U5WNSLTT4 : sorry, ok :slightly_smiling_face:
U1NSCAY6R : One thing you're doing is changing the value of `post.image` each iteration of that loop
U1NSCAY6R : Is it `pass`ing each loop? Do any of those keys exist in `request.files`?
U1NSCAY6R : <@U5WNSLTT4> ^
U28MDQRL2 : you can add those file fields on your form <@U5WNSLTT4>. Then you can pas the `request.FILES` to the form constructor and it handles that for you.
U3G7RJP61 : Anyone know if this is the right shorthand for a one liner? `featured_image=(n == selectedImageIndex)`
U3G7RJP61 : n is from `enumerate` and would be `0` `1` `2`, etc. `selectedImageIndex` would be `1` in this case.
U3G7RJP61 : Python doesn't moan but it doesn't seem to be working
U5JG72GA2 : Is this the right channel to post novice Python questions?
U3G7RJP61 : Sure
U5JG72GA2 : Ok, cool. I'm trying to learn more about `super()`

Here's a code snippet I've written:
```
class MyList(list):
    def __len__(self):
        print("calculating the total number of items in the list")
        super(MyList, self).__len__()
```
And when I instantiate `MyList` I get the following error:
`TypeError: an integer is required`

what am I doing wrong?

U5NMSURAQ : - `super(MyList, self).__len__()`
U5NMSURAQ : + `return super(MyList, self).__len__()`
U5NMSURAQ : actually, what Python version are you targeting?
U5NMSURAQ : `return super().__len__()` is much nicer :slightly_smiling_face:
U3G7RJP61 : Instantiating in Python 3 with that code would be fine I think? But trying to do `x = MyList()` `x.len()` would return `AttributeError: 'MyList' object has no attribute 'len'`
U5JG72GA2 : I'm using Python 2.7.x
U5NMSURAQ : ... come to think of it, if you don't do any processing at all, you can just skip this method
U5NMSURAQ : if you don't implement it yourself, an original one will be used from `list`
U5JG72GA2 : well I'm just practicing with `super`. Let's say I want to log or print before `__len()__` is executed
U5JG72GA2 : ok why do I have to use `return`?
U3G7RJP61 : You're saying return the result of my parent classes `len` method.
U5JG72GA2 : ok so I did this as well but no return was necessary:
```
class MyDict(dict):
    def __setitem__(self, key, value):
        print("adding key: {} and value: {}".format(key, value))
        super(MyDict, self).__setitem__(key, value)

    def __getattr__(self, key):
        if key not in self:
            return 'key not recognized'
        else:
            return self[key]

```

U5NMSURAQ : `setitem` doesn't return anything useful
U5NMSURAQ : and there is a return statement in your `getattr` :))
U5JG72GA2 : ah I see
U0L051JUB : What is the best way to convert a list to a dict? The only thing is I want to specify my keys from outside the list. So the list is basically the values of the dict and I will provide the keys separately.
U5NMSURAQ :
```
&gt;&gt;&gt; values = ['one', 'two', 'three']
&gt;&gt;&gt; keys = [1, 2, 3]
&gt;&gt;&gt; d = dict(zip(keys, values))
&gt;&gt;&gt; d
```

{1: 'one', 2: 'two', 3: 'three'}
```

U0L051JUB : Perfect exactly what I was looking for.
U5WD1KS5P : Hey guys, I'm new to deploying django applications and I've been fighting an issue for two days now, could anyone help me with this? <https://stackoverflow.com/questions/44666555/django-wsgi-no-module-named-site> I just posted a question in stack overflow and I'm completely stuck at the moment
U1BP42MRS : <@U5WD1KS5P> I think you have a type-o in your question:
```
activate_this = os.path.join(PROJECT_DIR, 'env/shinra/bin', 'activate_this.py'$
```

The `$` is invalid and would cause a syntax error

U5WD1KS5P : oh yeah let me edit that, thanks!
```