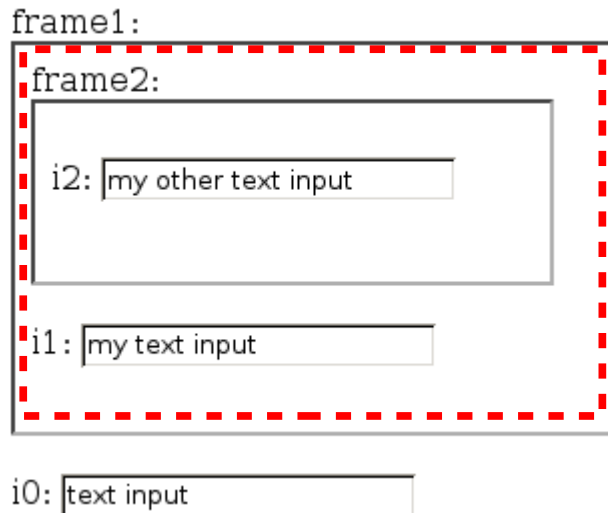
Default Context →



```
<html>
  <body>
    <label for="frame1">frame1: </label>
    <iframe id="frame1" src="inner_page.html"></iframe>
    <br/><br/>
    <label for="i0">i0: </label>
    <input type="text" id="i0" value="text input"></input>
  </body>
</html>
```

How do Iframes work

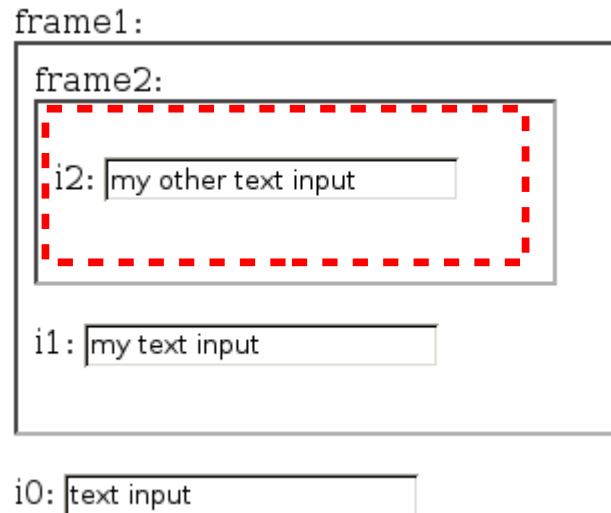
Frame1 Context →



```
<html>
  <body>
    <label for="frame2">frame2: </label>
    <iframe id="frame2" src="another_page.html"></iframe>
    <br/><br/>
    <label for="i1">i1: </label>
    <input type="text" id="i1" value="my text input"></input>
  </body>
</html>
```


How do Iframes work

Frame2 Context →



```
<html>
  <body>
    <label for="i2">i2: </label>
    <input type="text" id="i2" value="my other text input"></input>
  </body>
</html>
```

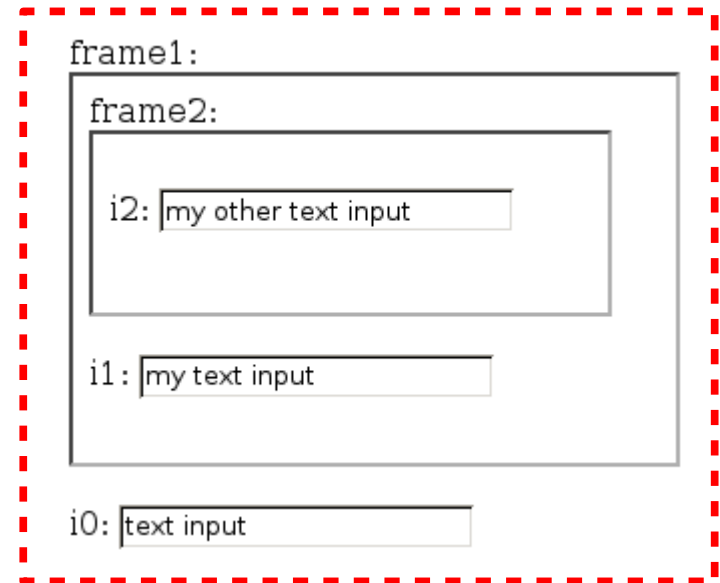
Page Object for i0

```
class Text(BasePageWidget):
    def __init__(self, locator):
        ...

    # getter
    def value(self):
        e = self.find_element(self.locator)
        return e.get_attribute('value')

    # setter
    def value(self, text):
        e = self.find_element(self.locator)
        e.clear()
        e.send_keys(text)

    def append(self, text):
        e = self.find_element(self.locator)
        e.send_keys(text)
```



