UNIVERSITY OF CAMBRIDGE

# Here’s Your Mistake. . .

## Taking a closer look at students’ mistakes

Dr. Tobias Kohn

## What is a “syntax error”?

## Where do syntax errors come from?

## What can we do about syntax errors?

## I. UNDERSTANDING THE PROBLEM

## Counting Letters

Write a program that counts how often the letter *e* occurs in a string.

## Counting Letters

Write a program that counts how often the letter *e* occurs in a string.

```
count = 0
for c in my_text:
    if c = 'e':
        count += 1

print(count)
```

## Counting Letters

Write a program that counts how often the letter *e* occurs in a string.

```
count = 0
for c in my_text:
    if c = 'e':
        count += 1

print(count)
```

**SyntaxError: invalid syntax**

UNIVERSITY OF CAMBRIDGE

## Counting Letters

Write a program that counts how often the letter *e* occurs in a string.

```
count = 0
for 'e' in my_text:
    count += 1

print(count)
```

## Counting Letters

Write a program that counts how often the letter *e* occurs in a string.

```
count = 0
for 'e' in my_text:
    count += 1

print(count)
```

**SyntaxError: can't assign to literal**

UNIVERSITY OF CAMBRIDGE

## Counting Letters

**“Minor” mistake**

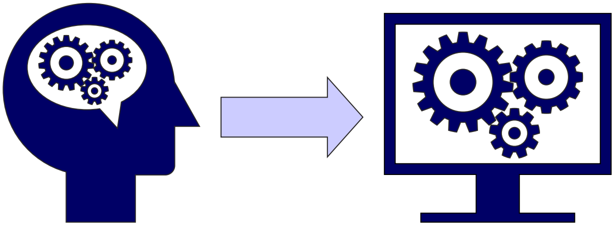
```
count = 0
for c in my_text:
    if c = 'e':
        count += 1

print(count)
```

**Misconception**

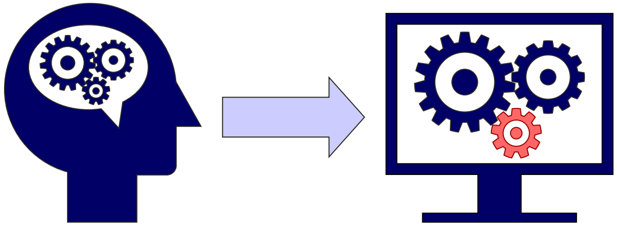
```
count = 0
for 'e' in my_text:
    count += 1

print(count)
```

UNIVERSITY OF CAMBRIDGE

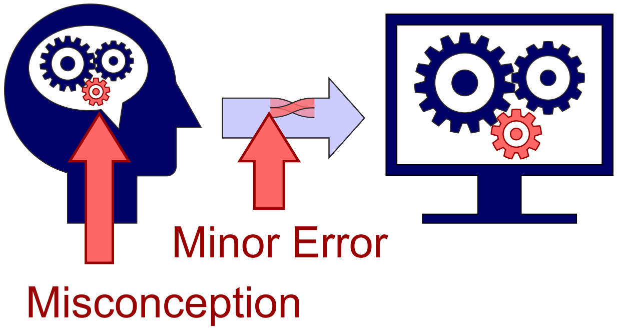
## Minor mistakes and misconceptions



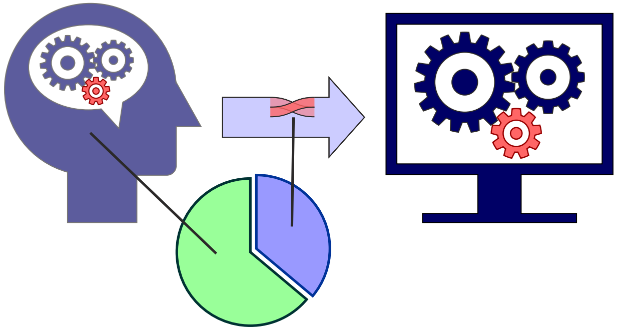
## Minor mistakes and misconceptions



## Minor mistakes and misconceptions



## Minor mistakes and misconceptions



## Some syntax errors are invisible

## Hidden syntax errors

```python
count = 0
for c in my_text:
    if c == 'e':
        count =+ 1

print(count)
```

## Hidden syntax errors

```
count = 0
for c in my_text:
    if c == 'e':
        count =+ 1

print(count)
```

## Hidden syntax errors

```python
def smallest(x, y, z):
    if x and y > z:
        return z
    if x and z > y:
        return y
    if y and z > x:
        return x
```

