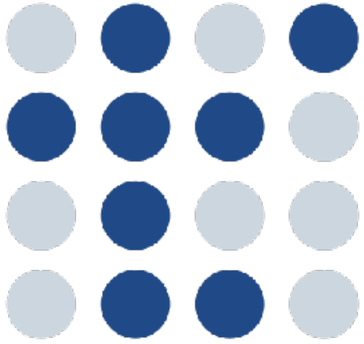
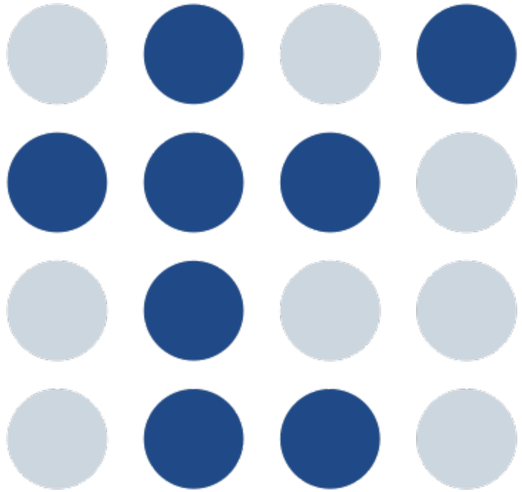


oxford university
compsoc

Introduction to Python

Learn to Code 2018



oxford university

compsoc

ox.compsoc.net

oxcompsoc

@oxcompsoc

Logistics

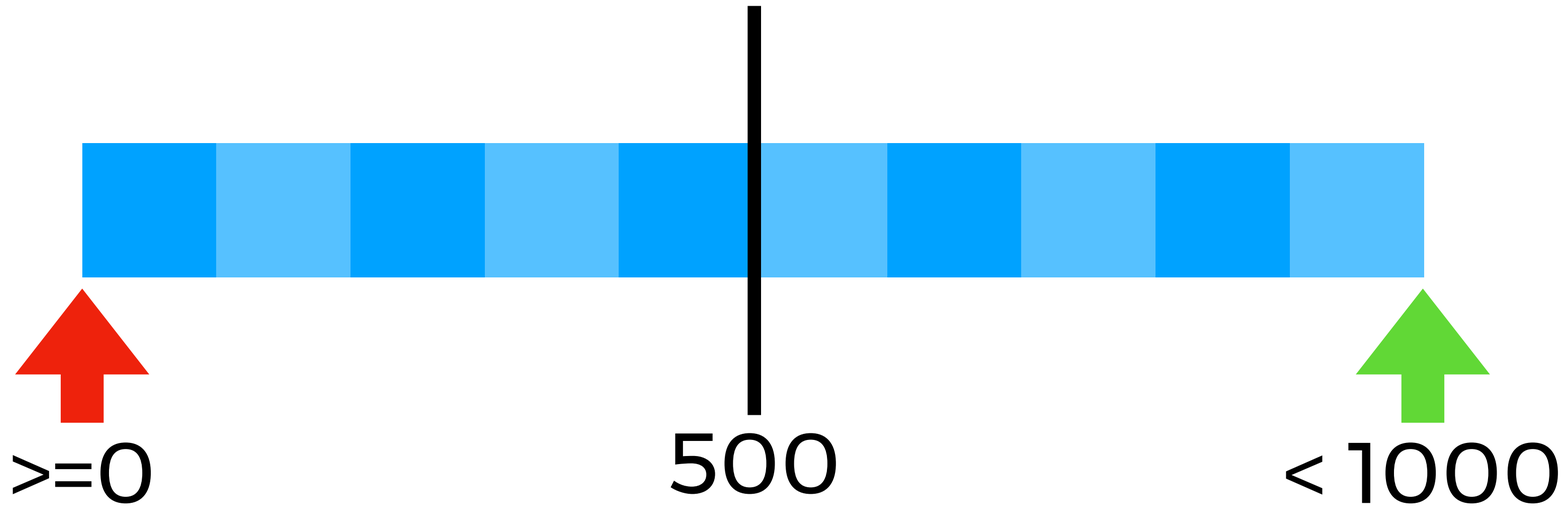
- Every Thursday (weeks 2-7) at 7pm for about an hour
- Lecture Theatre A, overflow into Lecture Theatre B (same content in both)
- Lectures with regular breaks for exercises