Default Context

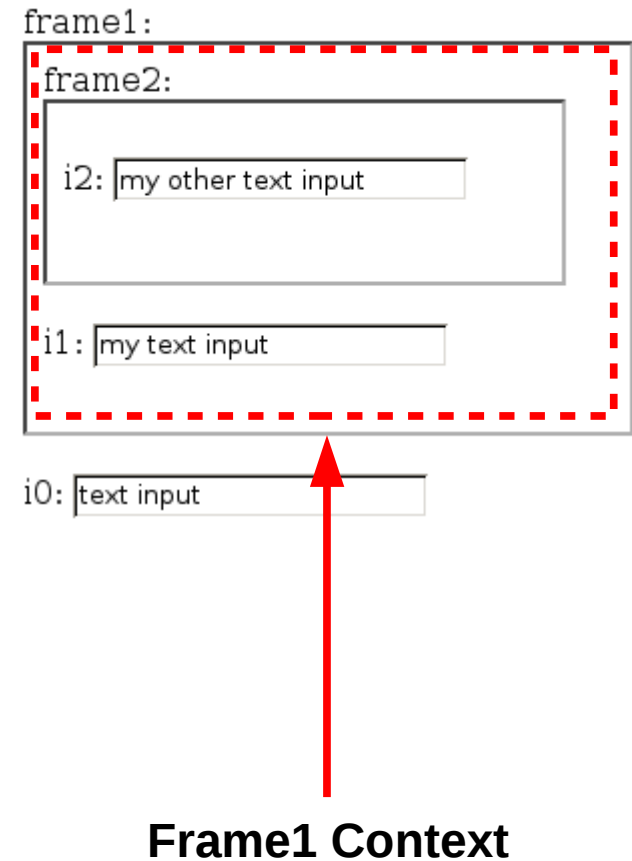
Page Object for i1

```
class Text1Frame(BasePageWidget):
    def __init__(self, locator):
        ...

    # getter
    def value(self):
        ...

    # setter
    def value(self, text):
        frame = self.find_element('#frame1')
        self._browser.switch_to_frame(frame)
        e = self.find_element(self.locator)
        e.clear()
        e.send_keys(text)
        self._browser.switch_to_default_content()

    def append(self, text):
        ...
```



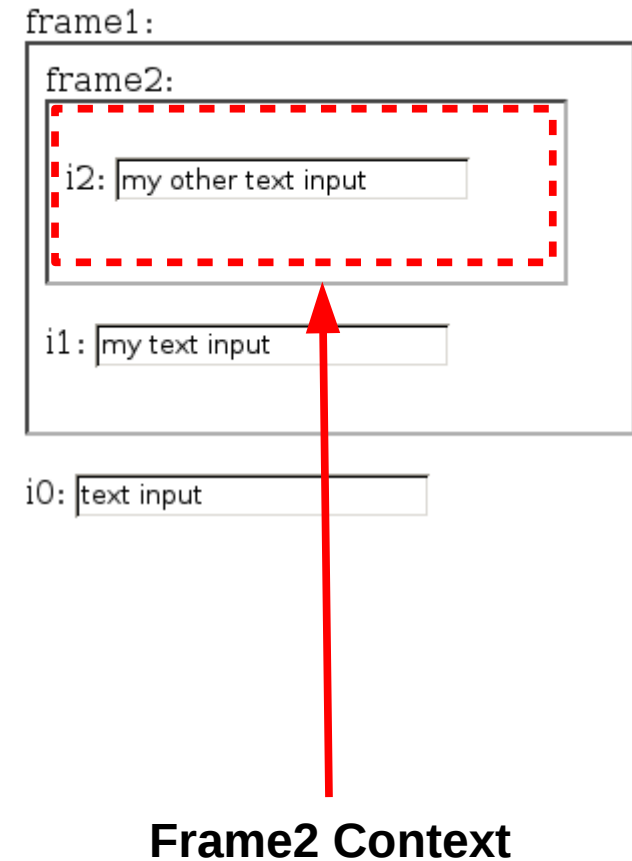
Page Object for i2

```
class Text2Frame(BasePageWidget):
    def __init__(self, locator):
        ...

    # getter
    def value(self):
        ...

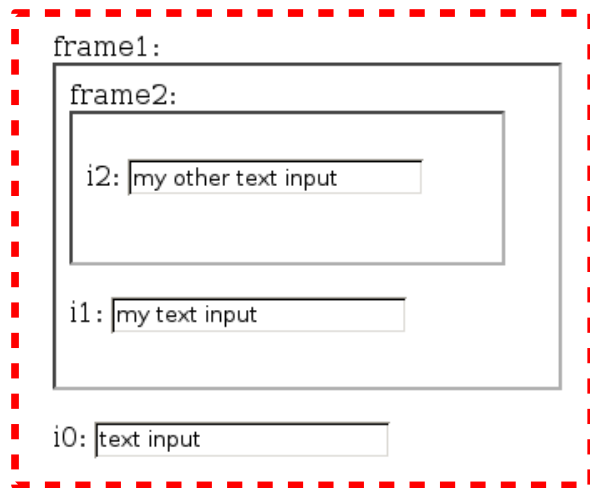
    # setter
    def value(self, text):
        frame1 = self.find_element('#frame1')
        self._browser.switch_to_frame(frame1)
        frame2 = self.find_element('#frame2')
        self._browser.switch_to_frame(frame2)
        e = self.find_element(self.locator)
        e.clear()
        e.send_keys(text)
        self._browser.switch_to_default_content()

    def append(self, text):
        ...
```

