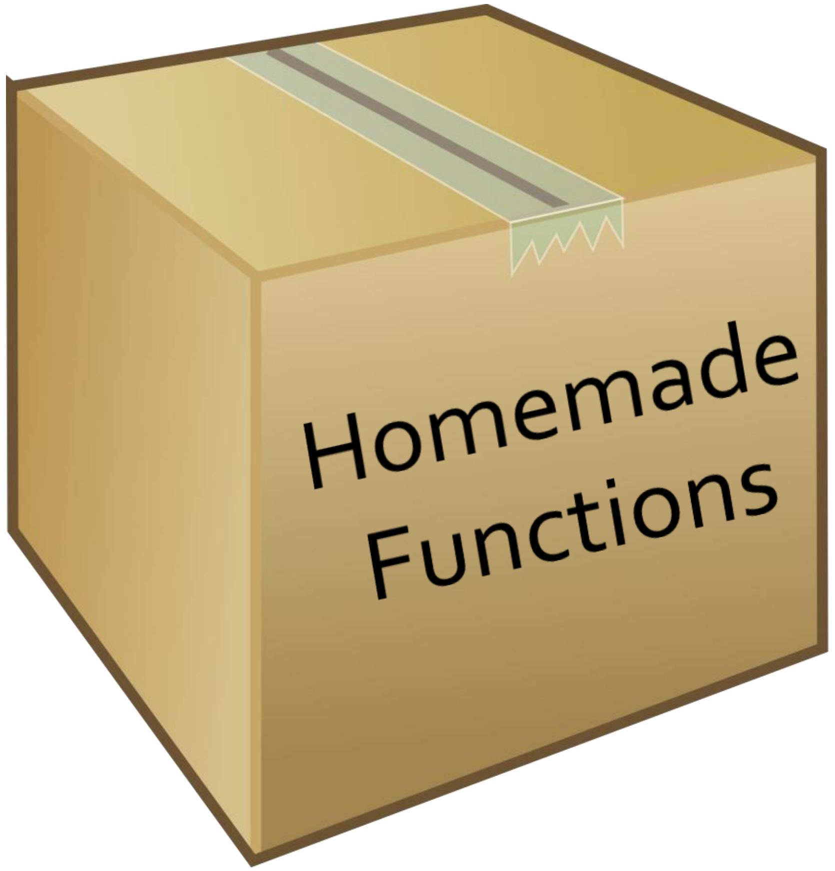
DEFINING FUNCTIONS

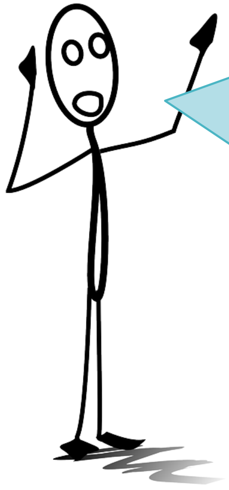
An Introduction to Computer Science

Defining Functions



Why Functions?