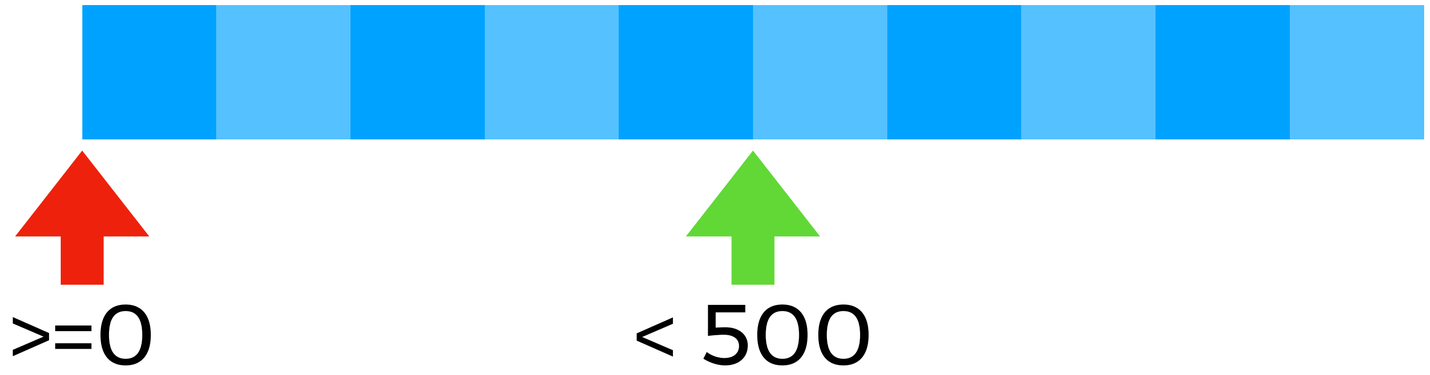
github.com/oxcompsoc/learntocode

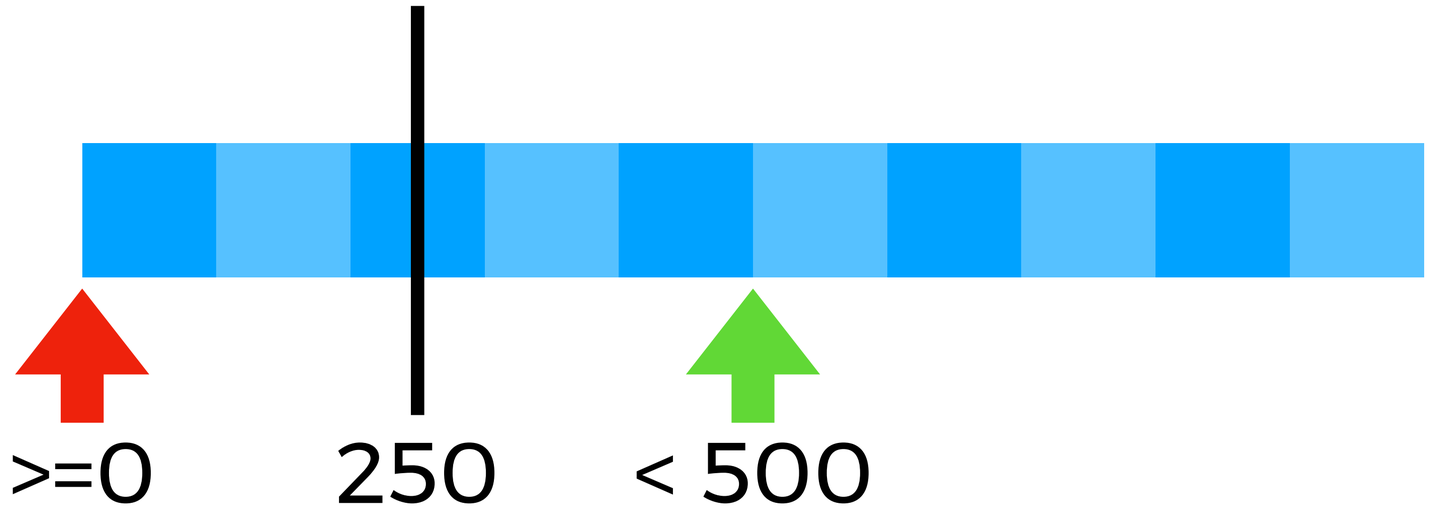
Guessing game

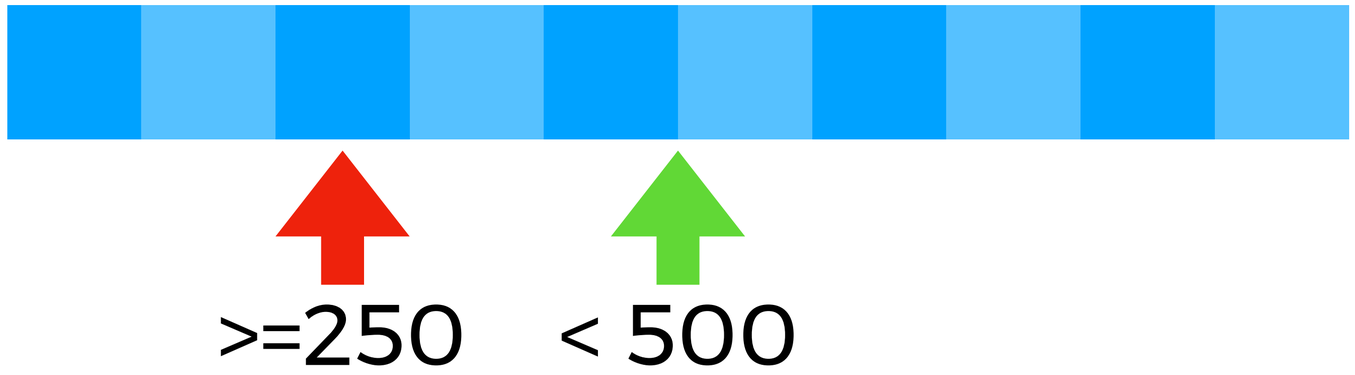
Computer guessing game

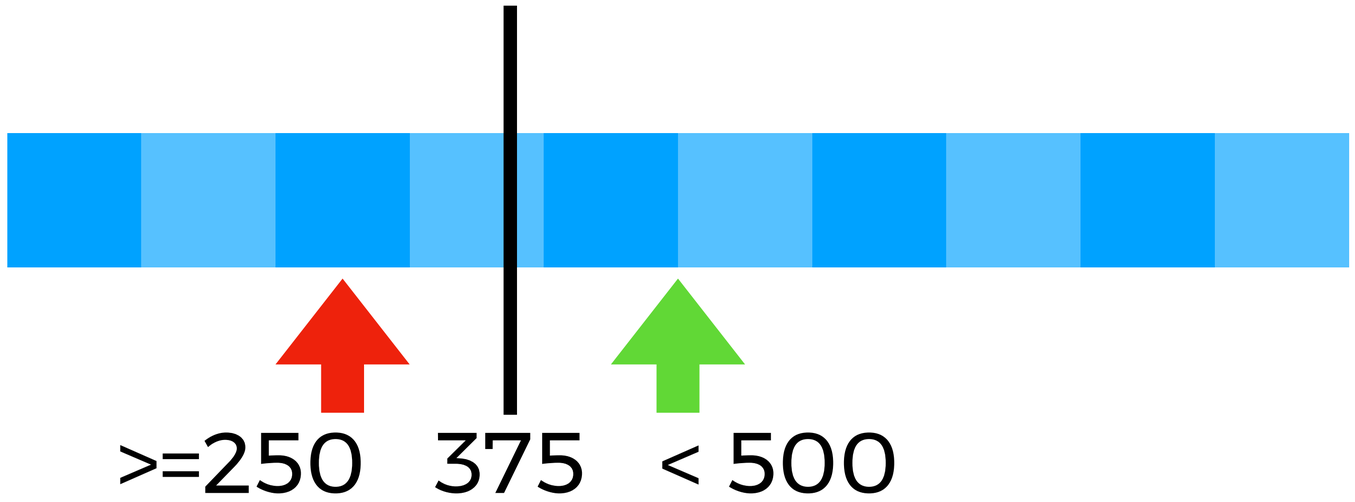

>=0
<1000

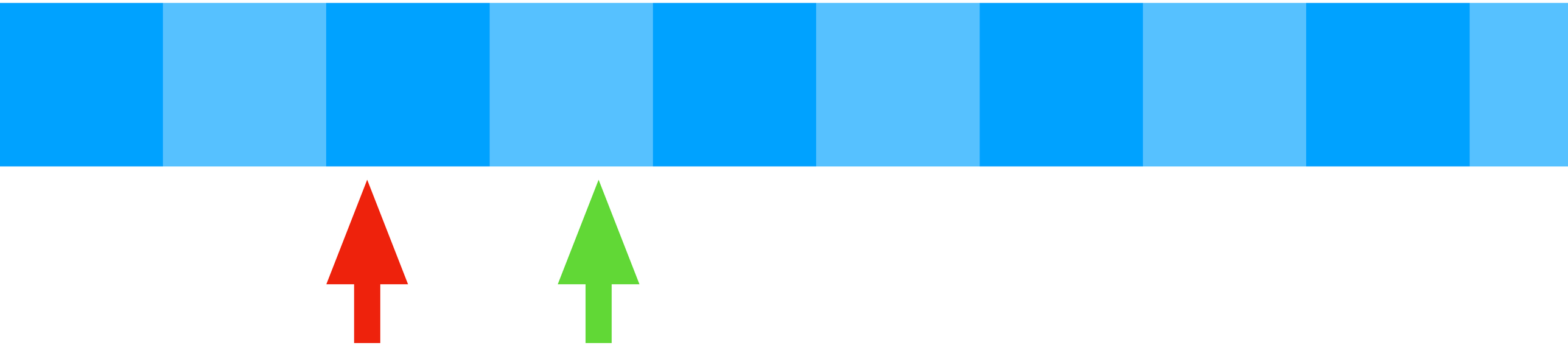

>=0
500
<1000

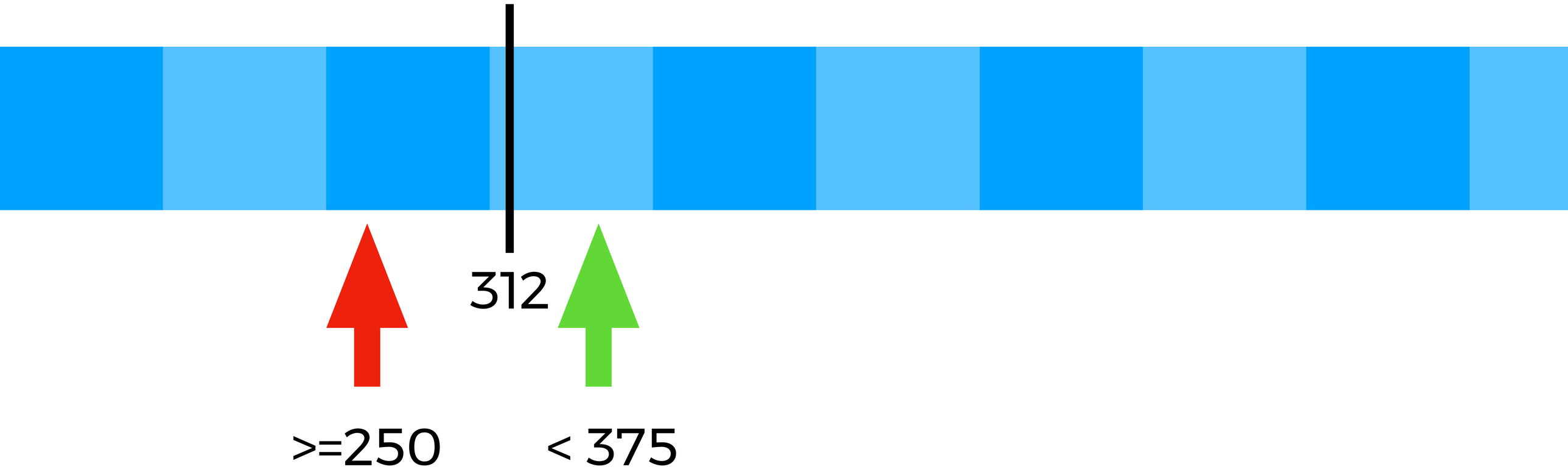