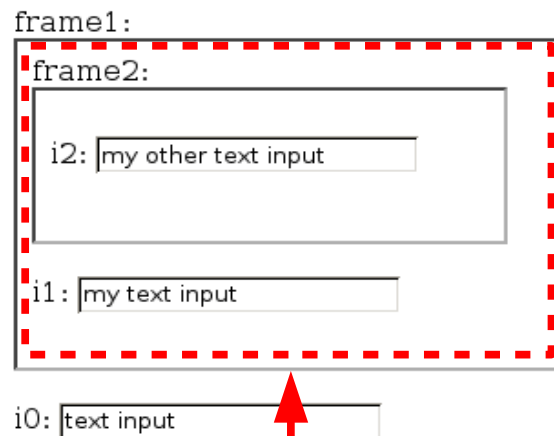


Dealing with Iframes

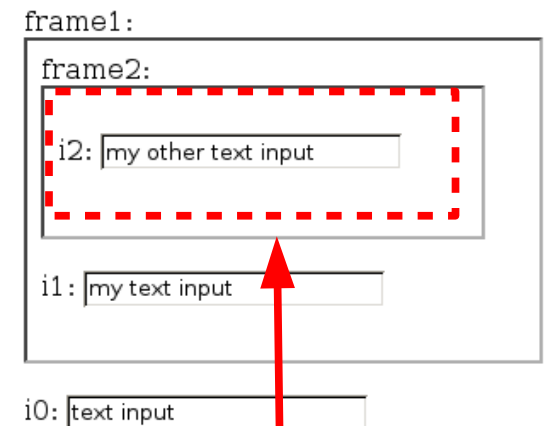
- Enter / Exit frames to interact with elements
- Must update all methods of page object.
- Create new page objects for Iframes?!?



Default Context



Frame1 Context



Frame2 Context

IframeWrap Design Pattern

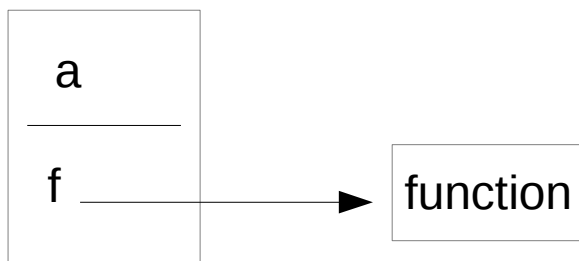
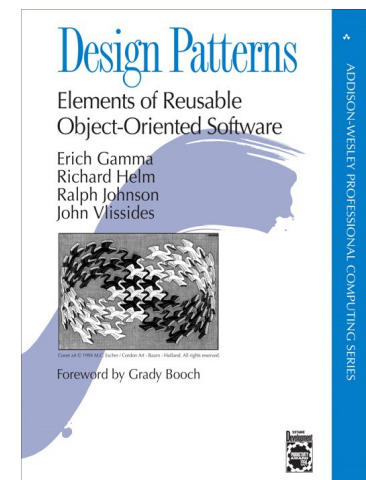
Use Decorator Pattern to wrap attributes of a page object with Enter / Exit Iframe calls.

