

# Ayudantía 7

**Exceptions + Testing**

# ¿Qué veremos?

Exceptions  
Testing

- Manejo de excepciones
- Lo que necesitan saber de testing
- Solución **rápida** de la AC de Metaclases

— — —

# Exceptions

# Tipos de excepciones

- `SyntaxError` exception
- `NameError` exception
- `ZeroDivisionError` exception:
- `IndexError` exception
- `TypeError` exception
- `AttributeError` exception
- `KeyError` exception
- `Exception`

# Manejo de excepciones

# Manejo de excepciones

---

try:

# código

**Aquí va nuestro código a probar**

except:

# Error

**Aquí el error que esperamos**

# Manejo de excepciones

---

```
def dividir(num,den):  
    try:  
        return float(num)/float(den)  
    except (ZeroDivisionError) as err:  
        print('Error: {}'.format(err))  
        print('El denominador debe ser distinto de 0')
```

# Generando excepciones

(código)



# Testing

# Unittest

# Unittest

## ¿Qué es y cómo funciona?

(código)

# Tipos de assert

- `self.assertEqual()`
- `self.assertNotEqual()`
- `self.assertTrue()`
- `self.assertFalse()`
- `self.assertIs()`
- `self.assertIsNot()`
- `self.assertIsNone()`
- `self.assertIsNotNone()`
- `self.assertIn()`
- `self.assertNotIn()`
- `self.assertIsInstance()`
- `self.assertNotIsInstanc`

— — —

# Métodos: setUp() y tearDown()

(código)

# ¿Cómo ignorar test que sabemos que fallaran?

- `expectedFailure`
- `skip(reason)`
- `skipIf(condition, reason)`
- `skipUnless(condition, reason)`

**Muy lindos los test Peter... pero  
¿cómo testeo mi programa?**  
(más código)

# AC07: Metaclasses