>=0
< 500

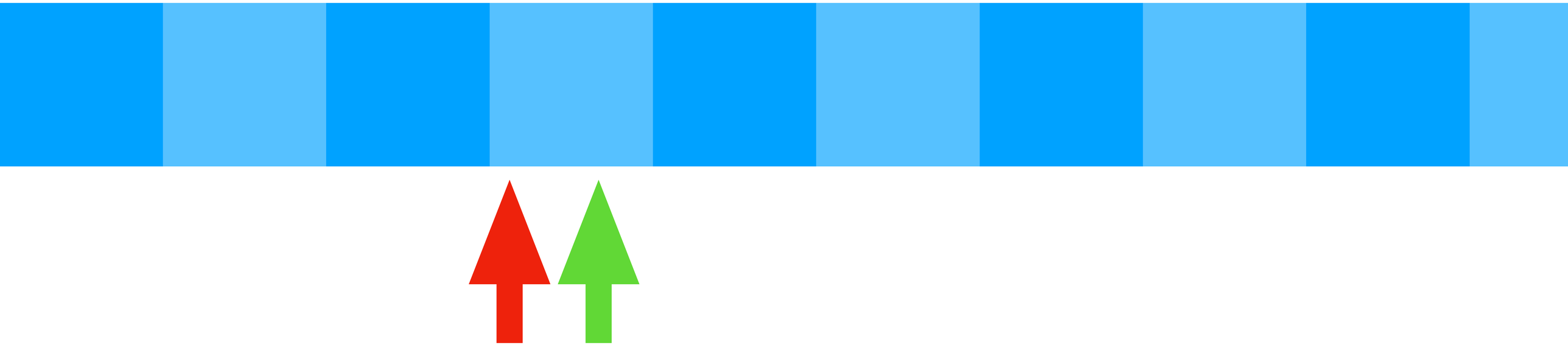

>=0
250
< 500


>=250
< 500

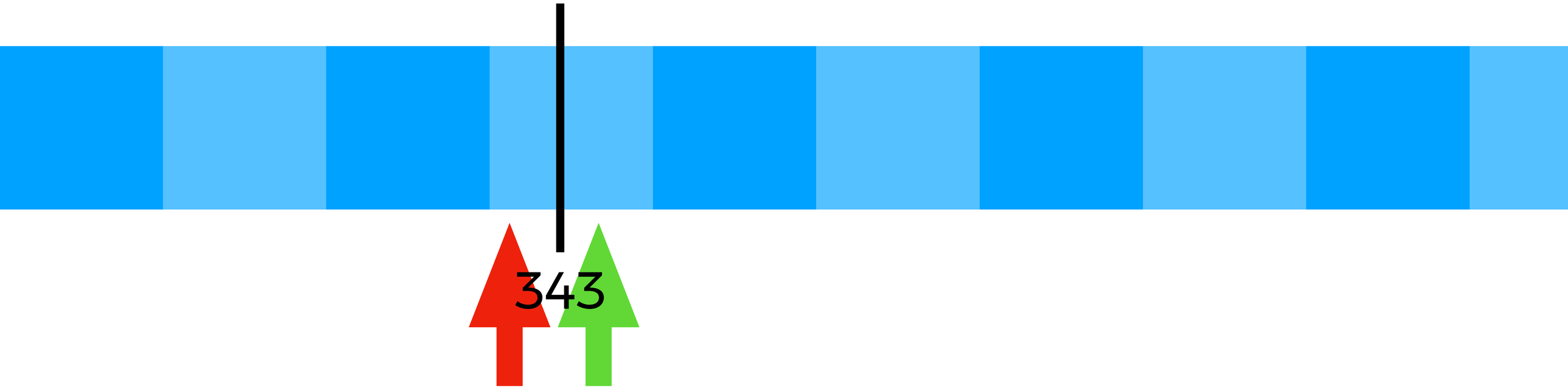

>=250
375
< 500

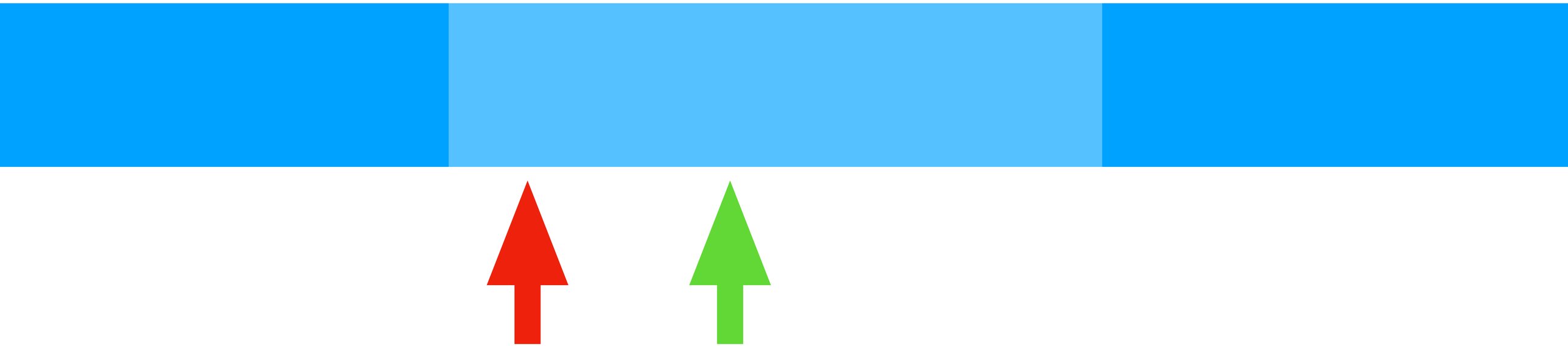

>=250
< 375

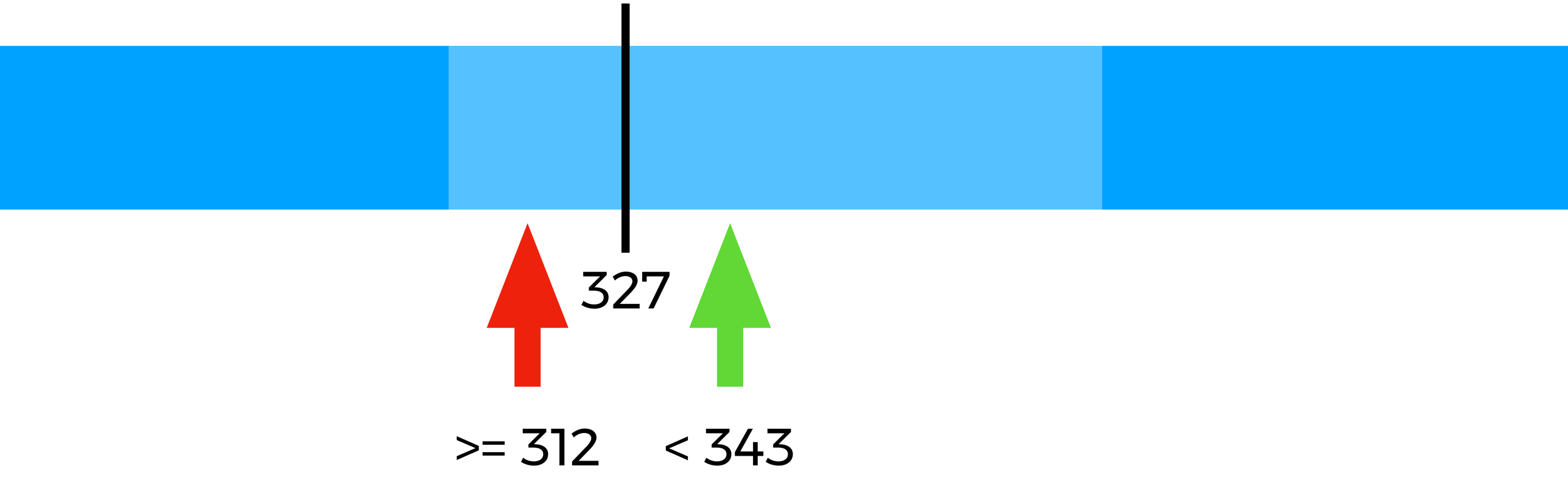

312
>=250
< 375

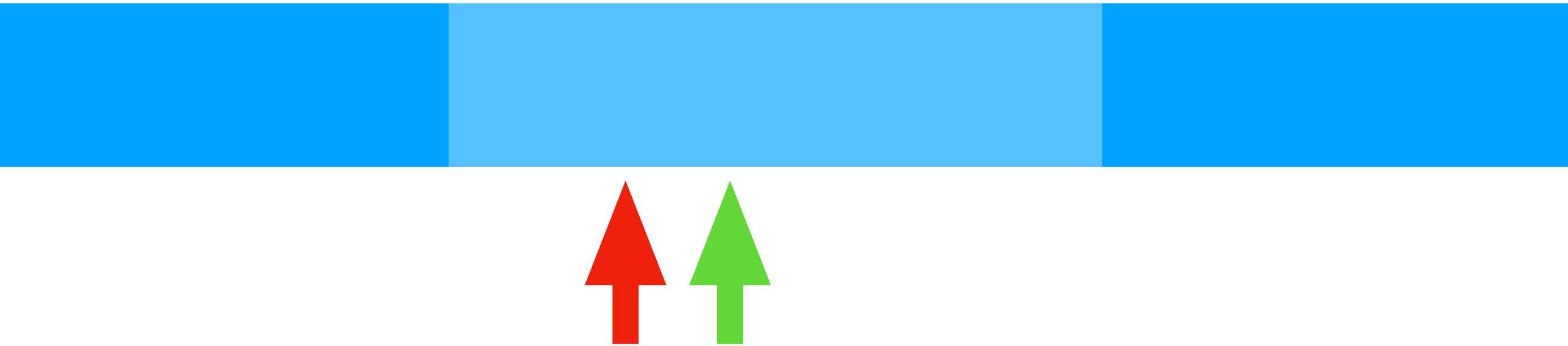


>= 312 < 375

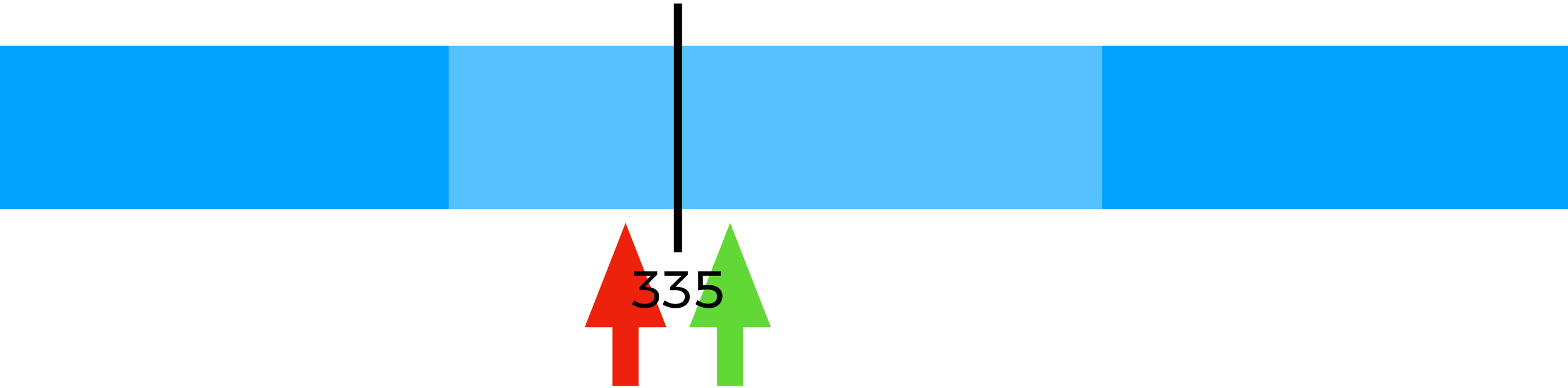