Decorator Pattern

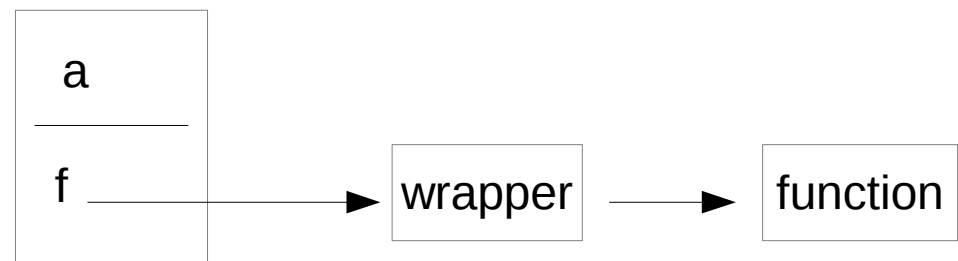
Attach additional responsibilities to an object dynamically and transparently.

```
class A(object):  
    def f(self):  
        do_some_stuff()
```

```
a = A()
```



Before Decoration



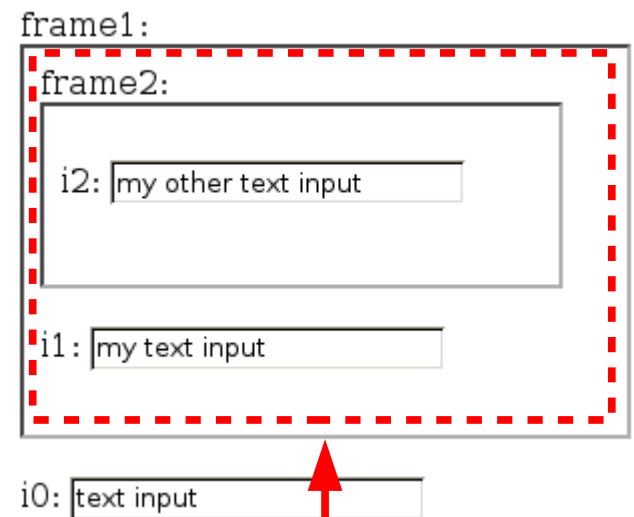
After Decoration

IframeWrap Design Pattern

Use Decorator Pattern to wrap attributes of a page object with Enter / Exit Iframe calls.

```
class Text(BasePageWidget):  
  
    # setter  
    def value(self, text):  
        e = self.find_element(self.locator)  
        e.clear()  
        e.send_keys(text)
```

```
i1 = Text('#i1')  
i1.value('new i1 text')
```

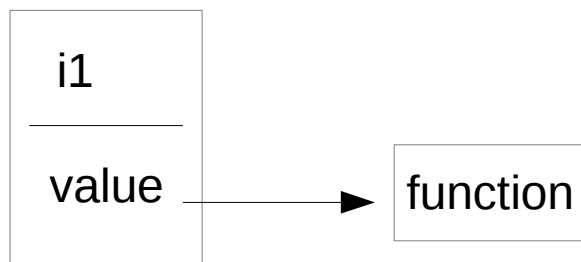
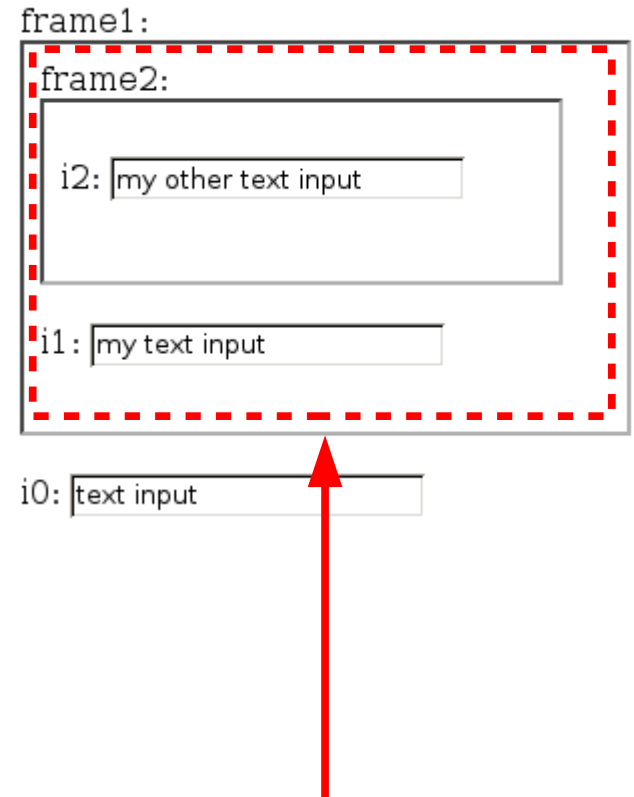


