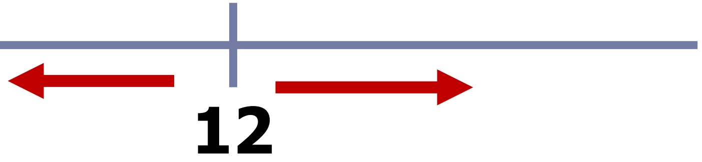
What is Testing?

- An **activity** in which a system or a component is executed under specific conditions, the **results** are observed or recorded, and an evaluation is made of some aspect of the system or component.
- Testing must be done in an **organised**, **systematic**, **manner**, following a process!
- "Program testing can be used to show the **presence of bugs**, but **never** to show their **absence**." (Dijkstra, 1969)
- Proofs of **correctness** are often as hard (if not harder) to get right than writing the code
- **Testing** vs **Debugging**:
  - Testing is (trying) to produce failures
  - Debugging is (trying to) find the fault that caused failure

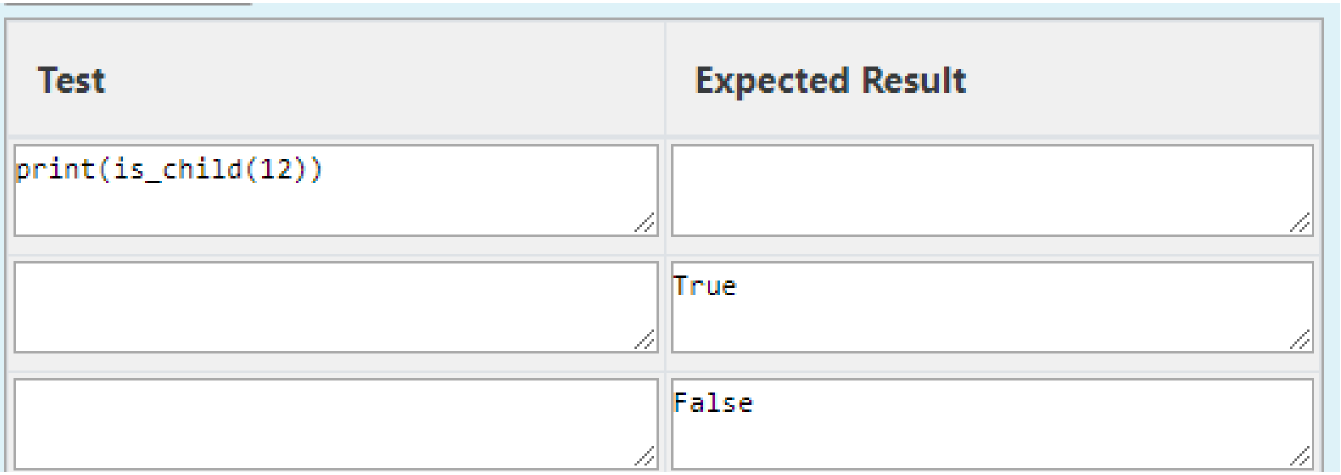
What is Testing? (continued)

- What kinds of "testing" do you do:
  - Compile the program
  - Enter example values to test your program
  - Desk-check or code walk through
  - Ask your friend to test your program