343
>= 312
< 375

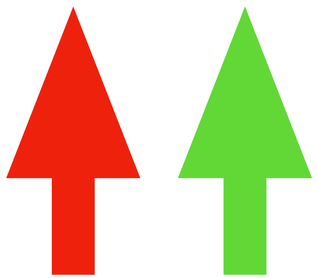

>= 312
< 343


327
>= 312
< 343



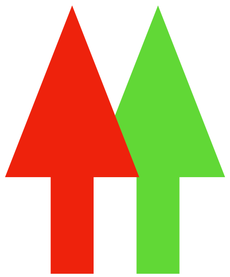
>= 327 < 343


335
>= 327
< 343



>= 327 < 335

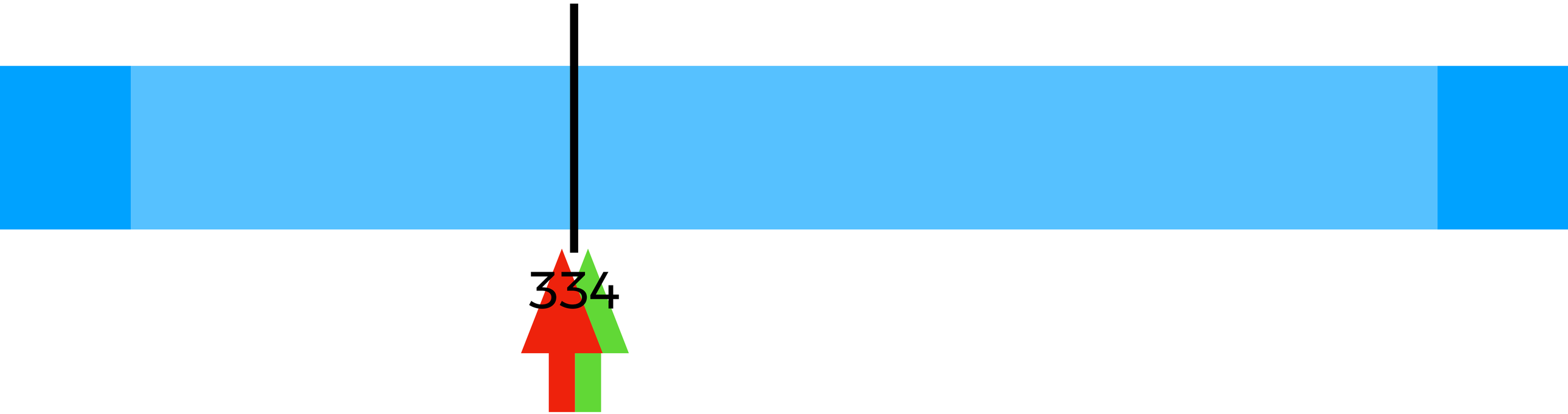

331
>= 327
< 335



>= 331 < 335


333
>= 331
< 335

>= 333 < 335



>= 333 < 335

>= 333 < 334

>= 333

< 334