## Hidden syntax errors

```
def smallest(x, y, z):
    if x and y > z:
        return z
    if x and z > y:
        return y
    if y and z > x:
        return x
```

## Hidden syntax errors

```
while True:
    key = getKey()
    if key == LEFT:
        left(90)
    elif key == RIGHT:
        right(90)
else:
        forward(1)
```

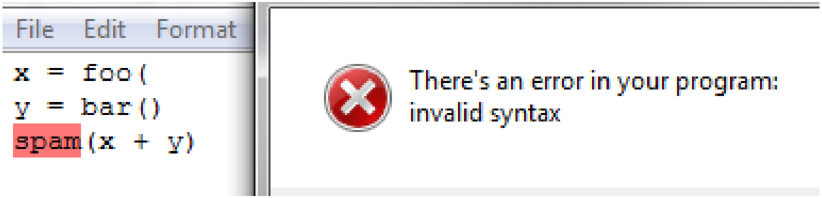
## Hidden syntax errors

```
while True:
    key = getKey()
    if key == LEFT:
        left(90)
    elif key == RIGHT:
        right(90)
else:
        forward(1)
```

## II. FINDING SOLUTIONS

## Better error messages

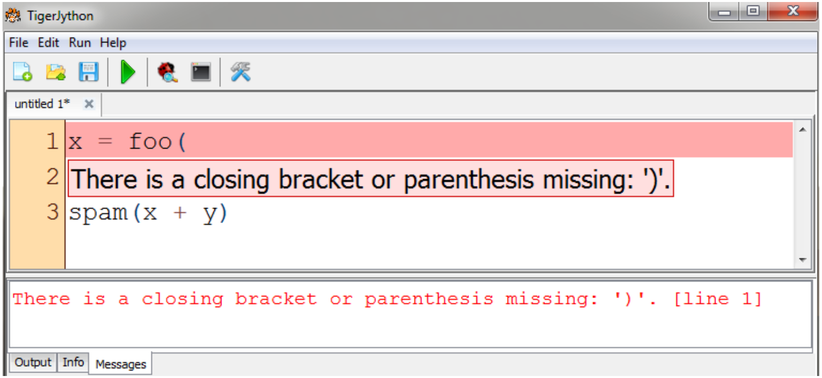
## Better error messages



```
x = foo(
y = bar()
spam(x + y)
```

There's an error in your program:
invalid syntax

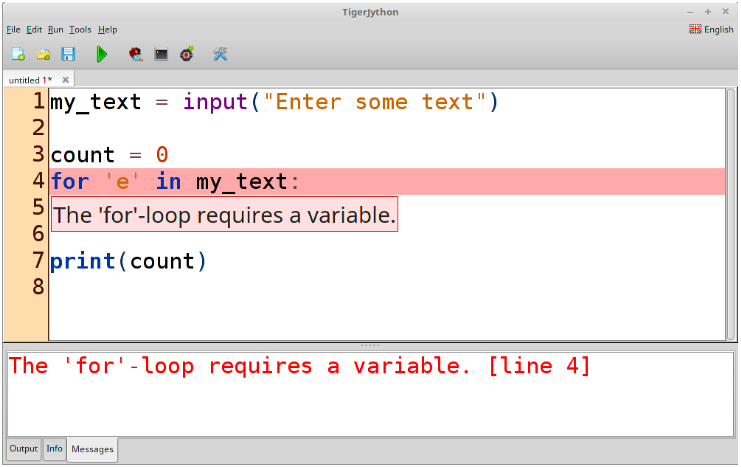
## Better error messages



```
x = foo(
There is a closing bracket or parenthesis missing: ')'.
spam(x + y)
```

```
There is a closing bracket or parenthesis missing: ')'. [line 1]
```

