

Important! There are three ways for values to get into **parameters**:

1. positional **argument**
2. keyword/named **argument**
3. default **argument**

For these puzzles, the best way to practice is to write down what you think, then actually run the code in Python (in PythonTutor!) to check your mental model. Try thinking of your own examples to try too (this is active learning).

```
def add(x, y):  
    print(x) # for debugging  
    print(y) # for debugging  
    return x+y
```

What will be in the parameters for each call? [some crash; mark them]

- | | |
|--------------------|---|
| 1. add(3) | crashed? ___ OR x=_____, y=_____ |
| 2. add(3, 4) | crashed? ___ OR x=_____, y=_____ |
| 3. add(4, 3) | crashed? ___ OR x=_____, y=_____ |
| 4. add(3, 4, 5) | crashed? ___ OR x=_____, y=_____ |
| 5. add(6, y=7) | crashed? ___ OR x=_____, y=_____ |
| 6. add(7, x=6) | crashed? ___ OR x=_____, y=_____ |
| 7. add(7, 8, x=9) | crashed? ___ OR x=_____, y=_____ |
| 8. add(x=10, y=11) | crashed? ___ OR x=_____, y=_____ |
| 9. add(y=10, x=11) | crashed? ___ OR x=_____, y=_____ |

```
def say(word, t=1, end=""):  
    # add prints here to debug  
    print(word * t + end)
```

- | | |
|-----------------------------|--|
| 10. say() | crashed? ___ OR word=_____, t=____, end=_____ |
| 11. say("ha") | crashed? ___ OR word=_____, t=____, end=_____ |
| 12. say("ha", t=2) | crashed? ___ OR word=_____, t=____, end=_____ |
| 13. say("ha", end="!") | crashed? ___ OR word=_____, t=____, end=_____ |
| 14. say(t=2, word="yo") | crashed? ___ OR word=_____, t=____, end=_____ |
| 15. say("W", 3, "eb") | crashed? ___ OR word=_____, t=____, end=_____ |
| 16. say(t=3) | crashed? ___ OR word=_____, t=____, end=_____ |
| 17. say("W", end=".", 3) | crashed? ___ OR word=_____, t=____, end=_____ |
| 18. say("huh", 1, "?", "!") | crashed? ___ OR word=_____, t=____, end=_____ |

19. What letters are printed, and in what order?

```
print("A")

def foo():
    print("B")
print("C")
foo()
print("D")
foo()
```

20. Repeat question 19, but imagine a tab before `print("C")`

21. What is printed, and in what order?

```
def func_c():
    print("C")

def func_b():
    print("B1")
    func_c()
    print("B2")

def func_a():
    print("A1")
    func_b()
    print("A2")

func_a()
```

22. What is printed, and in what order?

```
def f():
    print("A")
    return("B")
    print("C")
print("D")
x = f()
print("E")
print(x)
```