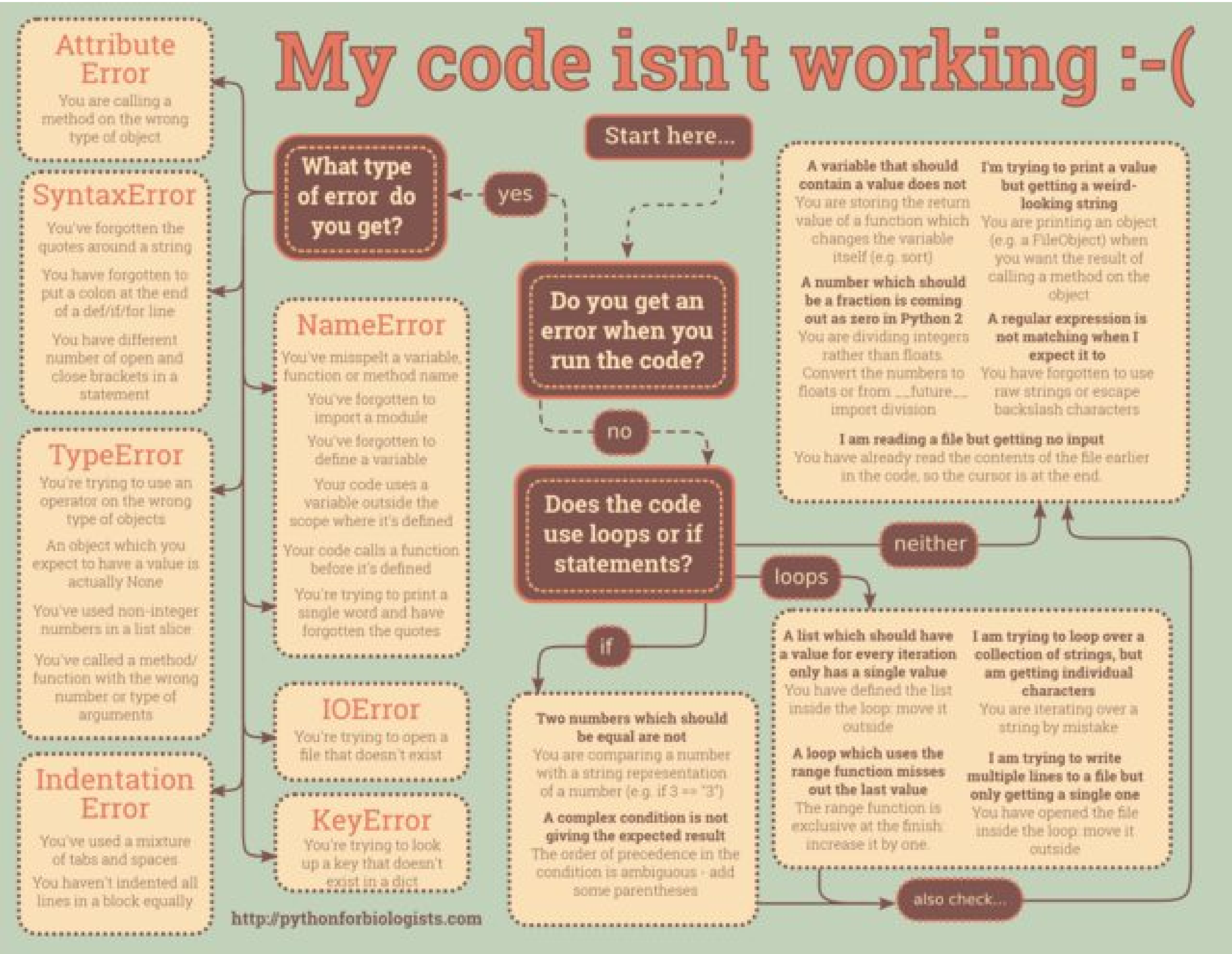
Demo: Q1



- Write a boolean-valued function is_child(age) which returns True if the given age value is 12 or less. Otherwise the function should return False. Note: you can assume that the parameter *age* is a positive integer."
- What are the sort of test cases you could put in the answer box?
- What sort of things do you need to consider?



Test	Expected Result
print(is_child(12))	
	True
	False



My code isn't working :-(

Start here...

Do you get an error when you run the code?

yes

What type of error do you get?