Frame1 Context

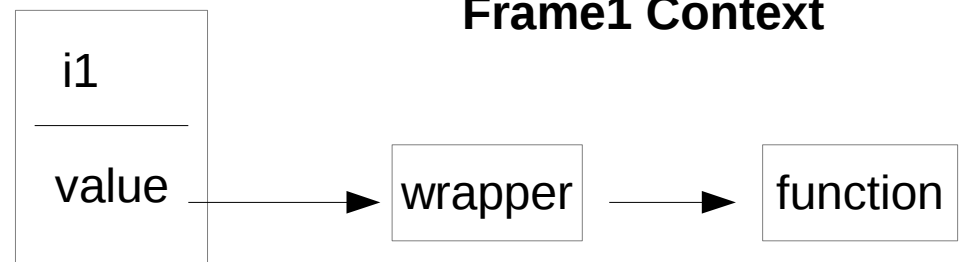
IframeWrap Design Pattern

```
class Text(BasePageWidget):  
  
    # setter  
    def value(self, text):  
        e = self.find_element(self.locator)  
        e.clear()  
        e.send_keys(text)
```

```
i1 = Text('#i1')  
i1.value('new i1 text')
```



Before Decoration

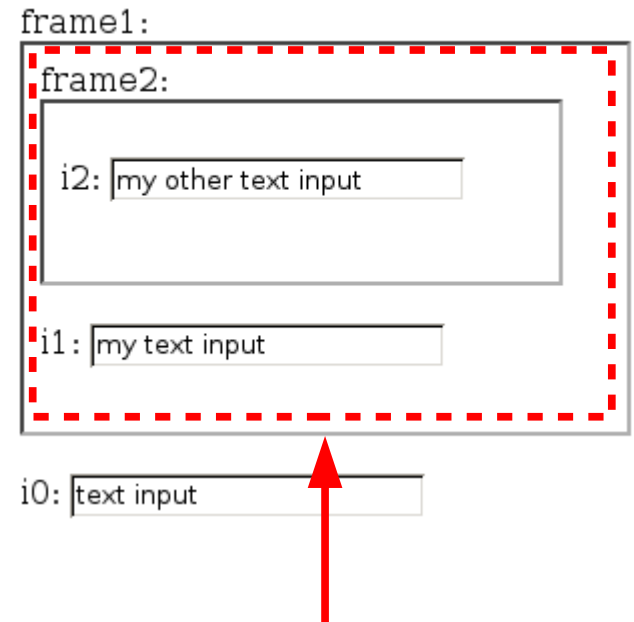


After Decoration

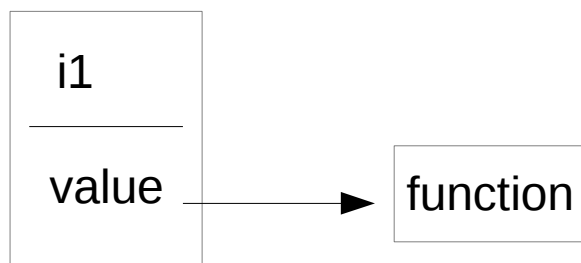
IframeWrap Design Pattern

```
def wrapper_method(self, f):  
    def wrapper(*args, **kwargs):  
        frame1 = self.find_element('#frame1')  
        self._browser.switch_to_frame(frame1)  
        rv = f(*args, **kwargs)  
        self._browser.switch_to_default_content()  
        return rv  
    return wrapper
```

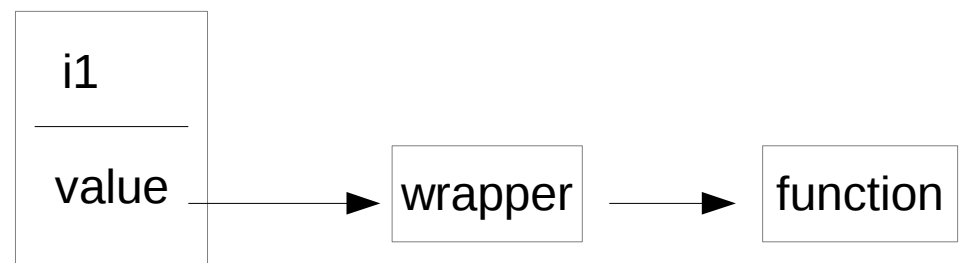
```
i1 = IframeWrap(Text('#i1'), ['#frame1'])  
i1.value('new i1 text')
```



Frame1 Context



Before Decoration



After Decoration

Page Objects for i0, i1, and i2

```
class FramedInputs(BasePageObject):  
    def __init__(self):  
  
        self.i0 = Text('#i0')  
  
        self.i1 = lframeWrap( Text('#i1'), ['#frame1'] )  
  
        self.i2 = lframeWrap( Text('#i2'), ['#frame2', '#frame1'] )
```

