



Creating a Try Except finally Block

Now we're going to see how to create a **try-except-finally block** in Python, and this is really great for when you're wanting to catch different kinds of errors.

```
try:
    x = 5 + 'ham'
```

So, basically I'm trying to add an integer and a string which we know isn't really going to work.

And then I'll create my except which is actually going to catch the exception:

```
try:
    x = 5 + 'ham'
except Exception as e:
    print(type(e))
    print(e)
    print("\n")
```

I'm catching this exception into parameter `e`. If the error happens, the block under the `except` will be run.

Division Error

All right, so now we're going to go into a different kind of handling:

```
try:
    x = 1/0
except Exception as e:
    print(e)
    print("\nnew line")
except ZeroDivisionError:
    print('ZeroDivisionError')
```

`ZeroDivisionError` is a type of the error that you'll get when trying to divide by zero.

Adding the 'finally' Now

You can also add the `finally` block:

```
try:
    x = 1/0
except Exception as e:
    print(e)
    print("\nnew line")
except ZeroDivisionError:
    print('ZeroDivisionError')
finally:
    print('finally hit')
```

This block is going to run regardless: it does not matter whether the error was raised or not.

Creating A 'try'

Create another exception by referencing a variable that does not exist:

```
try:
    x = 4 + test
except Exception as e:
    print(type(e))
    print(e)
    print("\n x = 4 + test")
except NameError as i:
    print('name error')
    print(i)
except:
    print('not a name error')
```