1. Code Reuse

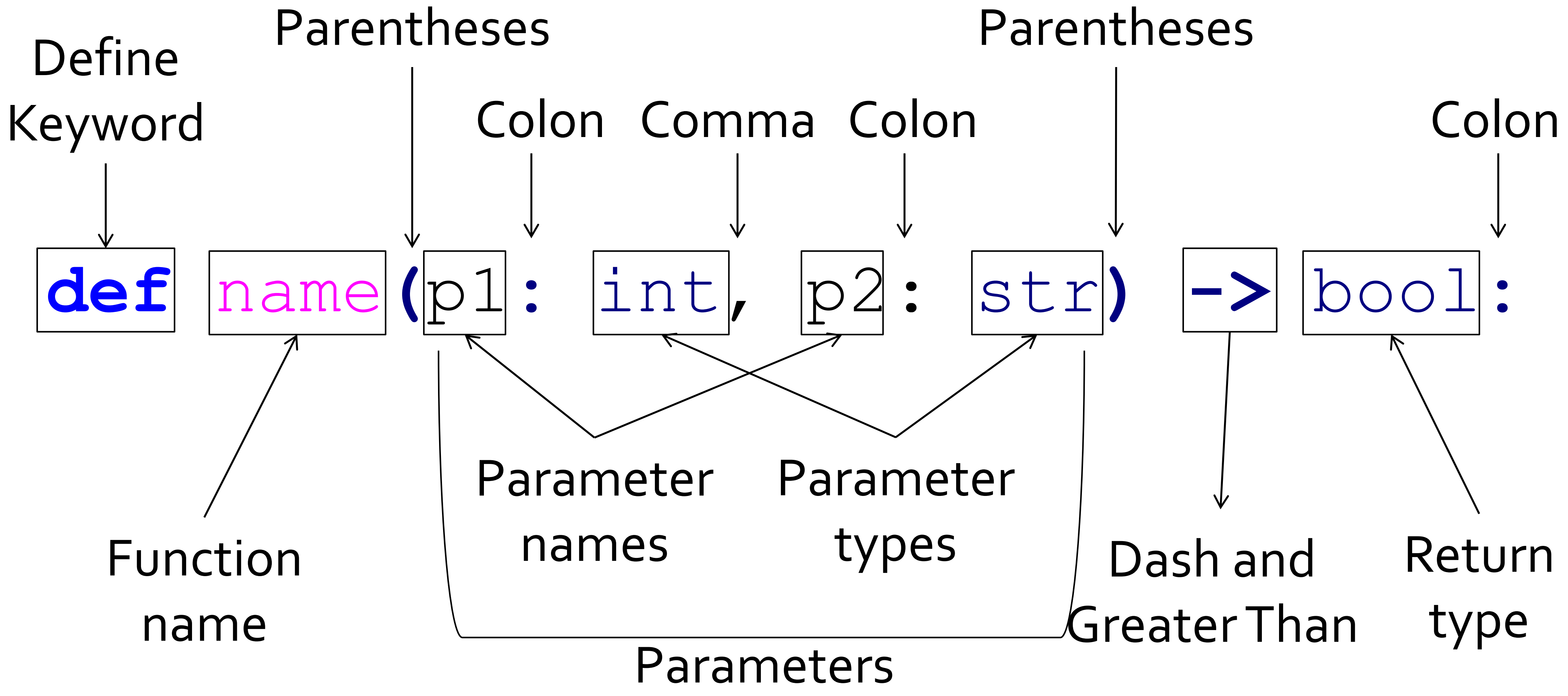


I need to use this multiple times! It should be a function.

2. Easier to debug

Why isn't this part working? I'll extract it into a function to test it.

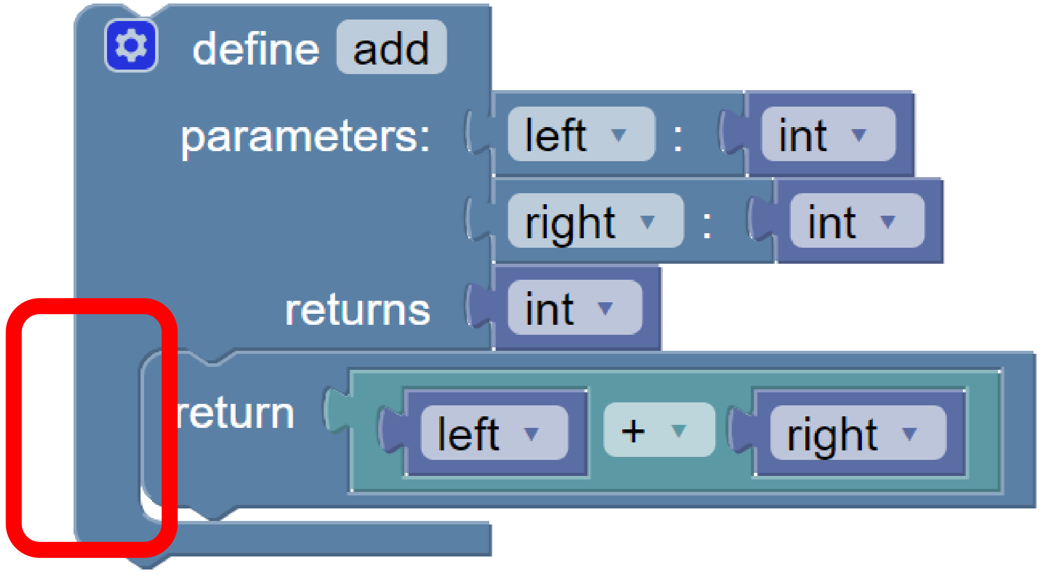
Definition Syntax



Function Body

```
def add(left: int, right: int) -> int:
    return left + right
```

Indent with 4 spaces



Naming a function

1. Use verbs
1. Function names can only have
   - Letters (abcABC)
   - Numbers (123)
   - Underscores (_)
2. Function Names must not begin with
   - Numbers

Calling Your Functions

```
def add5(a_number: int):
    return a_number + 5

add5(10)   15
add5(3)    8
```

Parameters

```python
def subtract(first: int, second: int) -> int:
    return first - second

subtract(3, 8)
subtract(-2, 5)
subtract(10, 10)
```

Parameters and Types

Parameter type

Parameter type

```
def get_speed(distance: int, time: int) -> float:
    return distance / time

get_speed(6, 2)
get_speed(3, 8)
```

These must match the parameter types!

Return

Return type

```python
def area(length:int, width:int) -> int:
    return length * width
```

Arrow

Return statement

Calling and Printing

```
def area(length:int, width:int) -> int:
    return length * width

print(area(3, 4))
```
Prints 12

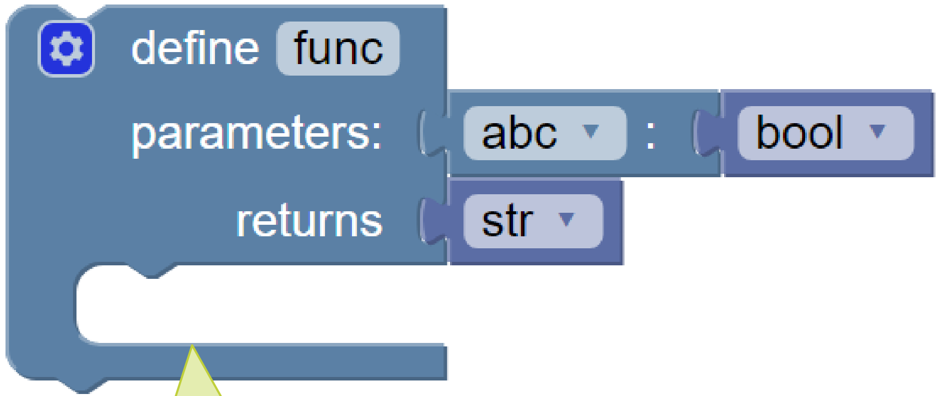
```
area(1, 8)
```
Calculates 8 but does not print!

```
print(area(5, 2))
```
Prints 10

Pass

```
def func(abc: bool)->str:
    pass
```

The "pass" means "do nothing"



Nothing there!