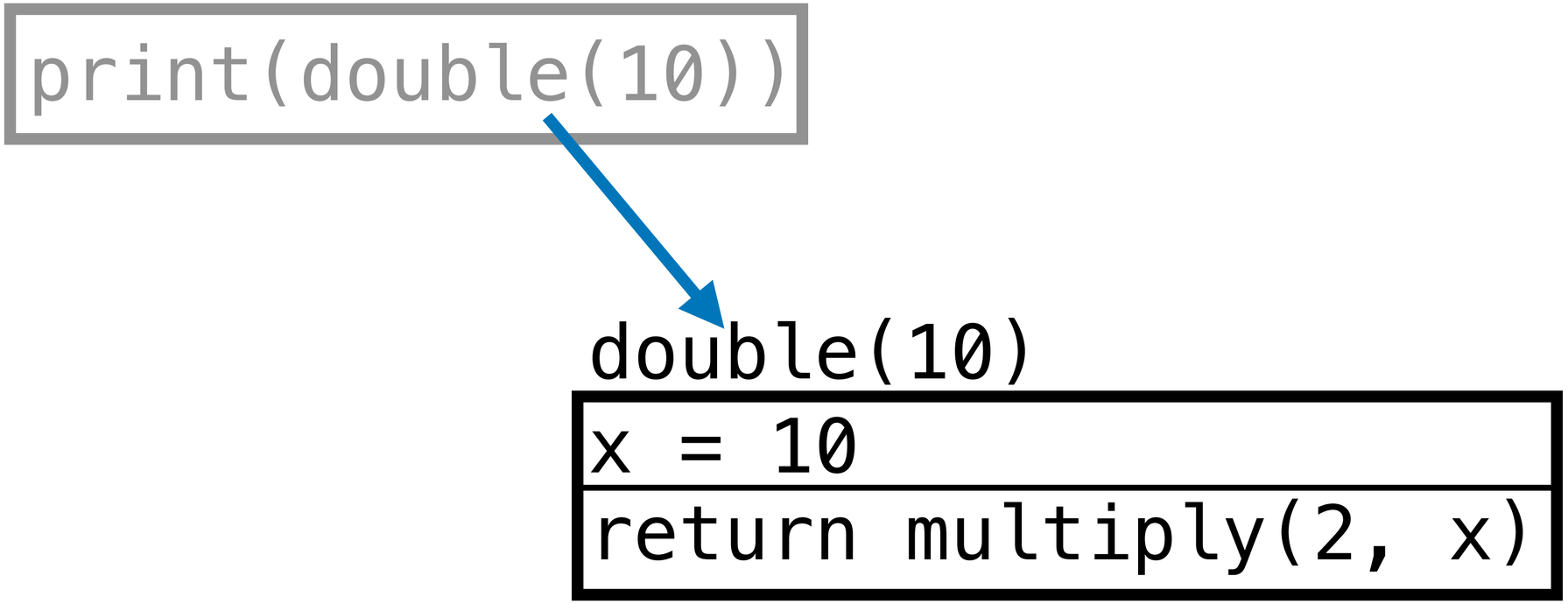
333 >= 333

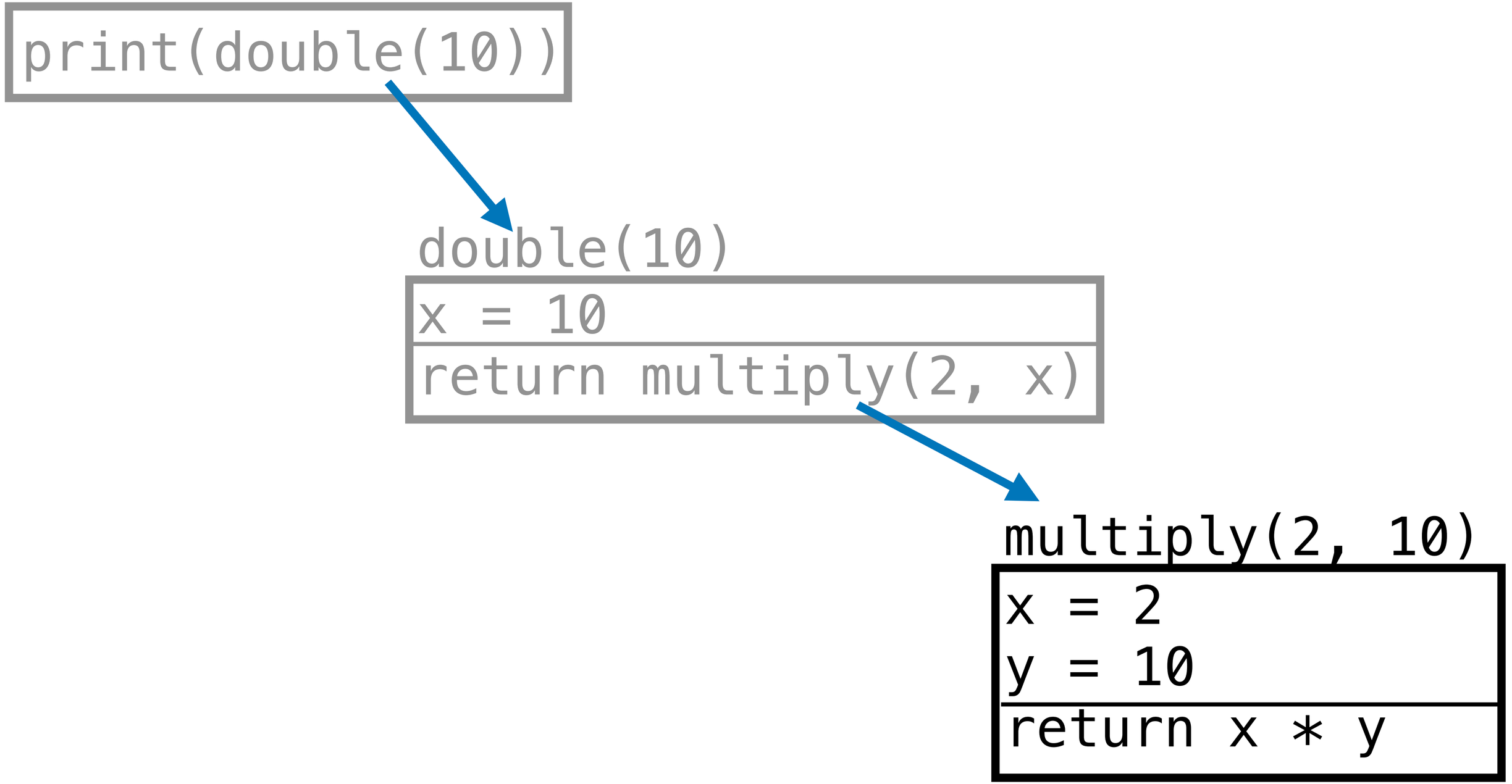
333 < 334

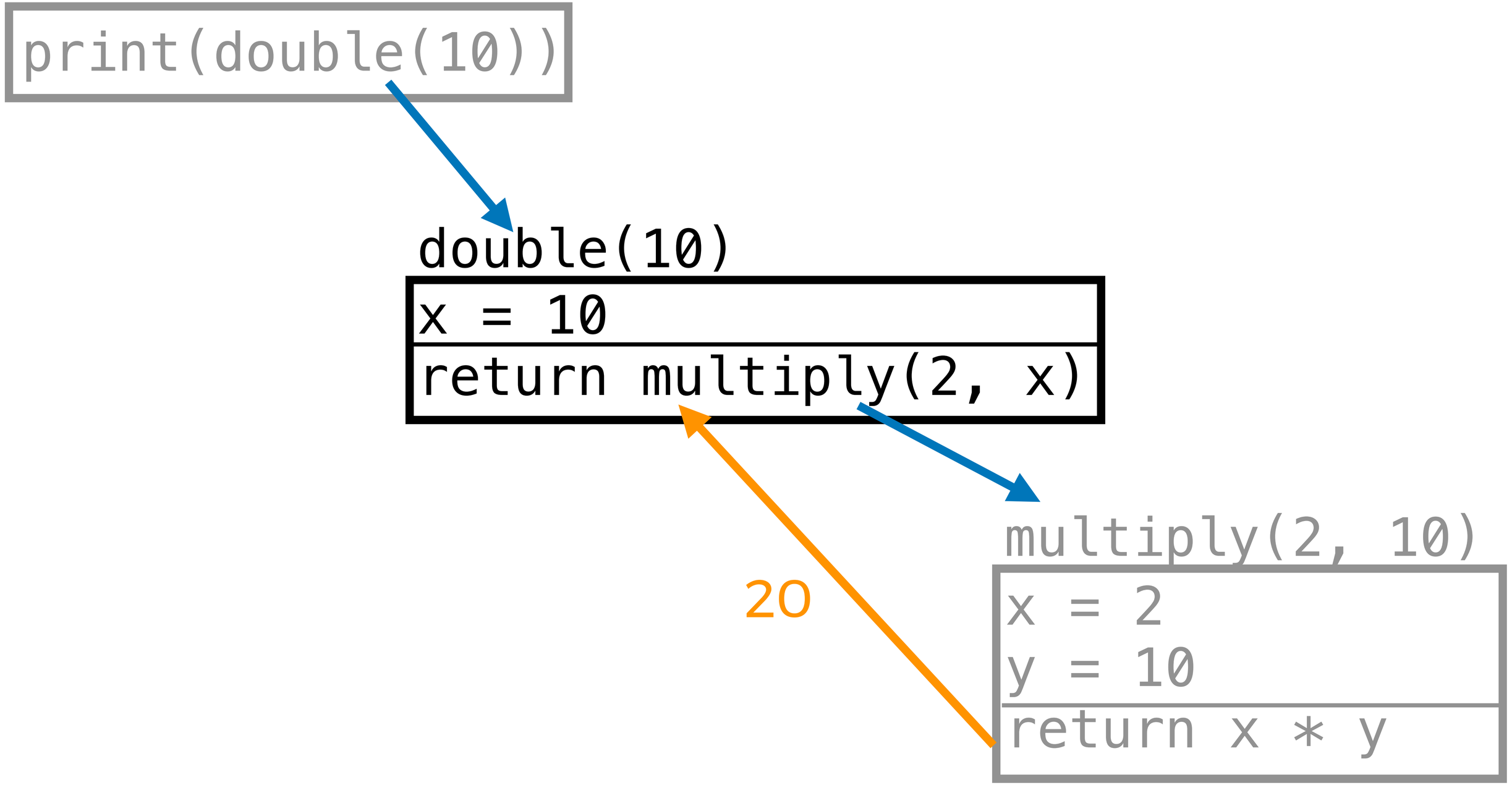
Binary search

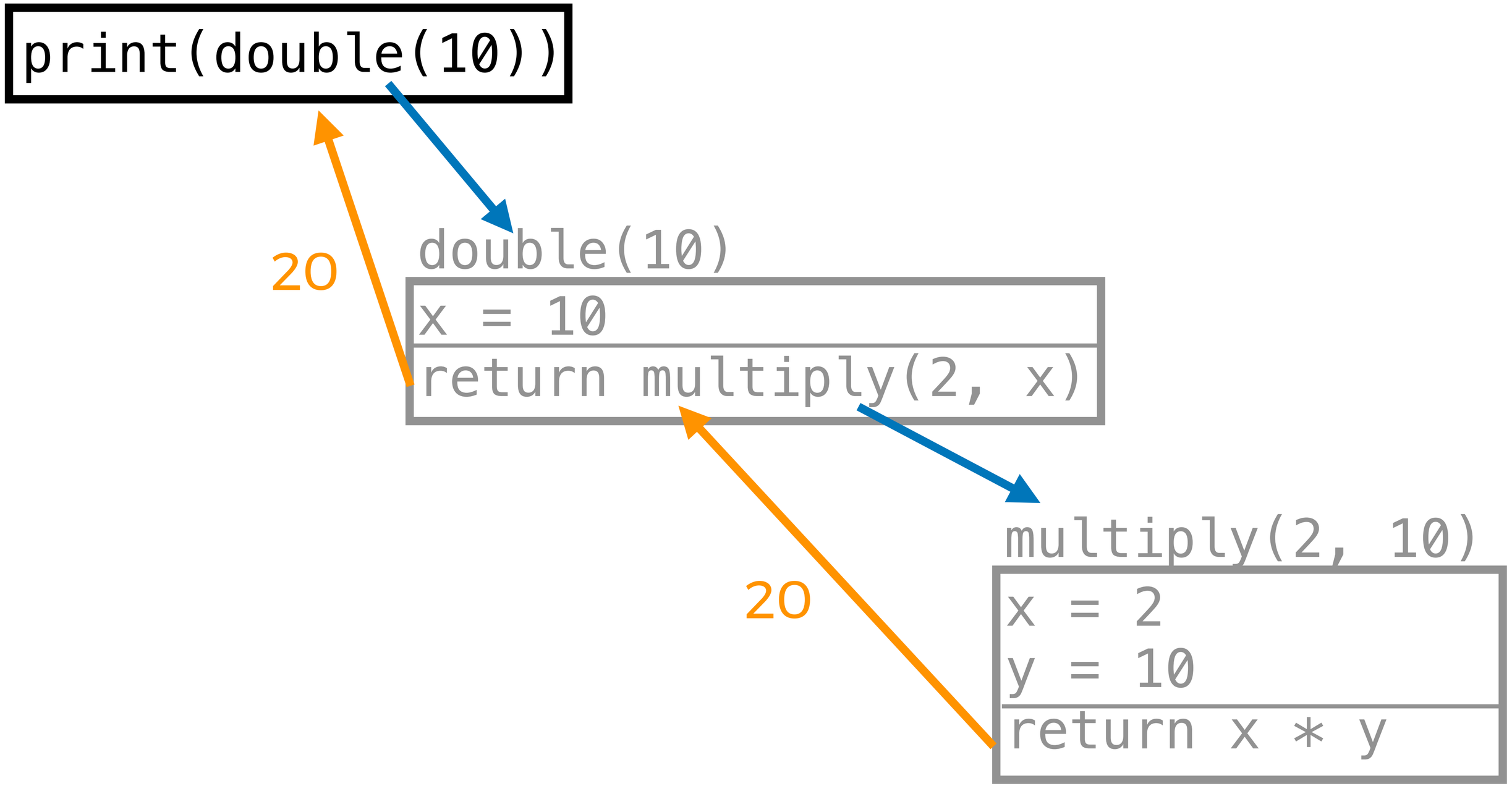
Functions

```
print(double(10))
```


print(double(10))
double(10)
x = 10
return multiply(2, x)

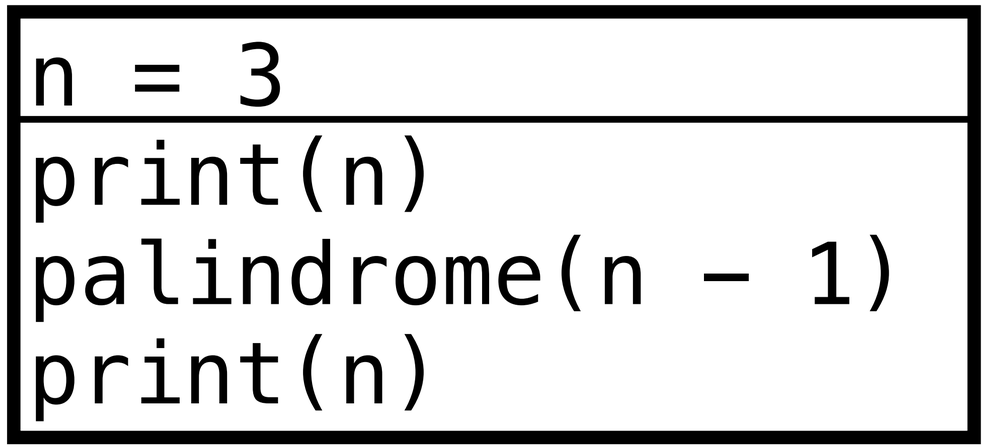

