

How to read an error message

```
Traceback (most recent call last):
  File "test.py", line 25, in ?
    triangle()
  File "test.py", line 12, in triangle
    inc_total_height()
  File "test.py", line 8, in inc_total_height
    total_height = total_height + height
UnboundLocalError: local variable 'total_height' referenced before assignment
```

```
Traceback (most recent call last):
  File "test.py", line 2, in nr
    nr = int('23.5')
ValueError: invalid literal for int() with base 10: '23.5'
```

```
Traceback (most recent call last):
  File "test.py", line 7, in
    main()
  File "test.py", line 5, in main
    print hello
NameError: global name 'hello' is not defined
```

Debugging tips:

- don't panic
- read from the bottom
- learn the rules (= google the error)
- use **print** statements in places where variables are created or changed

ErrorName	Example	Description (non-exhaustive)
Syntax error	1) flowers = ['rose','lily','gerbera' 2) for flower in flowers print flower	Forgetting the '.' at the end of def if/elif/else when for. Forgetting to close quotes and parentheses.
ValueError	int('rose')	Assigning an innappropriate value
Attribute Error	a = 3 a.replace(3,1)	Calling an inappropriate method
TypeError	3 + 'rose'	An operation or function is applied to an object of inappropriate type
NameError	flowers = ['rose','lily','gerbera'] print fowlers	A particular variable name is not found.
KeyError	things = {'flower': 'rose', 'bodypart': 'nose'} things['car']	A dictionary key is not found in the set of existing keys
IndexError	flowers = ['rose', 'lily', 'gerbera'] print flowers[30]	When a sequence subscript is out of range.
Indentation Error	for flower in flowers: print flower	Incorrect indentation.
Semantic Error		Will not generate any error messages, but the script will not do what you wanted it to do.