## Better error messages

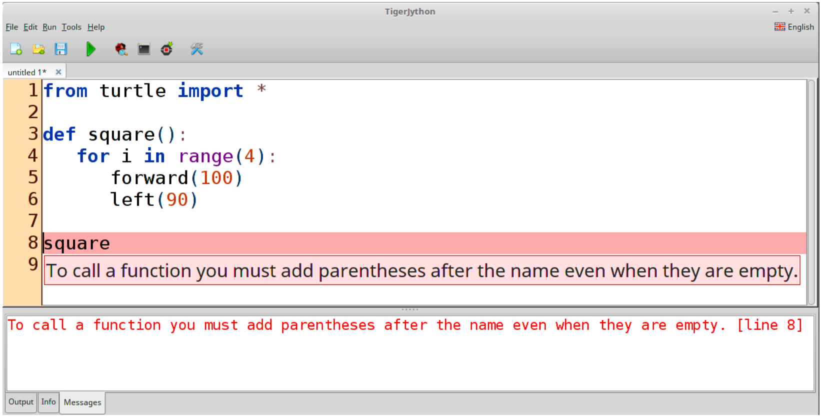


```
my_text = input("Enter some text")

count = 0
for 'e' in my_text:
The 'for'-loop requires a variable.

print(count)
```

```
The 'for'-loop requires a variable. [line 4]
```

UNIVERSITY OF CAMBRIDGE

## Better error messages



```
from turtle import *

def square():
    for i in range(4):
        forward(100)
        left(90)

square
```

To call a function you must add parentheses after the name even when they are empty.

```
To call a function you must add parentheses after the name even when they are empty. [line 8]
```

## Does it work? – Yes, but. . .

## Students’ understanding is brittle

## Students’ understanding is brittle

**NameError: name ’S’ is not defined**

```
def hexagon(s):
    for i in range(6):
        forward(S)
        right(60)

hexagon(100)
```

## Students' understanding is brittle

**NameError: name 'S' is not defined**

```
def hexagon(s):
    for i in range(6):
        forward(S)
        right(60)

hexagon(100)
```

```
def hexagon(s):
    for i in range(6):
        forward(100)
        right(60)

hexagon(100)
```

## Students’ understanding is brittle

**IndentationError: expected an indented block**

```
for i in range(4):
forward(100)
right(90)
```

## Students' understanding is brittle

**IndentationError: expected an indented block**

```python
for i in range(4):
forward(100)
right(90)
```

```python
forward(100)
right(90)
forward(100)
right(90)
forward(100)
right(90)
forward(100)
right(90)
```

## Students’ understanding is brittle

**You need parentheses to call a function**

```
def square():
    for i in range(4):
        forward(100)
        left(90)

square
```

## Students' understanding is brittle

**You need parentheses to call a function**

```
def square():
    for i in range(4):
        forward(100)
        left(90)

square
```

```
def square():
    for i in range(4):
        forward(100)
        left(90)

(square)
```

## Addressing miconceptions

## Variables and Assignment

Test whether *x* is positive before computing the square root of *x*.

## Variables and Assignment

Test whether *x* is positive before computing the square root of *x*.

```
x = input("Enter a number:")
y = sqrt(x)
if x >= 0:
    print(y)
else:
    print("No result exists")
```

## Variables and Assignment

*What image does the turtle draw?*

```
s = 1
t = 3 * s + 1
for i in range(4):
    forward( t )
    left( 90 )
    s += 2
```

## Variables and Assignment

- Students use *mathematical* reasoning:
  `y = sqrt(x)` establishes a relationship between *x* and *y*.
- *Lazy* evaluation:
  *y* is (re)computed from *x* when *y* is *used/required*.
- Even variables and assignment can be difficult!

UNIVERSITY OF CAMBRIDGE

## Variables and Assignment

- Students use *mathematical* reasoning:
  `y = sqrt(x)` establishes a relationship between *x* and *y*.
- *Lazy* evaluation:
  *y* is (re)computed from *x* when *y* is *used/required*.
- Even variables and assignment can be difficult!
- **Make it explicit and discuss it in teaching!**

## Wrapping up...

## Wrapping up. . .

- **What is a “syntax error”?**
- **Where do syntax errors come from?**
- **What can we do about syntax errors?**

## Thank You