print(double(10))
double(10)
x = 10
return multiply(2, x)
multiply(2, 10)
x = 2
y = 10
return x * y


print(double(10))
double(10)
x = 10
return multiply(2, x)
20
multiply(2, 10)
x = 2
y = 10
return x * y

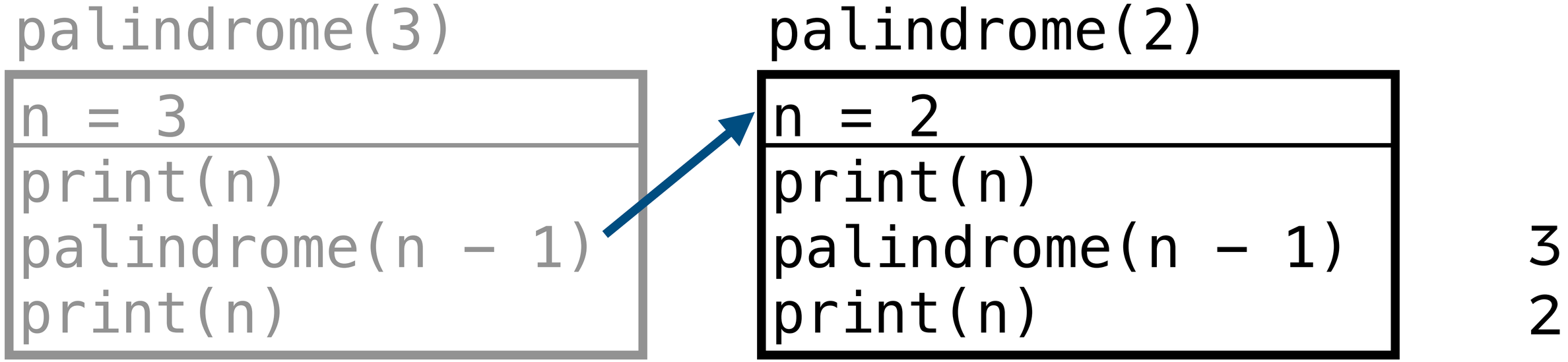

print(double(10))
20
double(10)
x = 10
return multiply(2, x)
20
multiply(2, 10)
x = 2
y = 10
return x * y