Attribute Error
You are calling a method on the wrong type of object

SyntaxError
You've forgotten the quotes around a string
You have forgotten to put a colon at the end of a def/if/for line
You have different number of open and close brackets in a statement

TypeError
You're trying to use an operator on the wrong type of objects
An object which you expect to have a value is actually None
You've used non-integer numbers in a list slice
You've called a method/function with the wrong number or type of arguments

Indentation Error
You've used a mixture of tabs and spaces
You haven't indented all lines in a block equally

NameError
You've misspelt a variable, function or method name
You've forgotten to import a module
You've forgotten to define a variable
Your code uses a variable outside the scope where it's defined
Your code calls a function before it's defined
You're trying to print a single word and have forgotten the quotes

IOError
You're trying to open a file that doesn't exist

KeyError
You're trying to look up a key that doesn't exist in a dict

no

Does the code use loops or if statements?

neither

A variable that should contain a value does not
You are storing the return value of a function which changes the variable itself (e.g. sort)

A number which should be a fraction is coming out as zero in Python 2
You are dividing integers rather than floats
Convert the numbers to floats or from __future__ import division

I'm trying to print a value but getting a weird-looking string
You are printing an object (e.g. a FileObject) when you want the result of calling a method on the object

A regular expression is not matching when I expect it to
You have forgotten to use raw strings or escape backslash characters

I am reading a file but getting no input
You have already read the contents of the file earlier in the code, so the cursor is at the end.

loops

A list which should have a value for every iteration only has a single value
You have defined the list inside the loop: move it outside

A loop which uses the range function misses out the last value
The range function is exclusive at the finish: increase it by one

I am trying to loop over a collection of strings, but am getting individual characters
You are iterating over a string by mistake

I am trying to write multiple lines to a file but only getting a single one
You have opened the file inside the loop: move it outside

if

Two numbers which should be equal are not
You are comparing a number with a string representation of a number (e.g. if 3 == '3')

A complex condition is not giving the expected result
The order of precedence in the condition is ambiguous - add some parentheses

also check...

http://pythonforbiologists.com

The 10 Most Common Mistakes That Python Developers Make

- Modifying a list while iterating over it

```
def multiply_numbers(numbers, factor):
    for value in numbers:
        value *= factor
```

- We cannot return a value after only looking at **one item**

```
def get_min(numbers):
    min_value = 0
    for value in numbers:
        if value < min_value:
            return value
        else:
            return min_value
```

- Are we merging two tuples together?

```
def tuple_append(tuple1, number):
    return tuple1[:2] + number
```

Default Argument:

- Misusing expressions as defaults for function arguments

```
    def append_to(element, original=[]):
        original.append(element)
        return original
```

```
my_list = [1, 2, 3]
print(append_to(100, my_list))
```

```
[1, 2, 3, 100]
```

✓

```
my_list = append_to(12)
print(my_list)

my_other_list = append_to(42)
print(my_other_list)
```

```
[12]
[12, 24]
```

✗

A new list is created **once** when the function is defined, and the same list is used in each successive call.

- https://www.toptal.com/python/top-10-mistakes-that-python-programmers-make

Coderunner – Tips

- Q1: It can be useful testing the value "boundaries" – what are the min and max valid values? The values that are just outside the valid range? It is also sometimes helpful to check some value in the middle of the range
- Q2: When testing strings try to think of unique cases – for example strings with punctuation or spaces (".", ",", " ", etc), strings with upper and lowercase letters, empty strings, strings with a mix of vowels and consonants, etc
- Q4: When testing lists there are several unique cases including empty lists, lists with one item, lists with even numbers of items, lists with odd numbers of items.
- Q5 Again think of unique cases. For example, empty lists, lists with one element, lists with just odd numbers, lists with even numbers, lists with a mix, lists with repeated numbers

Coderunner – Tips

- Q8: It might be helpful to make a tuple containing one item for this question. In python this can be done with the following notation:
  (my_element,)
  (note the comma at the end of the element in the tuple)
- Q9: Default values in function declarations are created once during the program (i.e. a new one is not created each function call) – you might have to think of another way to use default values to check if the function is passed a list or not.
- Q10: It might be useful to look up the difference between the .sort() and sorted(…) functions.
- Q11: Boss question – lots of small errors in this one. Try to understand what the code is trying to do and where some errors might be before making any changes