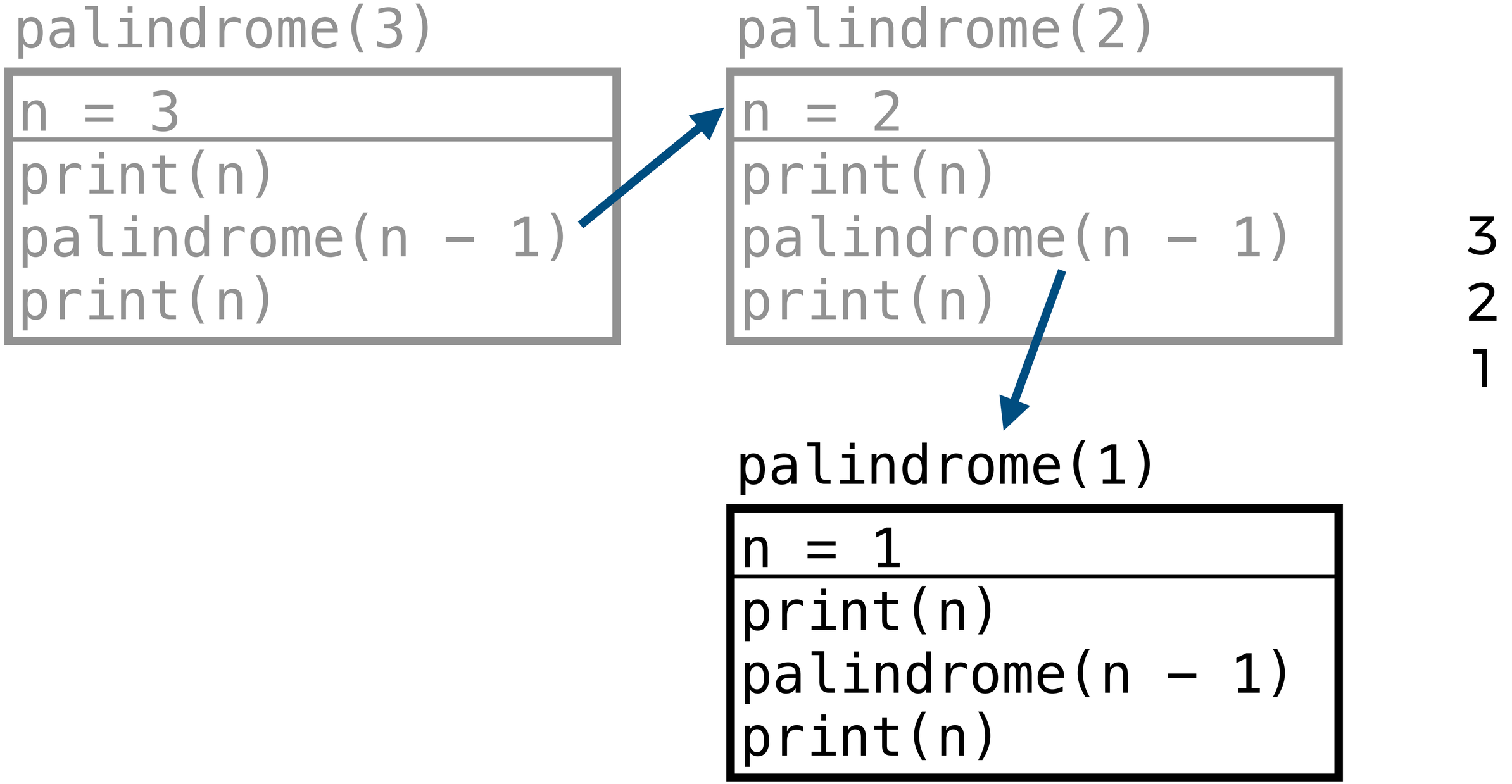
Recursion

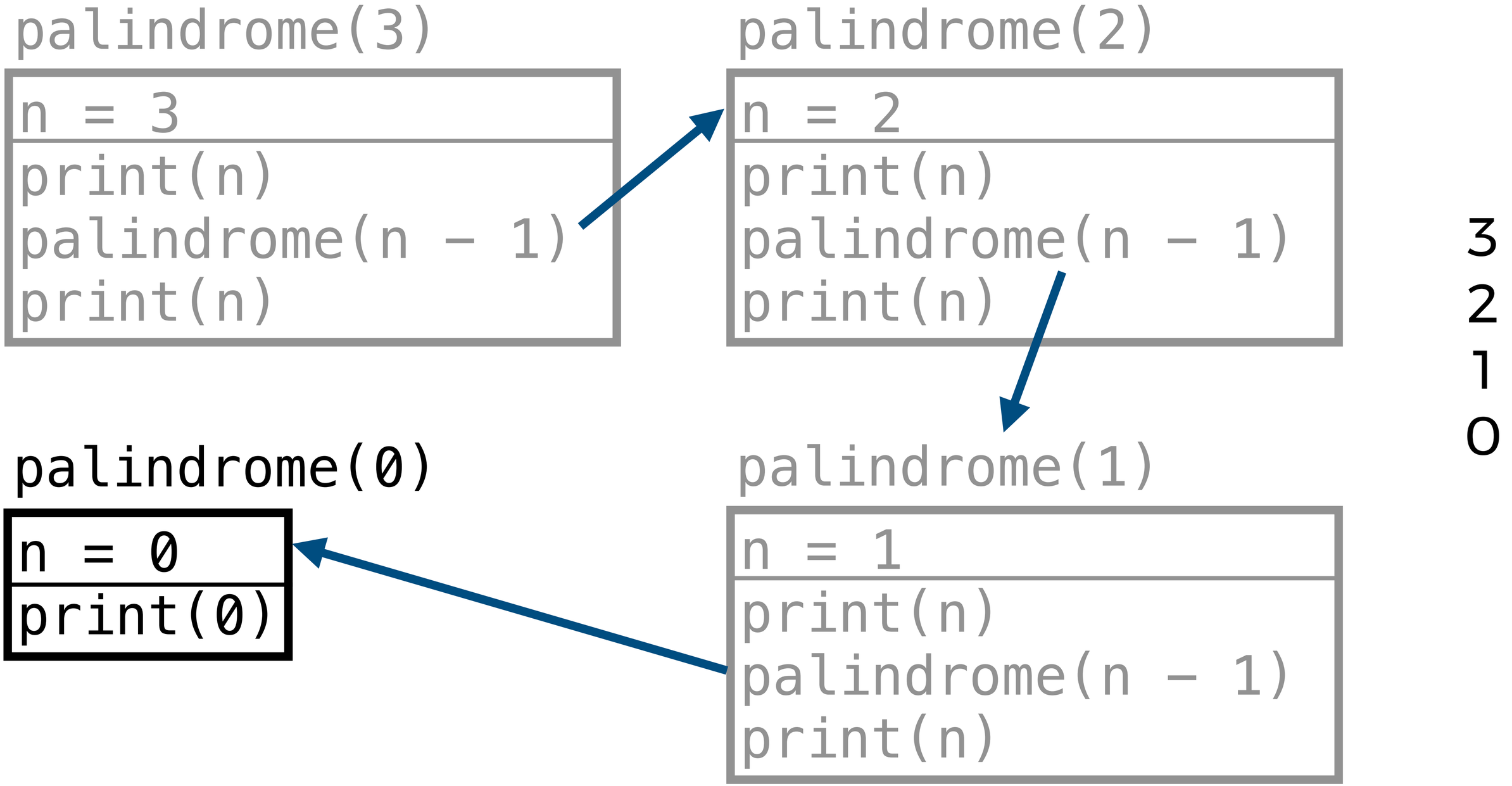
```
palindrome(3)
```

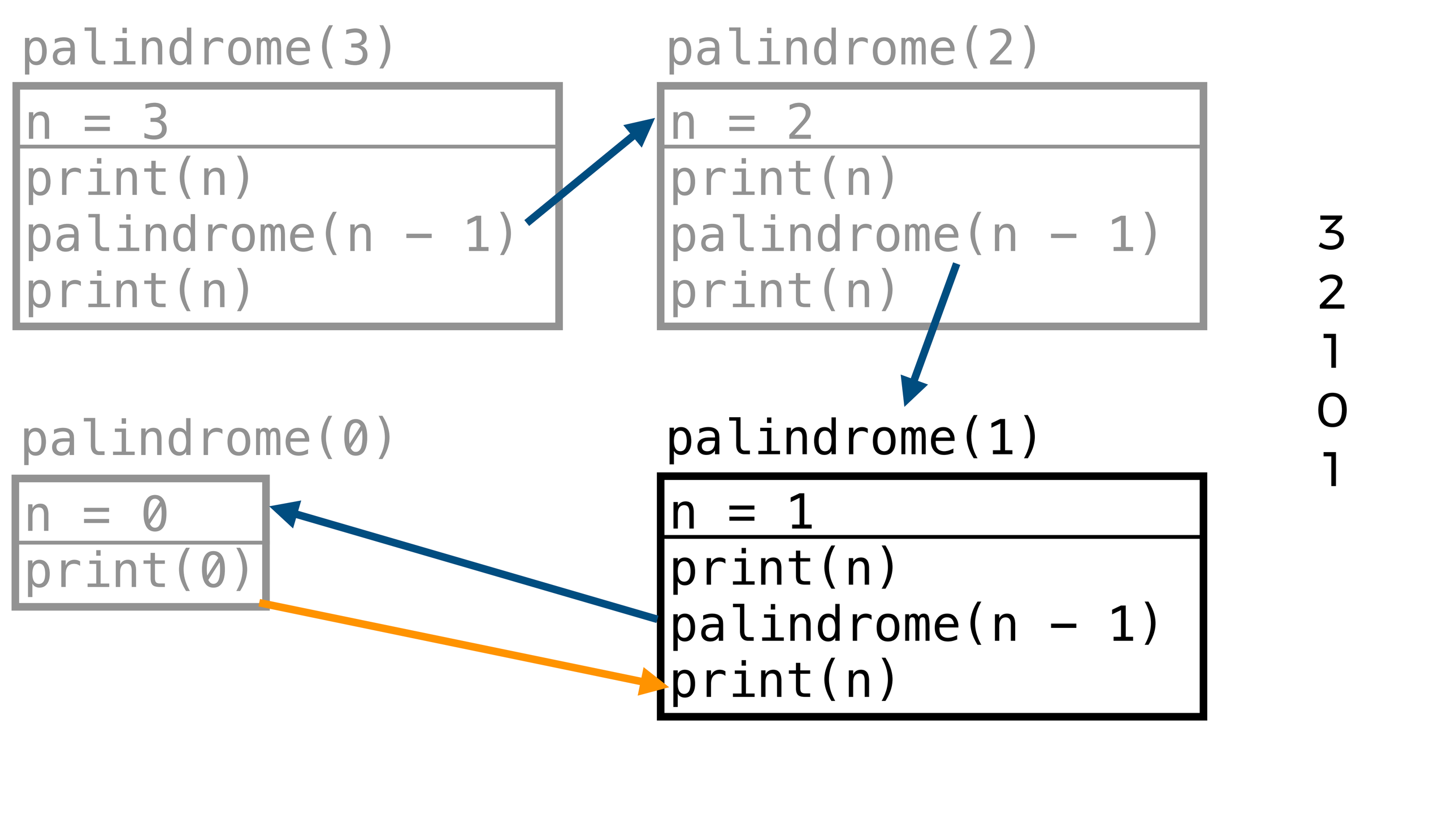


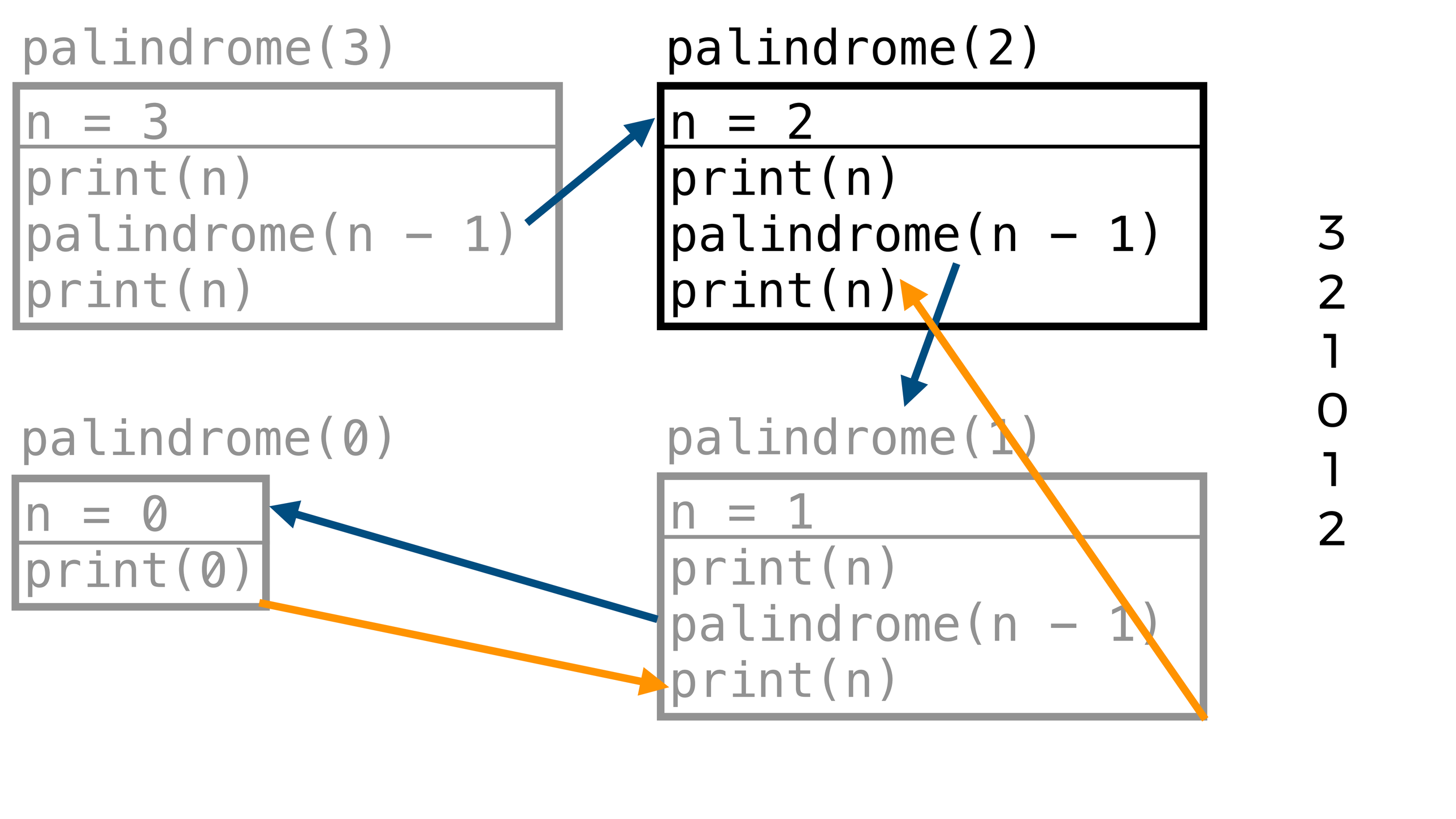
3

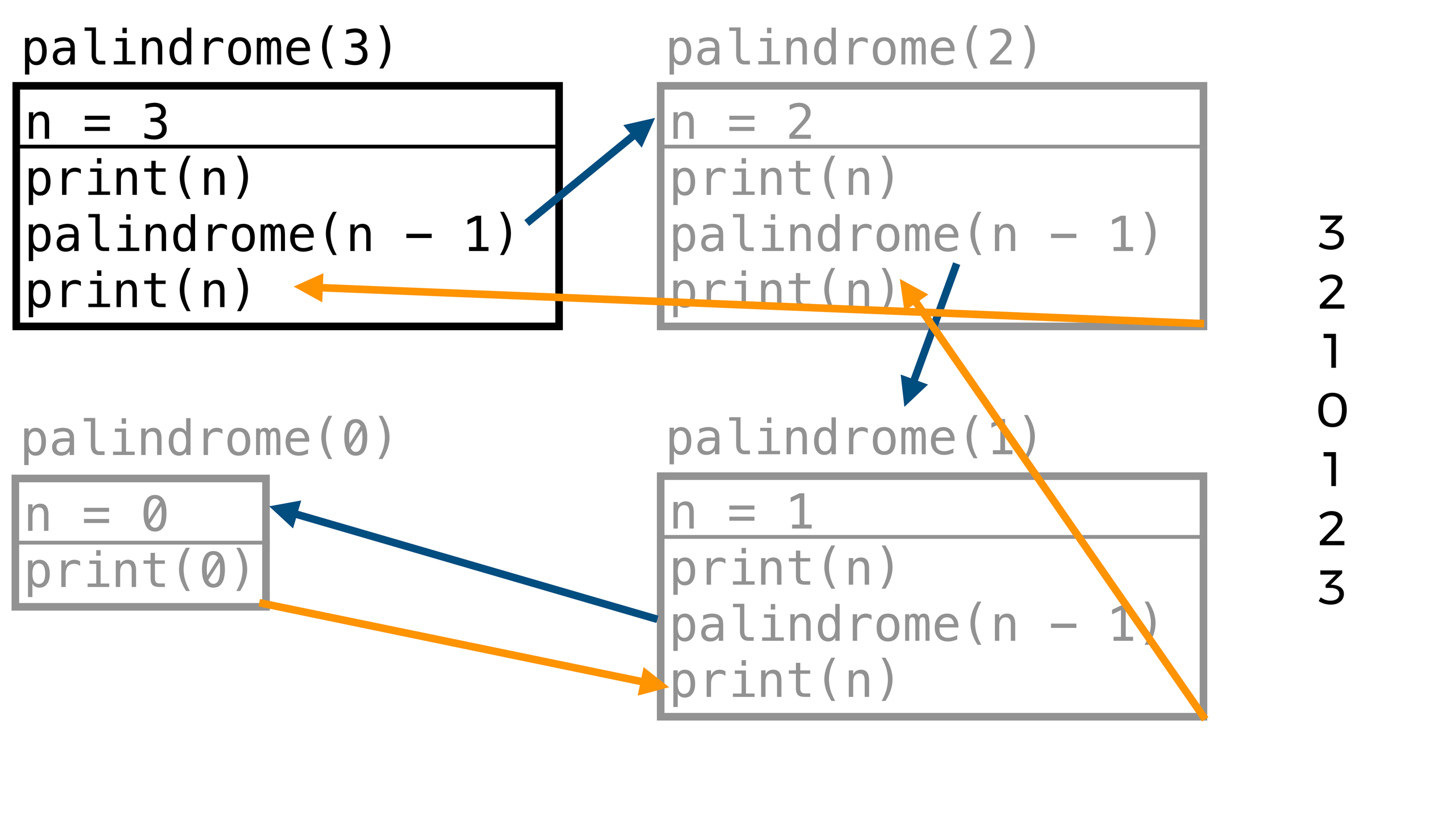

palindrome(3)
n = 3
print(n)
palindrome(n - 1)
print(n)
palindrome(2)
n = 2
print(n)
palindrome(n - 1)
print(n)
3
2


palindrome(3)
n = 3
print(n)
palindrome(n - 1)
print(n)
palindrome(2)
n = 2
print(n)
palindrome(n - 1)
print(n)
palindrome(1)
n = 1
print(n)
palindrome(n - 1)
print(n)
3
2
1


palindrome(3)
n = 3
print(n)
palindrome(n - 1)
print(n)
palindrome(2)
n = 2
print(n)
palindrome(n - 1)
print(n)
palindrome(1)
n = 1
print(n)
palindrome(n - 1)
print(n)
palindrome(0)
n = 0
print(0)
3
2
1
0


palindrome(3)
n = 3
print(n)
palindrome(n - 1)
print(n)
palindrome(2)
n = 2
print(n)
palindrome(n - 1)
print(n)
palindrome(0)
n = 0
print(0)
palindrome(1)
n = 1
print(n)
palindrome(n - 1)
print(n)
3
2
1
0
1


palindrome(3)
n = 3
print(n)
palindrome(n - 1)
print(n)
palindrome(2)
n = 2
print(n)
palindrome(n - 1)
print(n)
palindrome(0)
n = 0
print(0)
palindrome(1)
n = 1
print(n)
palindrome(n - 1)
print(n)
3
2
1
0
1
2


palindrome(3)
n = 3
print(n)
palindrome(n - 1)
print(n)
palindrome(2)
n = 2
print(n)
palindrome(n - 1)
print(n)
palindrome(0)
n = 0
print(0)
palindrome(1)
n = 1
print(n)
palindrome(n - 1)
print(n)
3
2
1
0
1
2
3

Fibonacci

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

github.com/oxcompsoc/learntocode